

A 40 nm 535 Mbps Multiple Code-Rate Turbo Decoder Chip Using Reciprocal Dual Trellis

Chen-Yang Lin, Cheng-Chi Wong, and Hsie-Chia Chang

Abstract—This paper presents a multiple code-rate turbo decoder using the reciprocal dual trellis to improve the hardware efficiency. For a convolutional code with code rate $k/(k+1)$, its corresponding reciprocal dual code with rate $1/(k+1)$ has smaller codeword space than the original code while $k > 1$, leading to a simplified trellis of the high code-rate code. The proposed decoder architecture can decode code rate $k/(k+1)$ constituent convolutional codes for $k = 1, 2, 4, 8$, and 16. Moreover, two parallel soft-in/soft-out (SISO) decoders are exploited in our turbo decoder by using the quadratic permutation polynomial (QPP) interleaver to improve the decoding speed. After fabricated in 1P9M CMOS 40 nm process, the proposed decoder with 1.27 mm² core area can achieve 535 Mbps throughput at 8/9 code rate, and the energy efficiency is 0.068 nJ/bit/iteration at 0.9 V.

Index Terms—High code rate, quadratic permutation polynomial (QPP) interleaver, reciprocal dual trellis, turbo decoder.

I. INTRODUCTION

TURBO codes were impressive with their excellent error correction ability [1]. Recently, several standards for wireless communication systems, such as 3GPP LTE [2], choose turbo codes as the channel codes to ensure reliable data transmission. A rate-1/3 turbo encoder consists of two recursive systematic convolutional (RSC) codes and one interleaver. The codeword contains the systematic information data as well as two parity check parts, which are coded from the information in normal and interleaved order respectively. A turbo decoder is constituted by interleavers and *Soft-In/Soft-Out* (SISO) decoders. The iterative decoding methodology, which contains two half-iterations including normal and interleaved decoding procedure, is adopted to the turbo decoder. During each half-iteration, the maximum *a posteriori* (MAP) probability algorithm [3] or Max-Log-MAP algorithm [4] is commonly employed in SISO decoders to compute the extrinsic value for estimating the *a priori* probability for the next half-iteration.

In the past few years, turbo decoders such as [5] and [6] usually focused on the optimization of throughput, hardware complexity, or power consumption; but only rate-1/3 decoders

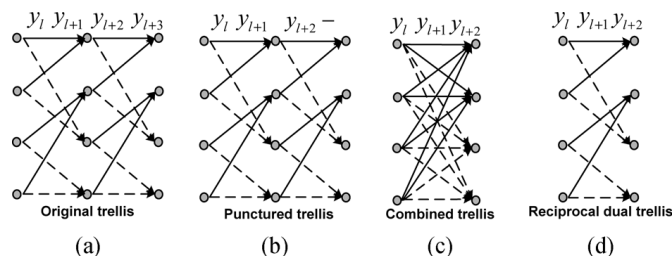


Fig. 1. Trellis transformation from punctured trellis to reciprocal dual trellis. (a) Original trellis with rate-1/2. (b) Punctured trellis with rate-2/3. (c) Combined trellis with rate-2/3. (d) Reciprocal dual trellis with rate-1/3.

are under consideration. Due to the growing demand of bandwidth efficiency, a turbo decoder supporting various code-rate schemes is required. The typical method to achieve a desired code rate is to puncture the rate-1/2 constituent convolutional code [7]. We usually prune $(k-1)$ parity check bits from every k codeword bits and get a punctured code with code rate $k/(k+1)$ for higher bandwidth efficiency; k is the number of information bits in a single trellis stage. Fig. 1 shows the trellis transformation from original convolutional trellis to its reciprocal dual trellis. After puncturing one parity bit within two trellis stages, the original trellis with rate-1/2 in Fig. 1(a) turns into punctured trellis with rate-2/3 as shown in Fig. 1(b). The punctured radix-2 trellis can be applied to decode punctured codes by using simple radix-2 Add-Compare-Select (ACS) circuit, however, the throughput can not be enhanced while the code rate rises. In order to improve the throughput at high code-rate codes, a high radix trellis structure is usually used. As illustrated in Fig. 1(c), two punctured trellis stages are merged into one, and there are two respective information bits at each combined trellis. More generally, k successive radix-2 punctured trellis stages can be merged into one radix- 2^k trellis stage. Since the total stage number of the radix- 2^k trellis is only $1/k$ of that of the punctured radix-2 trellis, the decoding speed based on the combined trellis is almost k times faster than that of the punctured radix-2 trellis. However, the complexity of the combined trellis increases exponentially according to the code rate, and the implementation of a decoder with such a radix- 2^k trellis structure is infeasible for large k . These disadvantages limit the development of turbo decoders applied for high code-rate schemes.

The reciprocal dual code, presented in [8], reduce the trellis complexity of the high code-rate decoder for a convolutional code with rate $k/(k+1)$. For such a punctured code, the code rate of its reciprocal dual code is $1/(k+1)$, which indicates that the number of codewords in the reciprocal dual trellis is smaller than the punctured trellis and the combined trellis. As depicted in Fig. 1(d), the combined radix-4 trellis with rate-2/3 can be transformed to its reciprocal dual trellis with rate-1/3.

Manuscript received February 02, 2013; revised May 15, 2013; accepted July 05, 2013. Date of publication August 15, 2013; date of current version October 19, 2013. This paper was approved by Guest Editors Hong June Park and Chang-Hyun Kim. This work was supported by the NSC of Taiwan, R.O.C., under grant NSC 101-2220-E-009-060.

The authors are with the Institute of Electronics, National Chiao Tung University, Hsinchu, Taiwan (e-mail: talent31022@gmail.com; ccwong@oasis.ee.nctu.edu.tw; hcchang@si2lab.org).

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/JSSC.2013.2274883

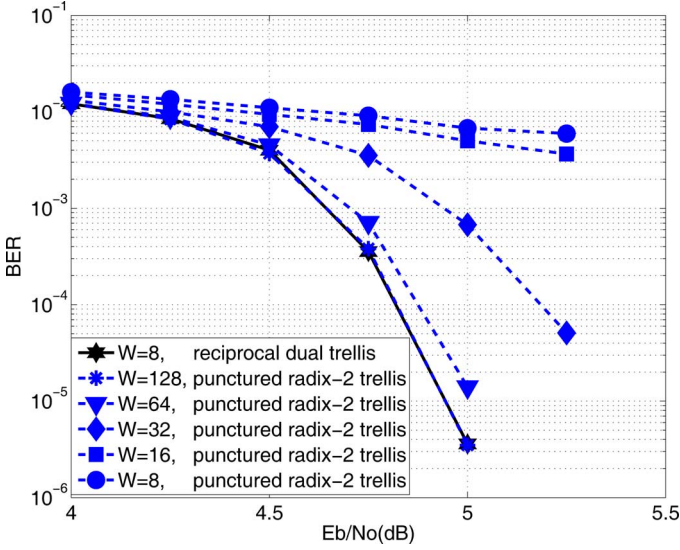


Fig. 2. Simulation with reciprocal dual trellis and punctured radix-2 trellis; block size 4096; 6 iterations.

TABLE I
PUNCTURED TABLES

k	1	2	4	8	16
Code-rate	1/3	1/2	2/3	4/5	8/9
Systematic	1	11	1111	11111111	1111111111111111
Parity 1	1	01	0001	00000001	0000000000000001
Parity 2	1	01	0001	00000001	0000000000000001
W	32	32	16	8	8

0: punctured; 1: transmitted

The combined trellis is a radix- 2^k structure and the number of the branches of each state is 2^k , whereas the corresponding reciprocal dual trellis is a radix-2 structure. To save the hardware complexity further, the logarithmic domain approach [9] is adopted to design our multiple code-rate turbo decoder. Besides the advantage of hardware reduction, the reciprocal dual trellis can reduce the length of sliding window [10] in high code-rate SISO decoder. The bit error rate (BER) performance of turbo decoders using reciprocal dual trellis and punctured radix-2 trellis with 8/9 code rate under additive white Gaussian noise (AWGN) is shown in Fig. 2, and the encoder is defined in 3GPP LTE [2]. The punctured RSC code is obtained by using the punctured pattern shown in Table I. The results show that the reciprocal dual trellis with sliding window length $W = 8$ has almost the same BER as the punctured radix-2 trellis with $W = 128$ at $\text{BER} = 10^{-5}$, and we define that a single trellis stage is a unit of sliding window. Due to smaller length of sliding window in high code-rate decoders, the decoding latency can be reduced. This feature makes the reciprocal dual trellis a preferable solution to high code-rate turbo decoders requiring faster processing speed.

In this work, hardware solutions for high code-rate turbo decoders are developed for communication systems requiring high decoding speed. A multiple code-rate turbo decoder with reciprocal dual trellis and parallel SISO processors is implemented, and the highest throughput with 535 Mbps is designed to fulfill high speed requirement for the next wireless communication systems.

The remainder of this paper is organized as follows. Section II introduces the decoding algorithm of reciprocal dual trellis as well as the methodology of multiple access from memories using QPP interleaver. The architecture of proposed turbo decoder using the reciprocal dual trellis is described in Section III. Section IV introduces the proposed multiple code-rate turbo decoder. Section V discusses the performance of the proposed architecture and gives the implementation results of the proposed turbo decoder. Finally, the conclusion is given in Section VI.

II. RECIPROCAL DUAL TRELIS WITH QPP INTERLEAVER

We briefly introduce the algorithm of turbo decoders using the reciprocal dual trellis and the QPP interleaver. According to [8], the log-likelihood ratio (LLR) of the extrinsic value $L(\hat{u}_l)$ at l th bit can be calculated from the reciprocal dual codes in (1):

$$L(\hat{u}_l) = \ln \frac{\sum_{\tilde{c}_i^\perp \in \tilde{\mathcal{C}}^\perp} \prod_{j=0, j \neq l}^{N-1} (q_j)^{\tilde{c}_{ij}^\perp}}{\sum_{\tilde{c}_i^\perp \in \tilde{\mathcal{C}}^\perp} (-1)^{\tilde{c}_{il}^\perp} \prod_{j=0, j \neq l}^{N-1} (q_j)^{\tilde{c}_{ij}^\perp}}. \quad (1)$$

Notice that \mathcal{C} is the codeword space of the original block code, and $\tilde{\mathcal{C}}^\perp$ is the codeword space of the reciprocal dual code of \mathcal{C} ; $\tilde{c}_i^\perp = (\tilde{c}_{i,0}^\perp, \tilde{c}_{i,1}^\perp, \dots, \tilde{c}_{i,N-1}^\perp)$, $1 \leq i \leq 2^{N-K}$, is the codeword of $\tilde{\mathcal{C}}^\perp$; $\tilde{c}_{i,j}^\perp \in \{0, 1\}$; N and K are the number of codeword bits and information bits, respectively. We define $q_j = \tanh(L(c_j; y_j)/2)$ as the bit metric and y_j is the received symbol at j th bit. $L(c_j; y_j)$ is equivalent to (2):

$$\ln \frac{P(c_j = 0; y_j)}{P(c_j = 1; y_j)} = \ln \frac{P(c_j = 0)}{P(c_j = 1)} + \ln \frac{P(c_j = 0|y_j)}{P(c_j = 1|y_j)}. \quad (2)$$

To simplify the computation of extrinsic value in (1), [9] and [11] adopt the sign-magnitude (SM) numerical type to represent the bit metric. In the following paragraph, the arithmetic operations using SM numerical type and a proposed equation for calculating the extrinsic value will be introduced.

A. Sign-Magnitude Scheme

In [9], a real number x is represented as $x = X_S e^{-X_M} \equiv [X_S; X_M]$, where $X_S \in \{1, -1\}$ and $X_M = -\ln(|x|)$ are the sign part and the magnitude part of x , respectively. The arithmetic with SM scheme can be defined as follows:

Addition:

$$\begin{aligned} & x + y \\ & \equiv \min^*(x, y) \\ & = \begin{cases} [X_S; X_M - \ln(1 + X_S Y_S e^{Y_M - X_M})], & \text{if } Y_M > X_M \\ [Y_S; Y_M - \ln(1 + X_S Y_S e^{X_M - Y_M})], & \text{otherwise} \end{cases} \end{aligned}$$

Multiplication:

$$x \times y \equiv \text{sum}^*(x, y) = [X_S Y_S; X_M + Y_M].$$

Division:

$$x \times y^{-1} \equiv \text{sub}^*(x, y) = [X_S Y_S; X_M - Y_M].$$

B. A New Equation for Extrinsic Value

With the SM representation, we propose an equation for calculating the extrinsic value. Similar to the conventional approach, the calculation of extrinsic values in the reciprocal dual

trellis also requires the forward recursion metric α , backward recursion metric β , and branch metric γ . Each of them can be written as:

- The branch metric γ :

$$\gamma_t(\mathbf{s}_{t-1}, \mathbf{s}_t) = \text{sum}^*(g_j^{b_j(\mathbf{s}_{t-1}, \mathbf{s}_t)}, \dots, g_{j+n-1}^{b_{j+n-1}(\mathbf{s}_{t-1}, \mathbf{s}_t)}).$$

- The forward recursion metric α :

$$\alpha_t(\mathbf{s}_t) = \min^*(\text{sum}^*(\alpha_{t-1}(\mathbf{s}_{t-1}), \gamma_t(\mathbf{s}_{t-1}, \mathbf{s}_t))).$$

- The backward recursion metric β :

$$\beta_{t-1}(\mathbf{s}_{t-1}) = \min^*(\text{sum}^*(\gamma_t(\mathbf{s}_{t-1}, \mathbf{s}_t), \beta_t(\mathbf{s}_t))).$$

Notice that g_j is the SM type of bit metric q_j , $t = \lfloor j/n \rfloor$ is the time index ($\lfloor x \rfloor$ denotes the integer part of x), $n = k + 1$, \mathbf{S} is the set of possible transitions from state \mathbf{s}_{t-1} to \mathbf{s}_t , and $b_j(\mathbf{s}_{t-1}, \mathbf{s}_t)$ is the transition bit from \mathbf{s}_{t-1} to \mathbf{s}_t at $(j - t \cdot n)$ position. The variable Q_l^b in (3) denotes the SM representation of metric summation which has bit 'b' at decoding index l th.

$$\begin{aligned} Q_l^b &\equiv [Q_{l,S}^b; Q_{l,M}^b] \\ &= \min^*(\text{sum}^*(\text{sum}^*(\alpha_{t-1}(\mathbf{s}_{t-1}), \gamma_t(\mathbf{s}_{t-1}, \mathbf{s}_t)), \beta_t(\mathbf{s}_t))). \end{aligned} \quad (3)$$

After computing Q_l^0 and Q_l^1 , we define a variable U_l , and the SM representation of U_l can be written as

$$U_l = U_{l,S} e^{-U_{l,M}} \equiv [U_{l,S}; U_{l,M}] \equiv \text{sub}^*(Q_l^0, \text{sum}^*(Q_l^1, g_l)).$$

With U_l and the deduction of [11], (1) can be simplified to (4)

$$L(\hat{u}_l) = \ln \frac{1 + U_l}{1 - U_l}. \quad (4)$$

While using the relation

$$\tanh\left(\frac{x}{2}\right) = \frac{(1 - e^{-x})}{(1 + e^{-x})},$$

(4) can be simplified as

$$L(\hat{u}_l) = \begin{cases} -\ln\left(\left|\tanh\left(\rho \times \frac{U_{l,M}}{2}\right)\right|\right), & \text{if } U_{l,S} = 1 \\ \ln\left(\left|\tanh\left(\rho \times \frac{U_{l,M}}{2}\right)\right|\right), & \text{otherwise.} \end{cases} \quad (5)$$

The variable ρ is the scaling factor which is equal to 1 by default. After calculating $[U_{l,S}, U_{l,M}]$, the proposed equation for computing the extrinsic value can be realized by a lookup table (LUT). As compared with (1), (5) is more suitable for hardware implementation. The LLR of *a posteriori* probability $L(u_l)$ at l th bit can be written as

$$L(u_l) = L(c_l; y_l) + L(\hat{u}_l). \quad (6)$$

Finally, the decision can be made by the sign part of $L(u_l)$.

C. Multiple Access From Memory

To achieve high throughput purpose, parallel SISO decodes are applied to decode a codeword [6], [12]. The QPP interleaver [13] with contention-free property allows parallel SISO decoders to decode a codeword without memory accessing conflict. The closed form of this interleaver is defined as:

$$\pi(\tau) = (f_1 \cdot \tau + f_2 \cdot \tau^2) \bmod(K)$$

where τ is the original address in normal order, f_1 and f_2 are dependent on the block size K , and $\pi(\tau)$ is the interleaved address. The method for generating $\pi(\tau)$ recursively is presented in [14], which can generate one address at each recursion. However, in our design, each SISO decoder should access k soft values in a single trellis stage. Hence, we extend the concept of the recursive calculation and derive equations for generating k successive addresses at each recursion.

Assume that $\tau = (k \cdot i + j)$ and $0 \leq i < K/(P \cdot k)$; $0 \leq j < k$. Parameter P is the number of adopted parallel SISO decoders, and each SISO decoder is responsible for decoding K/P information bit of a codeword. The QPP interleaver is divided into P parts, and the size of each part will be $k \times K/(P \cdot k)$. Fig. 3 depicts the methodology for generating addresses from the interleaver, and suppose that the SISO decoder is required to decode k information bits at i th trellis stage. There are k generators, and Fig. 3 shows the operation of j th generator. The address can be generated recursively by calculating the descending and ascending order, and both of them are described as follows:

Descending Order:

$$\begin{aligned} \pi(k \cdot (i - 1) + j), \quad 0 < j \quad \text{and} \quad 0 \leq j < k \\ &= f_1(ki + j) - kf_1 + f_2(ki + j)^2 - 2f_2k(ki + j) + f_2k^2 \\ &= [\pi(k \cdot (i) + j) - G(k, i, j) + k^2 f_2] \bmod(K) \end{aligned}$$

where i is the index of each recursion step, and $G(k, i, j) = 2kf_2 \cdot (ki + j) + kf_1$. As the recursion calculation starts k initial addresses depending on the length of sliding window (W) are required for the first descending calculation.

Ascending Order:

$$\begin{aligned} \pi(k \cdot (i + w) + j), \quad w = 2 \cdot (W) - 1 \quad \text{and} \quad 0 \leq j < k \\ &= f_1(ki + j) + f_2(ki + j)^2 + w \cdot G(k, i, j) + f_2w^2k^2 \\ &= [\pi(k \cdot (i) + j) + w \cdot G(k, i, j) + (wk)^2 f_2] \bmod(K). \end{aligned}$$

With the descending and ascending orders, each SISO decoder can access k values from memory banks at the same time by using k address generators.

III. TURBO DECODER WITH RECIPROCAL DUAL TRELLIS

The architecture of turbo decoder with rate- $k/(k + 1)$ constituent codes is shown in Fig. 4(a). The turbo decoder adopts the MAP algorithm [3] with the reciprocal dual trellis to the SISO decoder. The input memory and the extrinsic memory store received codewords from channel and K extrinsic values from the SISO decoder, respectively. In our design, there are k systematic memory banks and two parity memory banks, and each of them requires single port RAM. There are also k extrinsic memory banks which require dual port RAM for reading and writing data in a clock cycle. The address generator is responsible for generating addresses in normal or interleaved order for decoding at different half-iteration. The iterative decoding process alternates between two half-iterations. By calculating the descending and ascending order, k interleaved addresses can be generated recursively and the SISO decoder could access multiple data from memory banks simultaneously. The metric pre-processor is used to transform a real number to SM numerical type. Each of them contains an adder and a LUT for approximating the equation $-\ln(\tanh(L(c_j; y_j)/2))$.

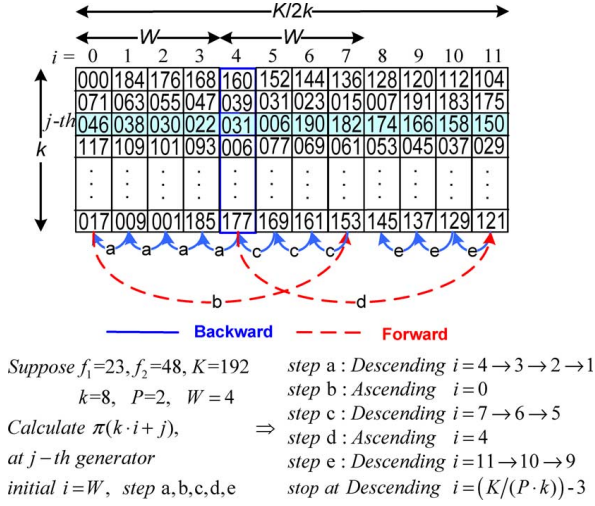


Fig. 3. Generator of QPP interleaver for first sub-block.

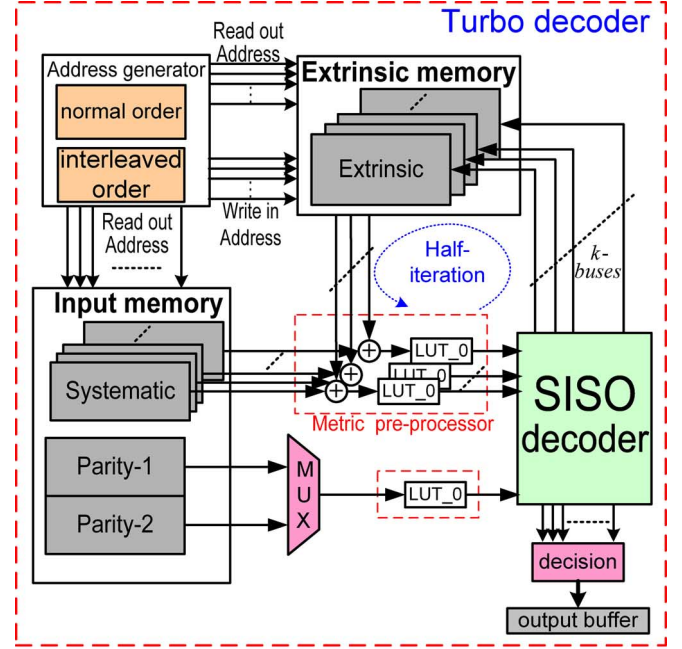
Hence, a soft value including its sign part and magnitude part will be stored in the following buffers of each SISO decoder.

The block diagram of the proposed SISO decoder is shown in Fig. 4(b). Since the technique of sliding window [10] is utilized to implement the SISO decoder, the window buffer is required to buffer the soft inputs for evaluating the path metrics. The γ , α , and β units are used to calculate the branch metrics, forward path metrics, and backward path metrics, respectively. The β_d unit, calculating the dummy backward path metric, is used to give more robust initialization for β unit [10]. The α -buffer performs the Last-In/First-Out for the reversing output order of α . In the SISO decoder, there are k parallel extrinsic units used for calculating extrinsic values in a single trellis stage, and the extrinsic values are produced in real number type. Based on the arithmetic of SM scheme, the following paragraph will introduce the related architectures of computational units in the SISO decoder.

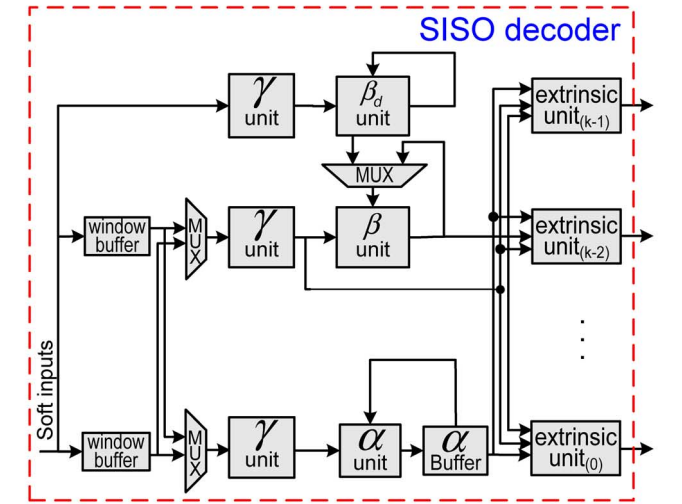
A. Branch Metric and Recursion Metric Architecture

Fig. 5 shows the branch metric unit for a constituent code with code rate $k/k+1$. The branch metric unit is responsible for performing sign-magnitude multiplication (SMM) successively and calculating the branch metric on a specific state transition within a single trellis stage. The branch metric unit consists of k SMM units, and each of SMM unit contains an exclusive-or gate, an adder and several multiplexers. The adder adds the magnitude part of two inputs, and the sign parts perform the exclusive-or operation. The binary bits $[b_0, b_1, \dots, b_k]$ are parts of a codeword from s_{t-1}^2 to s_t^0 on the decoding trellis. At last, the branch metric unit produces γ of the specific transition on the trellis.

The recursion metric unit performing SMM and sign-magnitude addition (SMA) is illustrated in Fig. 6. In radix-2 trellis structure, two sets of α (or β) and γ are inputs of SMA unit. Each SMM computes the SM pair of $[X_S; X_M]$ and $[Y_S; Y_M]$, respectively. The sign bit representing the sign part of the value $(X_M - Y_M)$ determines the sign part of the output of SMA unit, and the sign bit also selects the minimum value denoted as $\min(X_M, Y_M)$ from X_M and Y_M . The ABS unit produces the absolute value of the difference between X_M and Y_M . Two LUTs are used to approximate the correction terms of $-\ln(1 + e^{-|X_M - Y_M|})$ and $-\ln(1 - e^{-|X_M - Y_M|})$ described in



(a)



(b)

Fig. 4. Architecture of proposed turbo decoder. (a) Proposed turbo decoder using reciprocal dual trellis. (b) Proposed SISO decoder using reciprocal dual trellis.

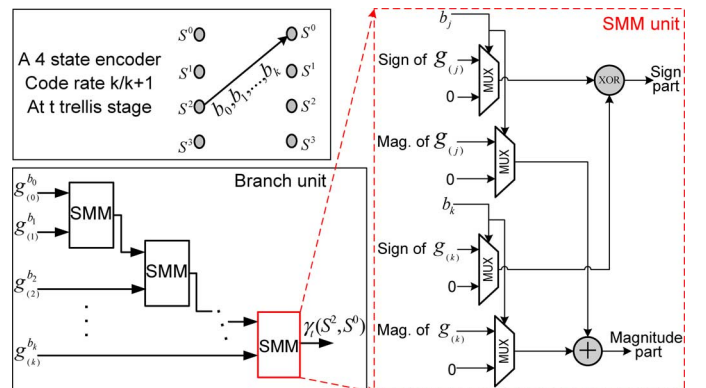


Fig. 5. Architecture of branch metric unit.

Section II-A. Finally, the adder calculates the magnitude part Z_M of the output in SMA unit by adding the correction term

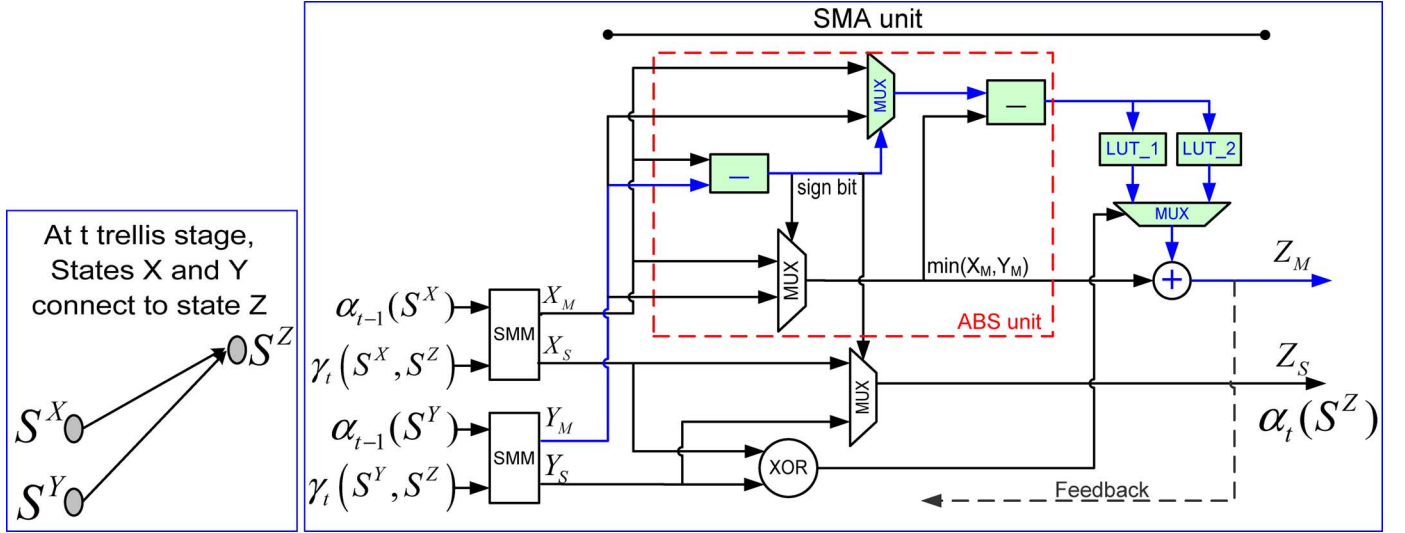


Fig. 6. Architecture of recursion metric unit.

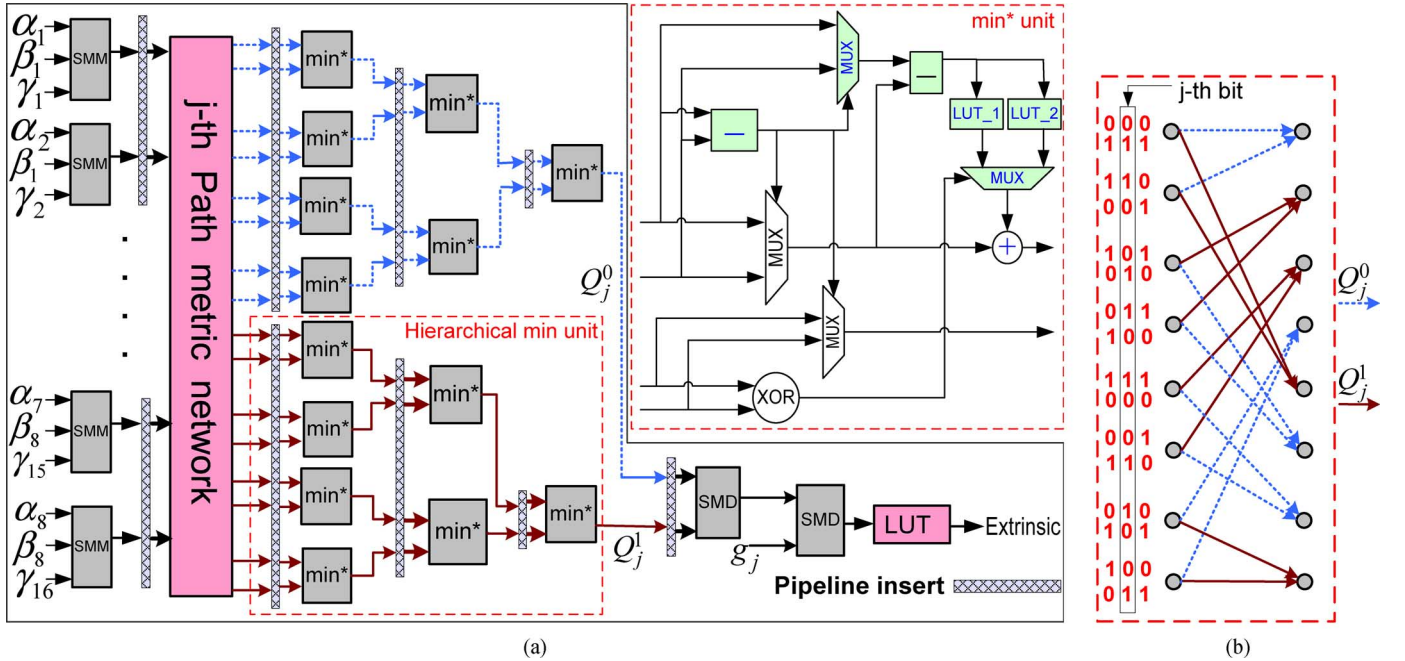


Fig. 7. Architecture of extrinsic unit. (a) Architecture of extrinsic unit. (b) The path metric network unit.

to the value of $\min(X_M, Y_M)$. To avoid overflow in finite data width, we adopt the modulo normalization of path metric [15] in our design. The data path of the recursion metric unit is the critical path of the SISO decoder.

B. Extrinsic Architecture

The architecture of extrinsic unit proposed according to Section II-B is illustrated in Fig. 7(a). The inputs are forward metrics, backward metrics, branch metrics, and the bit metric g_j is at the decoding index j . Due to radix-2 trellis architecture, there are 16 SMM units to compute path metrics. At first, each path metric including α , β , and γ will be accumulated and delivered to each extrinsic unit. The path metric network unit is the connection between path metrics and two hierarchical min processors. The connection of path metric network in each extrinsic unit is according to the decoding index j illustrated

in Fig. 7(b), and Fig. 7(b) shows the example of a constituent convolutional code with rate-2/3. Within each path metric network, if a transition bit at the decoding index j is 0, then the corresponding branch metric will be accumulated to calculate Q_j^0 ; otherwise, this branch metric will be accumulated to calculate Q_j^1 . The hierarchical min unit consisting of 7 \min^* units is used to calculate the summation metric described in (3). There are two hierarchical min units in one extrinsic unit: one is for the calculation of Q_j^0 , and the other is for Q_j^1 . The value U_l defined in II-B can be calculated by using SM division (SMD) units. Finally, a LUT is required to approximate $\ln(|\tanh(\rho \times U_{l,M}/2)|)$ and outputs the extrinsic value.

C. Design Merits of Reciprocal Dual Trellis

As shown in Fig. 1(c), high speed SISO decoders can be implemented by using the combined trellis because multiple infor-

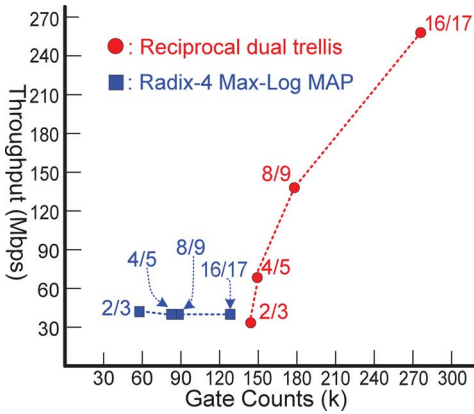


Fig. 8. Design space exploration for SISO decoders.

mation bits can be decoded in a single trellis stage. Hence, the number of systematic bits within a single trellis stage is a critical parameter to achieve high throughput SISO decoder. However, it's not easy to implement a code-rate $k/k+1$ SISO decoder with large k . Fig. 8 shows the design space of throughput and circuit area (k -gate) for SISO decoders using reciprocal dual trellis and conventional radix-4 Max-Log-MAP [4]. Four dedicated SISO decoders with convolutional code rates $2/3$, $4/5$, $8/9$, and $16/17$ are investigated and synthesized¹ by using 90-nm CMOS process.

In conventional radix-4 trellis, the circuit area increases for higher code-rate decoder because large sliding window size is required to obtain convergent BER. Because there are two systematic bits within a single radix-4 trellis stage, the throughput of each code-rate scheme is almost the same. On the other hand, due to the adoption of parallel extrinsic units, the throughput of SISO decoder using reciprocal dual trellis can be increased while decoding high code-rate codes. Though the circuit area grows as the code rate rises, the improvement of throughput of the reciprocal dual trellis is almost doubled. The hardware efficiency (Throughput/ k -gate) of rate- $16/17$ decoders are 0.93 and 0.29 for reciprocal dual trellis and conventional radix-4 Max-Log-MAP, respectively. For code rate $k/k+1$ convolutional codes, their reciprocal dual trellis are radix-2 structures and capable of computing k extrinsic values at the same time in a single trellis stage. SISO decoders with reciprocal dual trellis enjoy merits of punctured radix-2 trellis and the combined trellis structure both. Therefore, the throughput of the proposed SISO decoder can be improved while k increases. This is the key feature allowing reciprocal dual trellis to obtain high decoding speed.

IV. HIGH SPEED AND MULTIPLE CODE-RATE OPERATION

Another important design parameter regarding the high throughput target is the number of parallel SISO decoders adopted in a turbo decoder. In our design, in order to improve the throughput further, parallel SISO decoders with reciprocal dual trellis are employed. Based on the analysis of high speed design parameters, we choose $16/17$ as the maximum convolutional code rate and adopt two parallel SISO decoders to design

¹The throughput is estimated under 6 iterations and 4096 of block size. The circuit area is relevant to the size of sliding window which allows similar BER for two algorithms in each code-rate case.

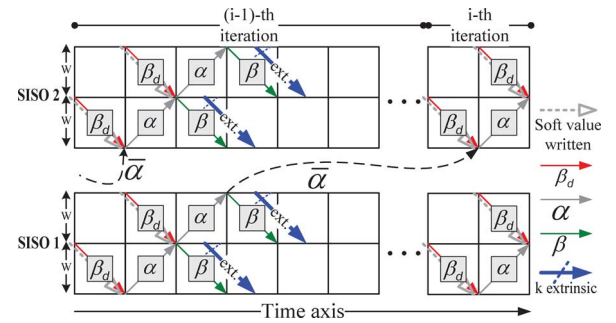


Fig. 9. Decoding schedule of parallel SISO decoders.

a turbo decoder exceeding 500 Mbps throughput. Furthermore, computational units such as branch metric and extrinsic units in highest code-rate decoders can support hardware requirements for low code-rate operations. Hence, a multiple code-rate turbo decoder model can be derived.

A. Decoding Schedule With Sliding Window

Fig. 9 shows the decoding schedule using two parallel SISO decoders to decode a codeword, and each SISO decoder is responsible for decoding half length of codeword. The soft values are written into the window buffers in the reversing order, and the branch metric unit for computing β_d is activated meanwhile. The β_d unit produces reliable initial boundary for β unit. The computational units γ , α , and β are successively calculated within this schedule of each window. Due to the sliding window approach and calculation of β_d , k parallel extrinsic units are activated to calculate extrinsic values while the β unit computes the backward recursion metric for each sliding window. Hence, k extrinsic values can be delivered from each single trellis stage simultaneously, leading to fast decoding speed. The $\bar{\alpha}$ is the forward path metric at the last trellis stage in SISO decoder 1. In order to maintain reliable BER performance, $\bar{\alpha}$ at $(i-1)$ th iteration is stored in the SISO decoder 1 and given (denoted as dotted curve arrow) to SISO decoder 2 at i th iteration to evaluate the initial forward path metric during two successive iterations [16].

B. Multiple Code-Rate Operation

In this paragraph, we describe issues of designing multiple code-rate turbo decoders based on the architecture of high code-rate turbo decoders. Suppose the highest code rate of the constituent encoder is $k/(k+1)$, and we use k memory banks for accessing k input symbols and the extrinsic values in one clock cycle. However, the access of successive k soft values at the same time may lead to conflicts between memory banks. To avoid the accessing conflicts, an interleaver with contention-free property [13] is adopted. Moreover, by adding several control circuits and multiplexers for memory addressing, the turbo decoder can decode the codeword with code rate $\kappa/(\kappa+1)$, where k is divisible by κ .

Fig. 10 shows the proposed turbo decoder with two SISO decoders. Each SISO decoder can access $2 \times \kappa$ soft values from each memory at one clock cycle, and $\kappa \in \{1, 2, 4, 8, 16\}$. In our design, the maximum constituent code rate is $16/17$, hence, 16 extrinsic units can be activated at the same time to compute the

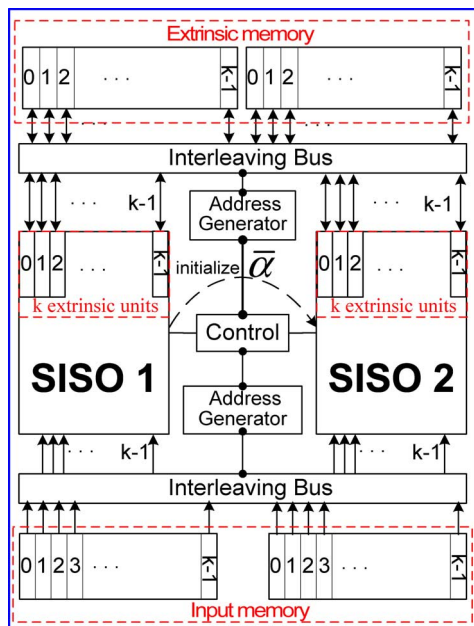


Fig. 10. Block diagram of multiple code-rate turbo decoder.

extrinsic values in each SISO decoder. In each half-iteration, the number of clock cycles for calculating K extrinsic values can be reduced to $K/(2 \times 16)$. Therefore, the throughput of the proposed architecture can be improved as κ increases, and it can be evaluated in (7). Notice that f is the operation frequency; L is the execution time of the first two sliding windows; C is the pipeline delays at each half-iteration; and η is the number of half-iterations.

$$\text{Throughput} \left(\frac{\text{bits}}{\text{s}} \right) = \frac{f \times K}{\eta \times \left(\frac{K}{2 \times \kappa} + L + C \right)}. \quad (7)$$

In addition, the codeword of each code-rate code in the reciprocal dual trellis is different. Therefore, the connection of the path metric network units and the transition bits of branch metric units vary with operated code-rate schemes. k parallel extrinsic units in highest code-rate design can be assigned to the decoding process for lower code-rate schemes.

V. SIMULATION AND IMPLEMENTATION RESULTS

In our design, the basic generator polynomial of the rate-1/2 constituent code is $[1 \quad 1 + D + D^3/1 + D^2 + D^3]$. With the punctured table in Table I, it can generate rate- $k/(k+2)$ turbo codes for $k = 1, 2, 4, 8$, and 16. This table also lists the length of sliding window (W) for acceptable performance at each code-rate code. To implement SMA and extrinsic units, three LUTs are applied to approximate equations: $-\ln(\tanh(d/2))$, $-\ln(1 - e^{-d})$, and $-\ln(1 + e^{-d})$. The first two equations are symmetric about equation d . By using this property, the number of parameters storing in each LUT can be reduced. In this design, scaling factors can be applied to slightly modify $\ln(\tanh(d/2))$ formulated in (5). Furthermore, we apply the scaling factor 0.75 to 1/3 and 1/2 rates and 0.875 to 2/3, 4/5, and 8/9 rates. These scaling factors are chosen owing to the consideration of hardware friendly multiplication and tolerable

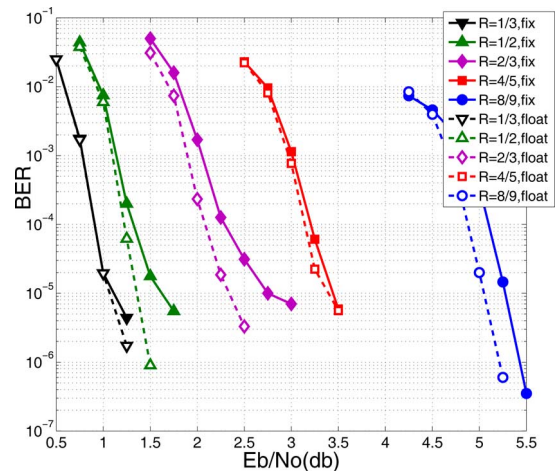


Fig. 11. Performance of various code rates ($R = 1/3, 1/2, 2/3, 4/5$, and $8/9$); the simulation is under AWGN channel, BPSK modulation, and fixed 6 iterations.

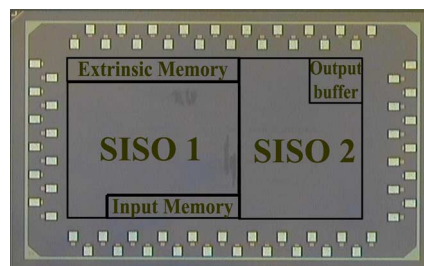


Fig. 12. Chip photo of multiple code-rate turbo decoder.

BER. The quantization scheme of input data consisting 3-bits integer part and 3-bit fraction part leads to less than 0.25 dB degradation at $\text{BER} = 10^{-4}$.

The proposed multiple code-rate turbo decoder is implemented by standard cell design flow and fabricated in 40 nm 1P9M CMOS process. The chip die photo is shown in Fig. 12, the core of the chip, contains 900 k gate counts (excluding memories) and uses 152 kb of RAM, occupies 1.27 mm² area. The measurement results of power consumption and throughput under different supply voltages and code-rate codes at $\text{BER} = 10^{-5}$ are illustrated in Fig. 13. Since the parallel extrinsic units are applied in each SISO decoder, the throughput of the turbo decoder can be enhanced while decoding on higher code rate. The throughput of our design can achieve 42, 78, 152, 294, and 535 Mbps while operating at code rate 1/3, 1/2, 2/3, 4/5, and 8/9, respectively. Since the speedup in throughput is larger than the growth in power, the energy efficiency is improved while the code rate rises. The measurement results indicate that the power consumption of the chip is 218 mW at 8/9 code rate with 0.9 V supply voltage, and the energy efficiency is 0.068 nJ/bit/iteration.

From the implementation results, we demonstrated that it's feasible to design a high throughput of high code-rate turbo decoder using the reciprocal dual trellis. Table II lists the comparison results with other works. The proposed design and [17] use reciprocal dual trellis for decoding while [18], [19], and [5] adopt the conventional trellis. Due to the iterative decoding

TABLE II
COMPARISON WITH OTHER TURBO DECODERS

	Proposed	JSSC'2011 [5]	ISCAS'2010 [17]	ISSCC'2003 [18]	DATE'2010 [19]
Technology	40 nm	130 nm	90 nm	180 nm	65 nm
Block size	4096	All LTE sizes	2048	32 to 432	All LTE sizes
Supply voltage (V)	0.9	1.2	0.9	1.8	1.1
Code rate	1/3, 1/2, 2/3, 4/5, 8/9	1/3	1/3, 1/2, 2/3, 4/5	1/3, 1/2, 2/3, 3/4	1/3 to 0.95
Operation frequency (MHz)	252	302	200	170.9	300
Max. throughput (Mbps)	535 at 6 iter.	390.6(358 ^a) at 5.5 iter.	101 at 6 iter.	80.7(53.8 ^a) at ~4 iter.	153(165.8 ^a) at 6.5 iter.
Core area (mm ²)	1.27	3.57	1.96	7.16	2.1
Gate counts (k)	900	553	370	373	N/A
Memory (kb)	152	129	58	36	N/A
Power consumption (mW)	218	788.9	71	N/A	300
Area efficiency (Mbps/k-gates) ^a	0.594	0.647	0.273	0.144	N/A
Energy efficiency (nJ/bit/iter) ^a	0.068	0.37	0.117	1.45	0.302
Status	Measurement	Measurement	Post-Layout	Measurement	Post-Layout

^a Throughput normalization to 6 iteration

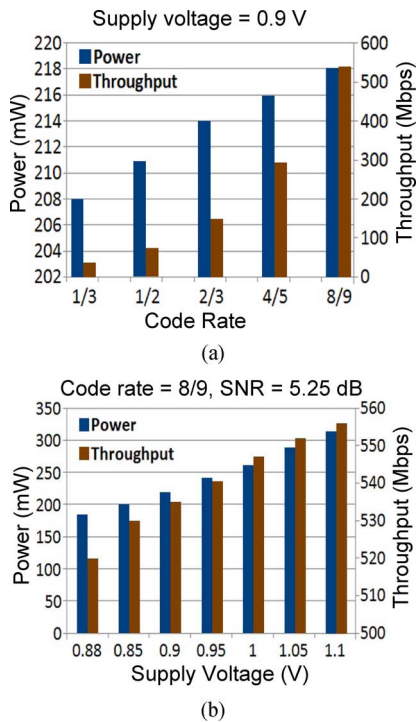


Fig. 13. Measured throughput and power. (a) At different code rates. (b) At different supply voltages.

procedure of turbo decoder, the throughput is inversely proportional to the number of iterations. Therefore, the throughput of each design is normalized to 6 iterations for comparison. The turbo decoder in [5] utilizes 8 SISO decoders with radix-4 trellis structure and has the highest area efficiency, this decoder is optimized for decoding 1/3 code-rate codes. Besides the adoption of higher code-rate scheme in two parallel SISO decoders, smaller size of sliding window is applied in the proposed design, hence, the proposed turbo decoder achieve higher area efficiency than our previous work [17]. As compared at the same implementation status, the proposed turbo decoder has higher throughput than other multiple code-rate turbo decoders.

VI. CONCLUSION

This work implemented a multiple code-rate turbo decoder by adopting the parallel processing with reciprocal dual trellis to avoid high radix trellis structure and achieve fast decoding speed at high code-rate schemes. In our design, the radix-2 trellis structure is applied to the convolutional decoder with code rate $k/(k+1)$. By exploiting two parallel SISO decoders and reciprocal dual trellis with 16 parallel extrinsic units, the proposed turbo decoder can achieve 535 Mbps. The measured power consumption shows that the energy efficiency of the proposed decoder is 0.068 nJ/bit/iteration while operating at 8/9 code-rate code under 0.9 V supply voltage. As compared with related state-of-the-art designs, we believe that the reciprocal dual trellis is the preferable solution to design high code-rate turbo decoders for high throughput applications.

ACKNOWLEDGMENT

The authors would like to thank TSMC for chip fabrication and Chip Implementation Center for providing the CAD tools and testing environment.

REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Commun.*, May 1993, pp. 1064–1070.
- [2] *Technical Specification Group Radio Access Network; Multiplexing and Channel Coding*, 2010, 3GPP TS 36.212 Std. V10.10.0.
- [3] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.
- [4] P. Robertson, E. Vilebrun, and P. Hoeher, "A comparison of optimal and sub-optimal decoding algorithms," in *Proc. IEEE Int. Conf. Commun.*, May 1995, pp. 1009–1013.
- [5] C. Studer, C. Benkeser, S. Belfanti, and H. Qiting, "Design and implementation of a parallel turbo-decoder ASIC for 3GPP-LTE," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 8–17, Jan. 2011.
- [6] C. C. Wong and H. C. Chang, "High-efficiency processing schedule for parallel turbo decoders using QPP interleaver," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 6, pp. 1412–1420, Jun. 2011.
- [7] L. H. C. Lee, "New rate-compatible punctured convolutional codes for viterbi decoding," *IEEE Trans. Commun.*, vol. 42, no. 12, pp. 3073–3079, Dec. 1994.

- [8] S. Riedel, "Map decoding of convolutional codes using reciprocal dual codes," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 1176–1187, May 1998.
- [9] G. Montorsi and S. Benedetto, "An additive version of the SISO algorithm for the dual code," in *IEEE Int. Symp. on Inform. Theory*, Jun./Jul. 2001, p. 27.
- [10] S. A. Barbulescu, "Sliding window and interleaver design," *IEEE Electronics Lett.*, vol. 37, no. 21, pp. 1299–1300, Oct. 2001.
- [11] S. Sudharshan and S. S. Pietrobon, "Log domain implementation of the dual-APP algorithm," in *Australian Commun. Theory Workshop*, Feb. 2005, pp. 100–104.
- [12] C. C. Wong and H. C. Chang, "Reconfigurable turbo decoder with parallel architecture for 3GPP LTE system," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 7, pp. 566–570, Jul. 2010.
- [13] O. Y. Takeshita, "On maximum contention-free interleavers and permutation polynomials over integer rings," *IEEE Trans. Inform. Theory*, vol. 52, no. 3, pp. 1249–1253, Mar. 2006.
- [14] R. Asghar and D. Liu, "Dual standard re-configurable hardware interleaver for turbo decoding," in *Proc. Int. Symp. Wireless Pervasive Computing ISWPC*, May 2008, pp. 768–772.
- [15] C. Shung, P. Siegel, G. Ungerboeck, and H. Thapar, "VLSI architectures for metric normalization in the Viterbi algorithm," in *Int. Conf. Commun.*, Atlanta, CA, Apr. 1990, vol. 4, pp. 1723–1728.
- [16] S. Yoon and Y. Bar-Ness, "A parallel MAP algorithm for low latency turbo decoding," *IEEE Commun. Lett.*, vol. 6, no. 7, pp. 288–290, Jul. 2002.
- [17] C.-Y. Lin, C.-C. Wong, and H.-C. Chang, "A multiple code-rate turbo decoder based on reciprocal dual trellis architecture," in *Proc. IEEE ISCAS*, May 2010, pp. 1496–1499.
- [18] B. Bougard, A. Giulietti, V. Derudder, J.-W. Weijers, S. Dupont, L. Hollevoet, F. Catthoor, L. VanderPerre, H. DeMan, and R. Lauwereins, "A scalable 8.7 nJ/bit 75.6Mb/s parallel concatenated convolutional (turbo-) codec," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2003, pp. 152–153.
- [19] M. May, T. Inseher, N. Wehn, and W. Raab, "A 150 Mbit/s 3GPP LTE turbo code decoder," in *Proc. DATE*, Dresden, Germany, Mar. 2010, pp. 1420–1425.



Chen-Yang Lin received the B.S. degree in electrical control engineering and the M.S. degree in electronics engineering in 2007 and 2009, respectively from National Chiao Tung University, Hsinchu, Taiwan. He is currently pursuing the Ph.D. degree in the Institute of Electronics, National Chiao Tung University.

His research interests include algorithm and VLSI architecture of wireless communication systems, especially for error control codes.



Cheng-Chi Wong received the B.S. and Ph.D. degrees in electrical engineering from National Chiao Tung University, Hsinchu Taiwan, in 2004 and 2010, respectively.

His research interests include algorithms and VLSI architecture of error correction codes.



Hsie-Chia Chang received the B.S., M.S., and Ph.D. degrees from National Chiao Tung University, Hsinchu, Taiwan, in 1995, 1997, and 2002, respectively, all in electronics engineering.

From 2002 to 2003, he was with OSP/DE1 in MediaTek Corporation, working in the area of decoding architectures for Combo single chip. In February 2003, he joined the faculty of the Electronics Engineering Department, National Chiao-Tung University, where he has been a Professor since August 2010. His research interests include algorithms and

VLSI architectures in signal processing, especially for error control codes and crypto-systems. Recently, he has also committed himself to designing high code-rate ECC schemes for flash memory and multi-Gb/s chip implementations for wireless communications

Dr. Chang served as an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I: REGULAR PAPERS since 2012. He also served as a Technical Program Committee (TPC) member for IEEE A-SSCC 2011, 2012, and 2013. He was the recipient of the Outstanding Youth Electrical Engineer Award from Chinese Institute of Electrical Engineering in 2010, and the Outstanding Youth Researcher Award from Taiwan IC Design Society in 2011.