

Low-Power Log-MAP Decoding Based on Reduced Metric Memory Access

Dong-Soo Lee, *Member, IEEE*, and In-Cheol Park, *Senior Member, IEEE*

Abstract—Due to the powerful error correcting performance, turbo codes have been adopted in many wireless communication standards such as W-CDMA and CDMA2000. Although several low-power techniques have been proposed, power consumption is still a major issue to be solved in practical implementations. Since turbo decoding is classified as a memory-intensive algorithm, reducing memory accesses is crucial to achieve a low power design. To reduce the number of memory accesses for maximum *a posteriori* (MAP) decoding, this paper proposes an approximate reverse calculation method that can be implemented with simple arithmetic operations such as addition and comparison. Simulation results show that the proposed method applied to the W-CDMA standard reduces the access rate of the backward metric memory by 87% without degrading error-correcting performance. A prototype log-MAP decoder based on the proposed reverse calculation achieves 29% power reduction compared to a conventional decoder that does not use the reverse calculation.

Index Terms—Low power design, maximum *a posteriori* (MAP) algorithm, memory optimization, reverse calculation, turbo codes, turbo decoding.

I. INTRODUCTION

SINCE turbo coding was introduced in 1993 [1], it has been recognized as one of the most powerful forward error correction codes. Recently, turbo codes were accepted in many standardized mobile radio systems such as W-CDMA, CDMA2000, DVB-RCS, and IEEE 802.16 (WiMax). Various studies have focused on their practical implementations [2]–[7], including a scalable turbo codec [8] and a unified turbo/Viterbi channel decoder [9].

A turbo decoder consists of two decoding components each of which operates iteratively to produce improved soft outputs by using the outputs of the other component. However, owing to its iterative decoding procedure and the requirement of frequent memory accesses, the turbo decoder suffers from long latency and high power consumption. Several techniques to overcome these problems have been presented to achieve low-power turbo decoders.

As the turbo decoding is an iterative algorithm that improves error correcting performance based on the previously calculated

log-likelihood ratios (LLRs), it is important to stop the iteration as early as possible in order to achieve a low-power design. As the number of iterations is directly related to the power consumption, a number of stopping criteria have been proposed to decide the termination point [10], [11]. Though the early termination is one of the most effective low power techniques [12], it already approaches the theoretical limit because the average number of iterations is very close to the minimal iterations.

Another useful technique is to supply a scaled voltage to noncritical blocks [13], [14]. The margin of critical timing is analyzed to lower the supply voltage for noncritical blocks operating at the same clock frequency as the timing-critical blocks. In addition, the decoder structure is reorganized to achieve more noncritical blocks. In [15], the interleaver memory is partitioned to make the access time shorter, and a scaled lower supply voltage drives the interleaver block. In [16], the folding and retiming techniques are applied to make more blocks operating with a lower supply voltage.

As the turbo decoder is included in a class of highly memory-intensive systems, a significant amount of power is consumed for memory accesses, resulting in a power bottleneck even though the decoder uses the sliding window processing to reduce the memory size significantly. It has been reported that the memory access power accounts for more than 50% of the entire power consumption [17]. A complex address generation algorithm for the interleaving is implemented on-the-fly instead of storing the interleaver addresses in a table [18]. The partial metric storage method proposed in [17] replaces some parts of the metric memory with a register file and computes the lost metrics redundantly. The weakness of this method is that the power consumed in the register file increases rapidly if many metric values are stored into a large-sized register file. In practice, therefore, the replacement is limited up to a quarter of the memory size. In addition, a tradeoff must be made to determine the replacement size by performing a simulation for the process technology to be used in actual design. A soft-input soft-output (SISO) architecture based on two memories has been proposed to reduce the memory accesses at the cost of some computational logic overhead [15].

The reverse calculation of state metrics is another efficient method to reduce memory accesses as reported in [19] and [20], which demonstrate that most metric memory accesses can be substituted by the reverse computation of forward or backward metrics. The rationale behind this approach is that the power needed to access the metric memory is greater than that of the corresponding computation, which is usually valid for today's deep submicrometer technology [21], [22]. However, the quantization and singular matrix problems are not solved in [20], and

Manuscript received August 15, 2004; revised July 8, 2005. This work was supported in part by the Institute of Information Technology Assessment through the ITRC, by the Korea Science and Engineering Foundation through MICROS center and by IC Design Education Center. This paper was recommended by Associate Editor Z. Wang.

D.-S. Lee is with System LSI Division, Samsung Electronics, Inc., Suwon 443-742, Korea (e-mail: dongsoo3.lee@samsung.com).

I.-C. Park is with the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Korea (e-mail: icpark@ee.kaist.ac.kr).

Digital Object Identifier 10.1109/TCSI.2006.870901

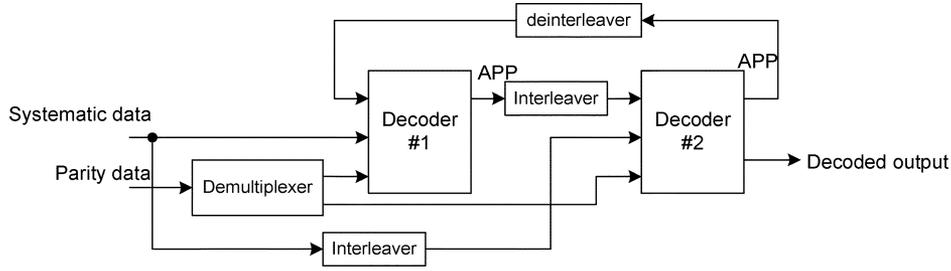


Fig. 1. Structure of a turbo decoder.

the modifications introduced to solve these problems are not efficient in view of real applications [19].

In this paper, we propose a new approximate reverse calculation for backward metrics. In the process of backward metrics calculation, only 13% of the calculated metrics corresponding to singular matrix calculations are written into a memory, while the others are not saved because they can be recovered by the reverse calculation. When backward metrics are needed, they are read from the memory or recovered by the proposed reverse calculation. As a result, we can reduce the metric memory accesses and the overall power consumption by 87% and 29%, respectively.

The remainder of the paper is organized as follows. In Section II, the log maximum *a posteriori* (MAP) decoder is described. The existence of the reverse butterfly structure and the approximate reverse calculation are presented in Section III. Section IV addresses a metric memory structure optimized for the reverse calculation. Experimental results are presented in Section V along with comparisons to a conventional turbo decoder.

II. LOG-MAP ALGORITHM

A turbo code is composed of the parallel or serial concatenation of two recursive systematic convolutional (RSC) codes. An interleaver is used to scramble the order of the input bits before feeding them into the second encoder. The output stream of the turbo encoder is generated by multiplexing the message bit and the parity bits obtained from the two RSC encoders.

The turbo decoding structure consists of two SISO decoding modules which are separated by a pseudo-random interleaver/deinterleaver. A conventional turbo decoder is shown in Fig. 1. Based on the MAP algorithm [1], the output for the k th symbol is expressed in LLR form as

$$\Lambda_k = \ln \frac{\sum_{s1} \alpha_k(s_k) \cdot \gamma_{k+1}(s_k \rightarrow s_{k+1}) \cdot \beta_{k+1}(s_{k+1})}{\sum_{s0} \alpha_k(s_k) \cdot \gamma_{k+1}(s_k \rightarrow s_{k+1}) \cdot \beta_{k+1}(s_{k+1})} \quad (1)$$

where α , β , and γ stand for forward, backward, and branch metrics, respectively, s_k represents a state of the encoder at time k , $s_k \rightarrow s_{k+1}$ is the state transition from state s_k to state s_{k+1} , and $s0$ and $s1$ denote the sets of all possible state transitions associated with message bit 0 and 1, respectively.

To simplify the calculation of α and β metrics by changing multiplication to addition, the Jacobian logarithm is applied to produce the following (2) and (3), where A is the set of states

at time $k-1$ that are connected to state s_k , and B is the set of states at time $k+1$ that are connected to state s_k

$$\begin{aligned} \ln[\alpha_k(s_k)] &= \bar{\alpha}_k(s_k) \\ &= \max_{s_{k-1} \in A}^* [\bar{\alpha}_{k-1}(s_{k-1}) + \bar{\gamma}_k(s_{k-1} \rightarrow s_k)] \end{aligned} \quad (2)$$

$$\begin{aligned} \ln[\beta_k(s_k)] &= \bar{\beta}_k(s_k) \\ &= \max_{s_{k+1} \in B}^* [\bar{\beta}_{k+1}(s_{k+1}) + \bar{\gamma}_{k+1}(s_k \rightarrow s_{k+1})]. \end{aligned} \quad (3)$$

In the above equations, \max^* is defined as

$$\max^*(x, y) = \ln(e^x + e^y) = \max(x, y) + \ln(1 + e^{-|y-x|}) \quad (4)$$

and γ metrics are represented as

$$\ln[\gamma_k(s_k \rightarrow s_{k+1})] = \bar{\gamma}_k(s_k \rightarrow s_{k+1}) = \ln[P(\mathbf{y}_k | \mathbf{x}_k) \cdot P(u_k)] \quad (5)$$

where u_k is the input bit that causes the transition from s_k to s_{k+1} , $P(u_k)$ is the *a priori* probability of u_k , and \mathbf{x}_k and \mathbf{y}_k are the transmitted and received codewords associated with this transition. A specific expression for γ metrics can be induced from the channel condition and the modulation scheme. Therefore, the LLR outputs can be obtained by

$$\begin{aligned} \Lambda_k &= \max_{s1}^* [\bar{\alpha}_k(s_k) + \bar{\gamma}_{k+1}(s_k \rightarrow s_{k+1}) + \bar{\beta}_{k+1}(s_{k+1})] \\ &\quad - \max_{s0}^* [\bar{\alpha}_k(s_k) + \bar{\gamma}_{k+1}(s_k \rightarrow s_{k+1}) + \bar{\beta}_{k+1}(s_{k+1})] \end{aligned} \quad (6)$$

where $s0$ and $s1$ are two state sets of time index k associated with message bit 0 and 1.

As indicated in (2) and (3), α and β metrics are recursively calculated in the forward and backward directions, and thus they are called forward and backward metrics, respectively. In a conventional way, as the directions of updating α and β metrics are opposite to each other, one of the two metrics is calculated and stored in a metric memory before computing the other metrics, and retrieved later when it is needed to compute the LLR output defined in (6). In this paper, β metrics are calculated prior to α metrics.

To reduce the memory size required to store metric values, a large frame is split into a number of windows, and the MAP decoding can be performed on each window independently. This technique is called sliding window processing [23]. If a frame is split into independent windows, it is difficult to know the initial metric values for a window. To have a reasonable initial metric, the metric calculation is performed for an overlapped window

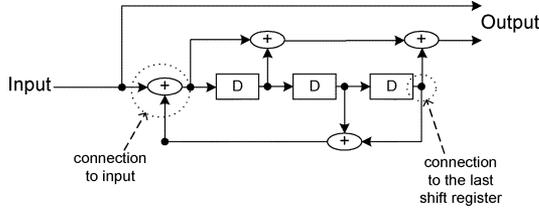


Fig. 2. A RSC encoder having butterfly pairs.

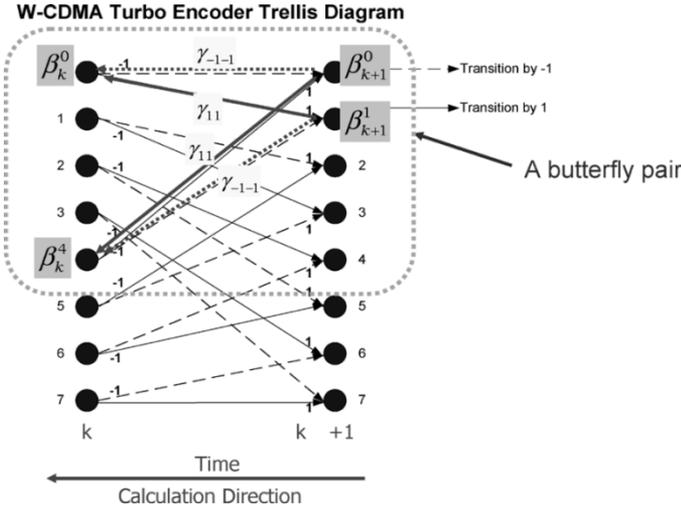


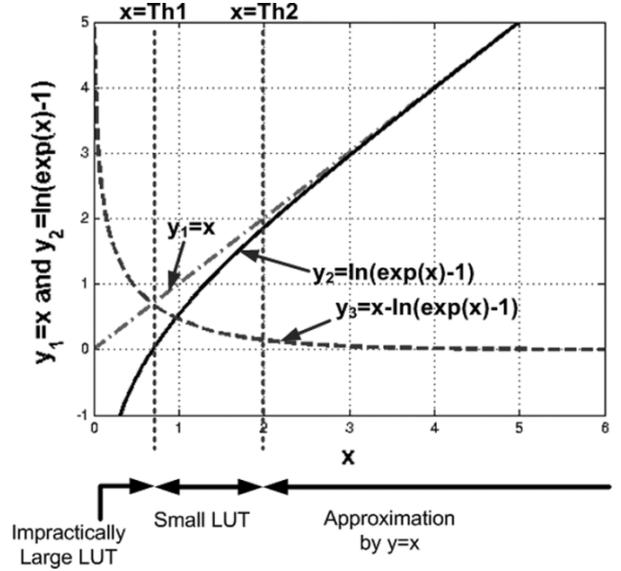
Fig. 3. Butterfly pairs in a W-CDMA turbo encoder trellis diagram.

that is enlarged by attaching a segment on both sides. For example, the processing of β metrics is started from the right segment with giving equal probabilities to all the β 's at the most right position. When the β metric processing reaches the right boundary of the window, the metric values at that position become the initial metric values of the window. Similarly, the initial α metric is obtained by starting from the left segment. The α metrics of the left segment and the β metrics of the right segment are not used in the final computation of extrinsic information. If the segment size is sufficiently long, we can obtain reliable initial metric values. As the sliding window technique can reduce memory size without sacrificing bit error rate (BER) performance significantly, it is regarded as a fundamental technique enabling an efficient implementation of the MAP algorithm.

III. APPROXIMATE REVERSE CALCULATION OF BACKWARD METRICS

A turbo encoder with binary-phase shift keying (BPSK) modulation can be represented by a trellis diagram that has butterfly pairs when the first and the last shift registers are connected in both of the feedback and feed-forward polynomials, as shown in Fig. 2. This is a valid condition for a good RSC encoder [20]. In W-CDMA, four butterfly pairs shown in Fig. 3 are constructed as

$$\left\{ (\beta_k^0, \beta_k^4, \beta_{k+1}^0, \beta_{k+1}^4), (\beta_k^1, \beta_k^5, \beta_{k+1}^1, \beta_{k+1}^5), (\beta_k^2, \beta_k^6, \beta_{k+1}^2, \beta_{k+1}^6), (\beta_k^3, \beta_k^7, \beta_{k+1}^3, \beta_{k+1}^7) \right\}. \quad (7)$$

Fig. 4. Approximation of $\ln(\exp(x) - 1)$.

In (3), each $s_{k+1} \in B$ has two elements as shown in the trellis diagram. The first pair of (7) is represented as

$$\begin{aligned} \ln[\beta_k^0] &= \bar{\beta}_k^0 = \max_{s_{k+1} \in B}^* [\bar{\beta}_{k+1}(s_{k+1}) + \bar{\gamma}_{k+1}(s_k \rightarrow s_{k+1})] \\ &= \max^* [\bar{\beta}_{k+1}^0 + \bar{\gamma}_{k,-1,-1}, \bar{\beta}_{k+1}^1 + \bar{\gamma}_{k,1,1}] \\ &= \ln(e^{\bar{\beta}_{k+1}^0 + \bar{\gamma}_{k,-1,-1}} + e^{\bar{\beta}_{k+1}^1 + \bar{\gamma}_{k,1,1}}) \\ \ln[\beta_k^4] &= \bar{\beta}_k^4 = \max_{s_{k+1} \in B}^* [\bar{\beta}_{k+1}(s_{k+1}) + \bar{\gamma}_{k+1}(s_k \rightarrow s_{k+1})] \\ &= \max^* [\bar{\beta}_{k+1}^0 + \bar{\gamma}_{k,1,1}, \bar{\beta}_{k+1}^1 + \bar{\gamma}_{k,-1,-1}] \\ &= \ln(e^{\bar{\beta}_{k+1}^0 + \bar{\gamma}_{k,1,1}} + e^{\bar{\beta}_{k+1}^1 + \bar{\gamma}_{k,-1,-1}}). \end{aligned} \quad (8)$$

Assuming BPSK modulation and the additive white Gaussian noise (AWGN) channel, the branch metric in log domain is expressed as

$$\bar{\gamma}_{k,d,c} = 0.5 \times \{d(y_s + La) + y_p c\} \quad (9)$$

where k is the time index, y_s is the channel observation of a message bit, y_p is the channel observation of a parity bit, La is *a priori* information and c and d are the parity and message bit anticipated from the trellis diagram, respectively. Since $\bar{\gamma}_{k,d,c}$ has the same value as $-\bar{\gamma}_{k,-d,-c}$, the reverse calculation of (8) can be derived as

$$\begin{aligned} \bar{\beta}_{k+1}^0 &= \ln\left(\frac{e^{\bar{\beta}_k^0 + \bar{\gamma}_{k,-1,-1}} - e^{\bar{\beta}_k^4 - \bar{\gamma}_{k,-1,-1}}}{e^{2\bar{\gamma}_{k,-1,-1}} - e^{-2\bar{\gamma}_{k,-1,-1}}}\right) \\ \bar{\beta}_{k+1}^1 &= \ln\left(\frac{e^{\bar{\beta}_k^4 + \bar{\gamma}_{k,-1,-1}} - e^{\bar{\beta}_k^0 - \bar{\gamma}_{k,-1,-1}}}{e^{2\bar{\gamma}_{k,-1,-1}} - e^{-2\bar{\gamma}_{k,-1,-1}}}\right). \end{aligned} \quad (10)$$

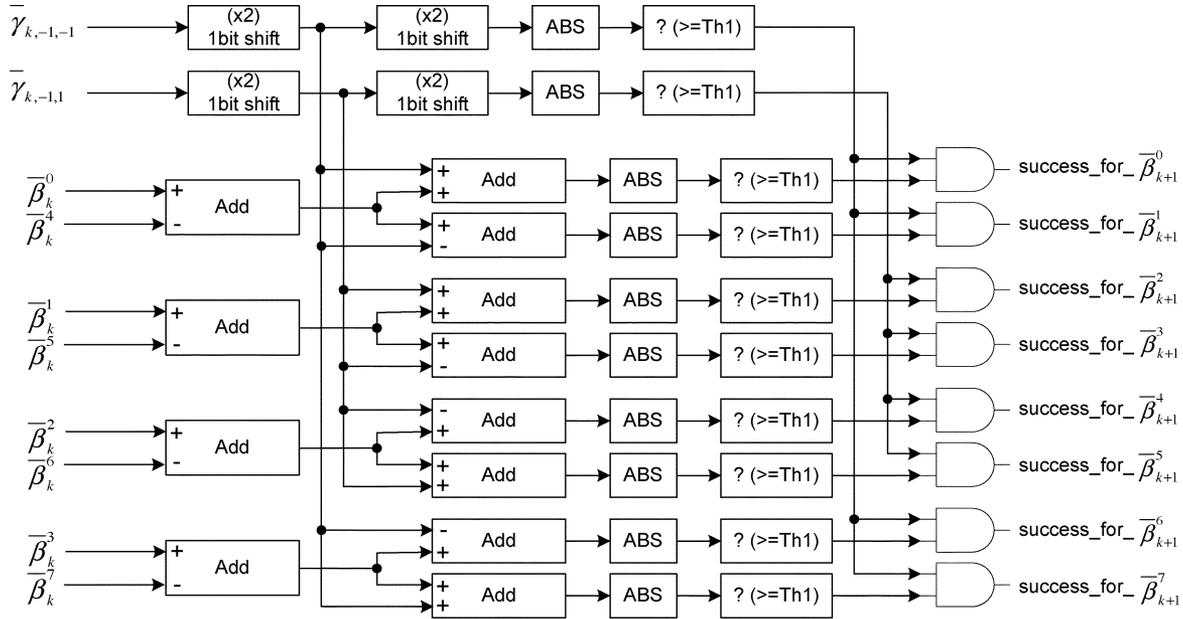


Fig. 5. Approximation checking unit for eight states.

The other butterfly pairs have the same structures as (10) except the superscripts. To achieve a practical implementation, (10) is simplified by using the following modification

$$\ln(|e^x - e^y|) = \min(x, y) + \ln(e^{|x-y|} - 1). \quad (11)$$

When $e^{|x-y|} < Th1$, the second term, $\ln(e^{|x-y|} - 1)$, is on the steep curve as shown in Fig. 4, requiring an impractically large look-up table. On the other hand, when $e^{|x-y|} > Th2$, the second term can be approximated to $|x - y|$. By applying (11), $\bar{\beta}_{k+1}^0$ is rearranged as

$$\begin{aligned} \bar{\beta}_{k+1}^0 &= \ln\left(\frac{e^{\bar{\beta}_k^0 + \bar{\gamma}_{k,-1,-1}} - e^{\bar{\beta}_k^4 - \bar{\gamma}_{k,-1,-1}}}{e^{2\bar{\gamma}_{k,-1,-1}} - e^{-2\bar{\gamma}_{k,-1,-1}}}\right) \\ &= \ln\left(|e^{\bar{\beta}_k^0 + \bar{\gamma}_{k,-1,-1}} - e^{\bar{\beta}_k^4 - \bar{\gamma}_{k,-1,-1}}|\right) \\ &\quad - \ln(|e^{2\bar{\gamma}_{k,-1,-1}} - e^{-2\bar{\gamma}_{k,-1,-1}}|) \\ &= \min(\bar{\beta}_k^0 + \bar{\gamma}_{k,-1,-1}, \bar{\beta}_k^4 - \bar{\gamma}_{k,-1,-1}) \\ &\quad + \ln\left(e^{|\bar{\beta}_k^0 - \bar{\beta}_k^4 + 2\bar{\gamma}_{k,-1,-1}|} - 1\right) \\ &\quad + \left\{ |2\bar{\gamma}_{k,-1,-1}| - \ln\left(e^{|4\bar{\gamma}_{k,-1,-1}|} - 1\right) \right\}. \quad (12) \end{aligned}$$

Based on the graph of Fig. 4, the calculation of $\bar{\beta}_{k+1}^0$ can be classified into the following two cases.

1) $|\bar{\beta}_k^0 - \bar{\beta}_k^4 + 2\bar{\gamma}_{k,-1,-1}| < Th1$ or $|4\bar{\gamma}_{k,-1,-1}| < Th1$: In this case, as $\ln(e^{|x|} - 1)$ is on the steep curve requiring a large-sized look-up table, it is difficult to apply the reverse calculation. The conventional way of storing the backward metrics values into a memory is applied instead of the reverse calculation. During the backward processing, the value of $\bar{\beta}_{k+1}^0$ is used to compute $\bar{\beta}_k$ metrics and stored in the memory. The stored $\bar{\beta}_{k+1}^0$ is retrieved from the memory when it is needed to compute LLR values. In this paper, we call this case an approximation failure.

2) $|\bar{\beta}_k^0 - \bar{\beta}_k^4 + 2\bar{\gamma}_{k,-1,-1}| \geq Th1$ and $|4\bar{\gamma}_{k,-1,-1}| \geq Th1$: In this case, $\bar{\beta}_{k+1}^0$ can be recovered by the reverse calculation. Therefore, $\bar{\beta}_{k+1}^0$ is calculated only for computing $\bar{\beta}_k$ but not stored into the memory. When $\bar{\beta}_{k+1}^0$ is required to compute the output LLR value, the reverse calculation of (12) is used to approximate the value. If the absolute value in this condition is between $Th1$ and $Th2$, the logarithm is approximated by referring to a small look-up table. If the absolute value is equal to or greater than $Th2$, the logarithm value is approximated by the value, i.e., $\ln(e^x - 1) \approx x$. We call this case an approximation success and a metric memory access can be substituted by a reverse calculation.

For the remaining $\bar{\beta}_{k+1}$ metrics, the conditions to decide the cases are equal to the above conditions. Therefore, the case checks required at a time index can be implemented with simple arithmetic operations such as shift and addition. Fig. 5 shows a block diagram of the approximation checking unit designed for the W-CDMA standard.

The values of $Th1$ and $Th2$ should be decided carefully by taking into account quantizing effects and the following criterion:

- 1) An approximation scheme using $Th1$ and $Th2$ should not degrade error-correcting performance considerably.
- 2) As the values of $Th1$ and $Th2$ are lowered, the approximation success rate becomes higher.
- 3) If $Th1$ is equal to $Th2$, the look-up table is not needed, leading to much simpler reverse calculation. If the approximation success rate in case of $Th1 < Th2$ is not higher considerably than the case of $Th1 = Th2$, making $Th1$ equal to $Th2$ can reduce the hardware complexity.

The conventional procedure for computing $\bar{\beta}$ metrics is shown in Fig. 6(a), which requires a memory access for every time index. As opposite to the conventional procedure, the proposed procedure does not store some of the $\bar{\beta}$ metrics if

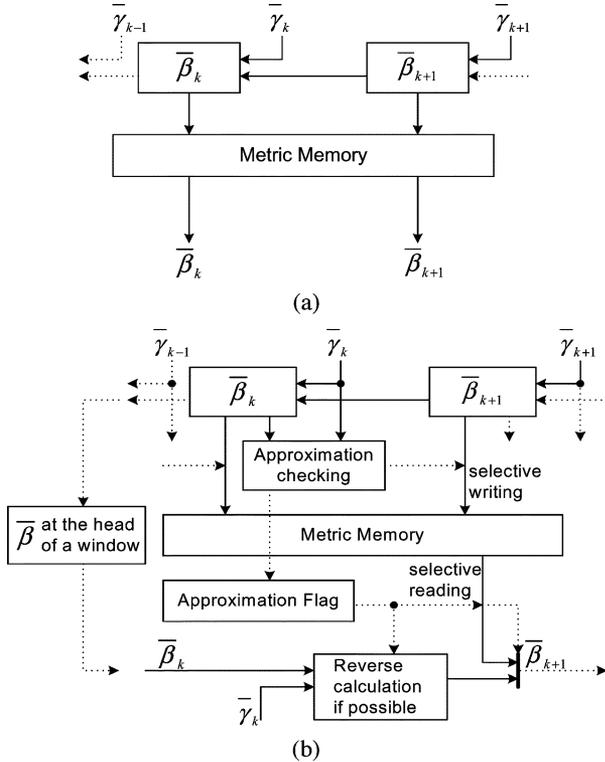


Fig. 6. Decoding procedure for backward metrics. (a) Conventional decoding. (b) Proposed decoding.

they can be recovered by the proposed reverse calculation, as shown in Fig. 6(b). Since only partial $\bar{\beta}$ metrics are stored into the metric memory during the backward processing, we have to know whether the metrics are in the memory or not during the forward processing of LLR values. A straightforward way is to record into a memory whether the approximation check fails or not during the backward processing, and then refer to the memory during the forward processing.

IV. MEMORY OPTIMIZING

As the positions where the approximation can be applied are not deterministic, the metric memory in the proposed decoding procedure must be of the same size as the conventional scheme that stores all the backward metrics. Furthermore, the approximations are successful for some states, not for all the states even at a time index. Since a memory can store multiple data in a memory word, the metric memory structure has to be determined by investigating the pattern of approximation successes.

Fig. 7 describes four possible memory structures designed for W-CDMA that has eight states [24]. The metric memory is partitioned into several banks each of which can be accessed separately. The four memory structures correspond to 1, 2, 4, and 8 banks, and the memory accesses are grouped according to the number of banks. The optimal memory structure, in terms of memory access power, depends on the number of states, memory word access power and the quantizing scheme. To find the optimal memory structure for W-CDMA, four possible memory structures are compared.

In the one-bank memory structure, the memory is accessed if at least one of the eight states fails for approximation. The eight states are divided into two groups, one group of odd states and the other group of even states, and each group is stored into a bank in the two-bank memory structure. If at least one among the odd/even states fails for approximation, the corresponding bank is accessed. In the 4-bank structure, each butterfly is stored into a bank. If one or two states belonging to a butterfly fail for approximation, the corresponding bank is accessed. As each state is stored in a separate bank in the 8-bank memory structure, each bank can be accessed separately for a state failed for approximation.

If a is the memory access rate in case of 8-bank memory structure, 4-bank, 2-bank, and 1-bank memory structures have the memory access rate of a^2 , a^4 , and a^8 , respectively. For a memory access, however, the memory structure with more-than-one banks consumes more power than the one-bank memory if the total memory size is equal to each other. Therefore, a tradeoff between the rate of memory access avoidance and the number of banks should be made to reduce the overall power.

A simple normalizing technique is proposed in [3], [4] to reduce the bit-length of a metric value to be stored in the metric memory. If one metric at a time index is larger than a certain value, all the metrics at the time index are subtracted by the value in the technique. As the memory access and the reverse calculation are mixed even at a time index, it is difficult in the proposed decoding to distinguish whether a metric value recovered by the reverse calculation is normalized or not. Fig. 8 shows an example of normalization errors when $\bar{\beta}$ metrics are quantized to 6 bits. If at least one metric is larger than 31 at a time index, all the four $\bar{\beta}$ metrics at the time index are normalized by subtracting 32. To avoid this error, a normalization flag is used for a time index to record whether the normalization occurs at the time index. When a backward metric is extracted by the reverse calculation, it is denormalized if the corresponding normalization flag is set. The number of normalizing flags is equal to the sliding window size, and independent of the number of states at a time index.

V. EXPERIMENTAL RESULTS

With the WCDMA quantizing scheme shown in Table I [25], the values of $Th1$ and $Th2$ are determined by simulation to achieve a high approximation success rate with considering simple implementation. As shown in Fig. 4, the value of $Th2$ is decided to 2.0 as the maximum reverse calculation error, $2y_3 = 2(x - \ln(e^{Th2} - 1)) \approx 0.2908$, is similar to the maximum quantizing error, 0.25. The value of $Th1$ is initially set to 0.75 because a small look-up table consisting of 4 entries results in ignorable reverse calculation error. However, the value is changed to 2.0 to eliminate the look-up table. For 2-dB signal-to-noise ratio (SNR), the approximation success rate for $Th1 = 0.75$, $Th2 = 2.0$ and $Th1 = Th2 = 2.0$ are 89% and 87%, respectively. Therefore, we decide $Th1 = Th2 = 2.0$, as the values still lead to a high approximation success rate and

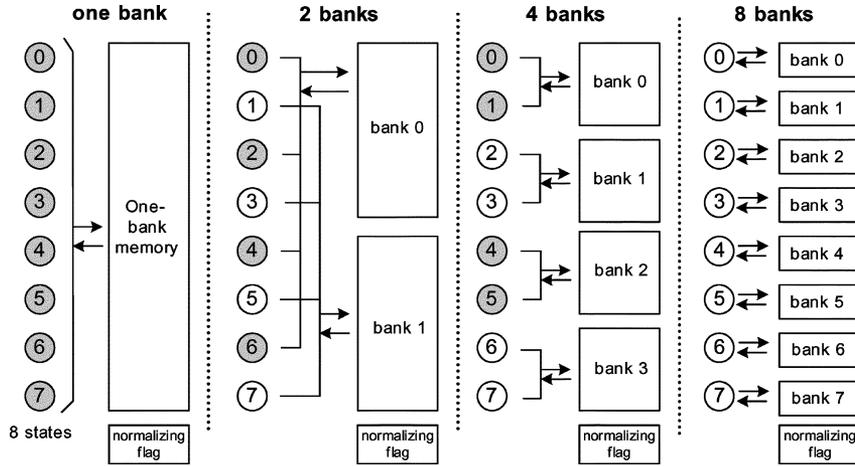


Fig. 7. Banked memory structures for W-CDMA.

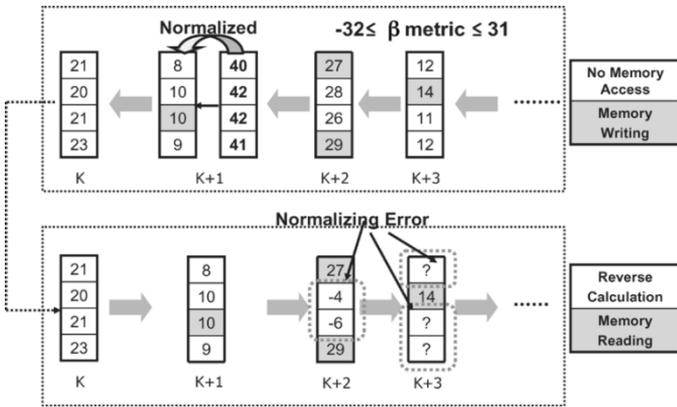


Fig. 8. Normalizing errors caused by the reverse calculation and memory access.

TABLE I
DATA QUANTIZATION FOR TURBO DECODING OF W-CDMA

Data type	Total bit-width	Fractional part
Received inputs	5	2
A priori inputs	8	2
Branch metric	7	2
State metric	9	2
Output LLR	9	2

enable a simpler implementation without degrading error-correcting performance. When $|\bar{\beta}_k^0 - \bar{\beta}_k^4 + 2\bar{\gamma}_{k,-1,-1}| \geq 2.0$ and $|4\bar{\gamma}_{k,-1,-1}| \geq 2.0$, (12) is approximated as

$$\bar{\beta}_{k+1}^0 \approx \max(\bar{\beta}_k^0 + \bar{\gamma}_{k,-1,-1}^0, \bar{\beta}_k^4 - \bar{\gamma}_{k,-1,-1}^0) - |2\bar{\gamma}_{k,-1,-1}^0|. \quad (13)$$

For the W-CDMA standard, the BER and frame-error rate (FER) performances associated with this approximation scheme are shown in Fig. 9, which are obtained by using AWGN channel, ten iterations, 1024 Frame size, and 32 sliding

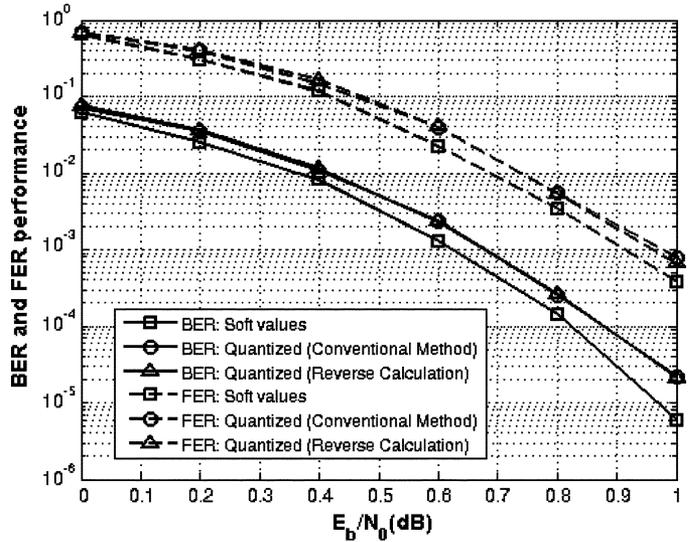


Fig. 9. BER and FER performance with W-CDMA standard, AWGN channel, ten iterations, 1024 frame size and 32 sliding window size.

window size. The BER and FER degradation caused by the reverse calculation is negligible because the error resulting from the reverse calculation is less than the maximum quantizing error in most cases.

Two block diagrams of the reverse calculation unit are shown in Fig. 10(a) and (b), which are basically derived from (12) and (13) for state transitions of $s_0 \rightarrow s$ and $s_1 \rightarrow s$, where s is a state of time index $k + 1$ and γ denotes a branch metric value of the $s_0 \rightarrow s$ transition. The reverse calculation unit is similar to the Add-Compare-Select-Offset (ACSO) unit except that the look-up table is replaced by $y = x$ when $x \geq Th_2$.

To determine the optimized memory structure, simulations are conducted. Fig. 11 shows the rate of approximation success or the rate of memory access avoidance plotted for the 8-bank memory structure, which is obtained with 4, 6, 8, and 10 fixed iterations for a range of SNRs. As indicated in the Fig. 11, the rate of memory access avoidance improves according to the number of decoding iterations, but not rapidly. About 87% memory accesses are replaced with the reverse calculations by employing

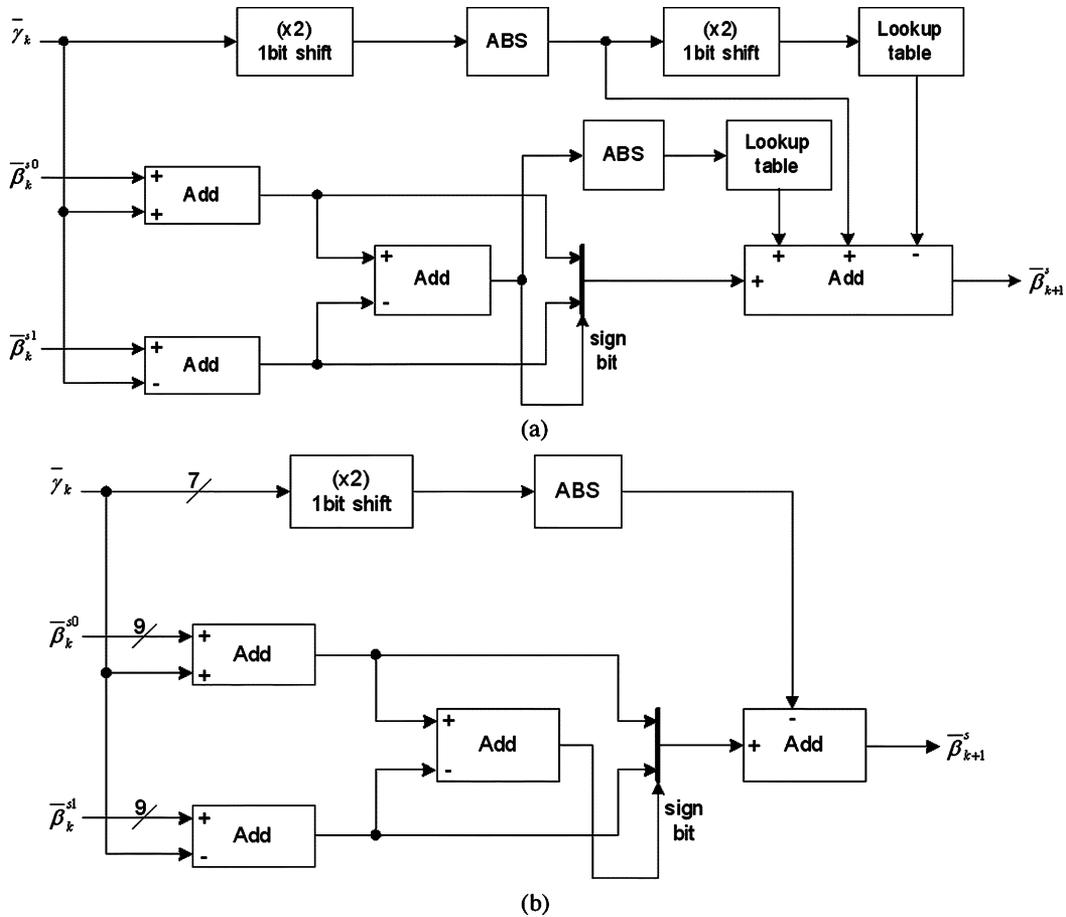


Fig. 10. Reverse calculation unit. (a) Implementation of equation (12). (b) Implementation of equation (13).

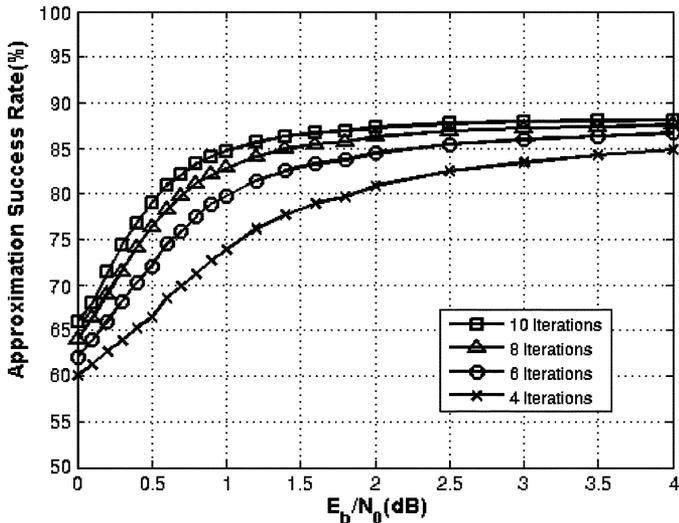


Fig. 11. Approximation success rate versus SNR (in decibels).

the 8-bank memory structure and 10 fixed iterations. Notice that the rate of memory access avoidance for the 8-bank memory structure is not decreased seriously even for 0.0-dB SNR.

From the read/write power consumption of the banked memory structures and the memory access rates, the average memory access power is obtained by using compiled memories

TABLE II
POWER COMPARISON OF β MEMORY ACCESS MEASURED AT 1 MHz FOR 2-dB SNR AND TEN ITERATIONS

Decoder	β memory configuration	No. of banks	Power of (read+write) for 72 bits	Memory access rate	Memory power
Conventional	(32x8x9)	1	818.7 μ W	1.00	818.7 μ W (100%)
	(32x8x9)	1	818.7 μ W	0.663	542.8 μ W (66.3%)
Proposed	(32x4x9) x 2	2	858.9 μ W	0.419	359.9 μ W (44.0%)
	(32x2x9) x 4	4	939.4 μ W	0.238	223.6 μ W (27.3%)
	(32x1x9) x 8	8	1100.2 μ W	0.127	139.7 μ W (17.1%)

designed in 0.25- μ m technology. Table II shows the powers of β metric memory consumed in the conventional decoder and the proposed decoder, which are obtained with a sliding window size of 32. In the proposed decoder, four different memory structures are considered for the β metric memory. The power consumed to access one word of 72 bits is also indicated in Table II, where 72 bits are required to store 8 backward metrics quantized to 9 bits. Though the 8-bank memory structure leads to the largest power consumption for a memory access, the resulting low memory access rate dramatically reduces the average β memory power consumption to 17.1% compared to a conventional decoder that stores all the β metrics.

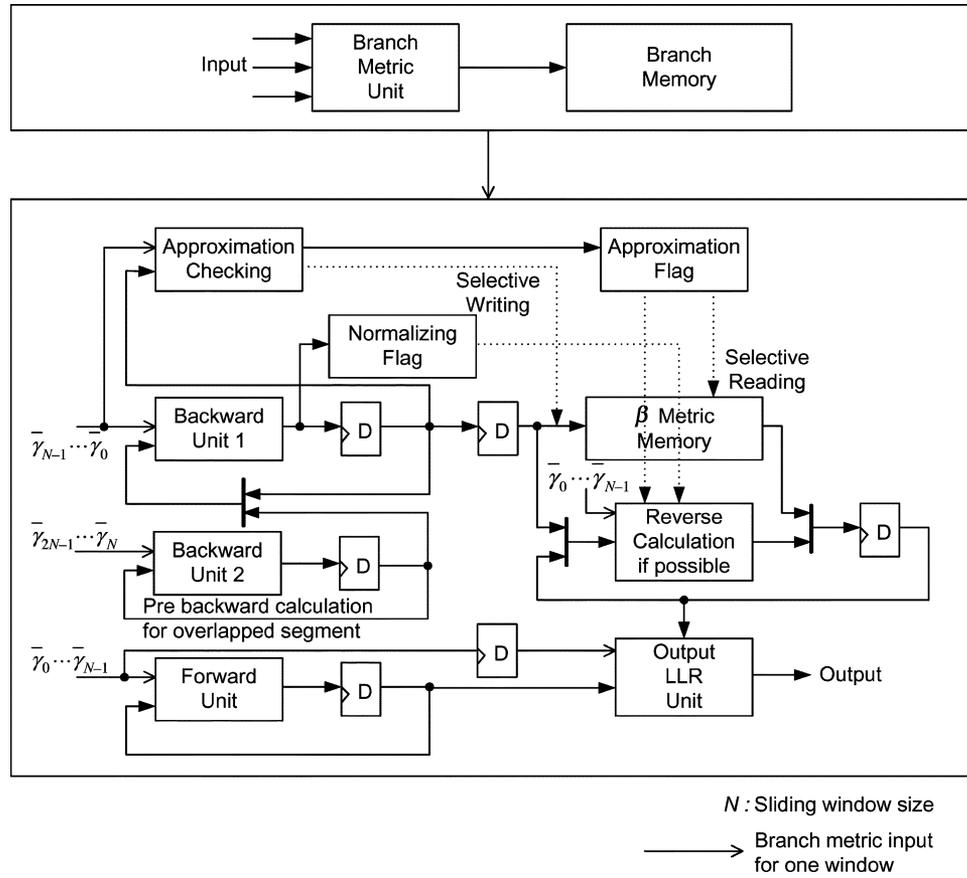


Fig. 12. Block diagram of the proposed log-MAP decoder.

The proposed log-MAP decoder was described in Verilog-HDL and synthesized with a 0.25- μm standard-cell library and compiled SRAM memories. The design is fully synchronized with a single clock to achieve easy timing verification. Design Compiler and DesignPower of Synopsys were used for the synthesis and power estimation, respectively. Switching activities resulting from gate-level simulation are annotated for gate-level power estimation. The proposed decoder drawn in Fig. 12 is compared with a conventional log-MAP decoder that accesses the β memory always, as summarized in Table III. In the proposed decoder, an approximation check unit, a reverse calculation unit of β metrics, normalization flags and approximation flags are added to the conventional decoder. One of two β metric calculation units (Backward Unit 2) is used to decide the initial β metric values. The branch metric input stream is fed into several blocks to decode a window. During the forward processing of LLR values, the approximation flag unit shown at the right-hand side decides whether the approximate reverse calculation is applicable for the next time index.

The rate of the β memory access power to the total power consumption is significantly reduced. If the number of iterations increases, the rate can be reduced more as the approximation success rate improves. In the proposed structure, therefore, more attention is paid to the power and delay optimization of logic modules. The delay optimization is required to compensate the control delay overhead caused by the proposed reverse calculation. The control delay overhead, however, can be hidden as the

TABLE III
ENERGY CONSUMPTION COMPARISON OF LOG-MAP DECODERS FOR 2-dB SNR AND 10 ITERATIONS

Component	Conventional Log-MAP decoder	Proposed Log-MAP decoder
Branch Memory	375.6 pJ/bit/iter (21.7%)	375.6 pJ/bit/iter (32.8%)
Beta Memory	818.7 pJ/bit/iter (47.3%)	139.7 pJ/bit/iter (8.8%)
SISO Module (+flags)	536.8 pJ/bit/iter (31.0%)	715.2 pJ/bit/iter (58.1%)
Total	1.731 nJ/bit/iter	1.231 nJ/bit/iter
Normalized Power	100%	71.1%

approximation check is separated from β metric update in the backward processing, and the delay in the forward processing of backward metrics is shorter than the computational delay of LLR values.

Table IV shows that the SISO module of a conventional decoder consists of 18 705 gates, while the module increases to 24 522 gates in the proposed decoder to include additional hardware blocks such as normalizing flags and approximation checking units. In the proposed decoder, the metric memory is partitioned to 8 banks and registers are inserted to hide the control delay overhead. Both of the decoders achieve the critical delay of 10.4 ns. As a result, the proposed log-MAP decoder can be operated at approximately 95 MHz, which is sufficient to the W-CDMA standard specification of 2 Mbps. In the favorable situation associated with high SNR and a large number of iterations, the proposed decoding procedure

TABLE IV
COMPARISON OF AREA AND CRITICAL PATH DELAY

	Conventional Log-MAP decoder	Proposed Log-MAP decoder
Metric Memory Size	Dual Port SRAM 2304bits	Dual Port SRAM 288bits x 8
Logic Gates	18705	24522
Critical Path Delay	10.43ns	10.43ns

consumes less power because of the improved approximation success rate, while the conventional decoder consumes a fixed power.

VI. CONCLUSION

This paper has presented an approximate reverse calculation to reduce the number of backward metric memory accesses required in turbo decoding. As turbo codes have butterfly structures in their trellis diagram that can be calculated in reverse, the reverse calculation is possible. Since the reverse calculation is complex and has singular points, however, it has not been used in real implementations. We have proposed in this paper a new approximation that can be implemented with simple arithmetic operations and small look-up tables, and applied the proposed reverse calculation to the backward metric processing.

Experimental results show that 87% of backward metric memory accesses can be substituted by the proposed approximate reverse calculation if the metric memory structure is organized suitably. Applying the approximate reverse calculation changes the power bottleneck from the memory accesses to the logic modules. Although addition control overhead is inevitable to decide whether the approximation is applicable or not, about 29% power consumption is reduced in a log-MAP decoder employing the proposed approximate reverse calculation. As the approximate reverse calculation is not directly related to other low power techniques such as voltage scaling and early termination, they can be combined to achieve further power reduction.

ACKNOWLEDGMENT

The authors would like to thank J.-Y. Kwak and M.-C. Shin of Samsung Electronics for the very valuable comments and suggestions.

REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error correcting coding and decoding: turbo codes," in *Proc. Int. Conf. Commun.*, May 1993, pp. 1064–1070.
- [2] G. Maserà, G. Piccinini, M. R. Roch, and M. Zamboni, "VLSI architectures for turbo-codes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 7, no. 3, pp. 369–379, Sep. 1999.
- [3] Z. Wang, "High Performance, Low Complexity VLSI Design of Turbo Decoders," Ph. D dissertation, ECE Dept., Univ. of Minnesota, Twin Cities, 2000.

- [4] Z. Wang, H. Suzuki, and K. K. Parhi, "VLSI implementation issues of turbo decoder design for wireless applications," in *Proc. IEEE Int. Workshop Signal Process. Syst.*, 1999, pp. 503–512.
- [5] J. Vogt, K. Koorä, A. Finger, and G. Gettweis, "Comparison of different turbo decoder realization for IMT-2000," in *Proc. Globecom*, 1999, pp. 2704–2708.
- [6] M. A. Bickerstaff, L. M. Davis, C. Thomas, D. Garrett, and C. Nicol, "A 24 Mb/s radix-4 logMAP Turbo decoder for 3GPP-HSDPA mobile wireless," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2003, pp. 150–151.
- [7] V. C. Gaudet and P. G. Gulak, "A 13.3-Mb/s 0.35- μ m CMOS analog turbo decoder IC with a configurable interleaver," *IEEE J. Solid-State Circuits*, vol. 38, no. 11, pp. 2010–2015, Nov. 2003.
- [8] B. Bougard, A. Giulietti, V. Derudder, J. W. Weijers, S. Dupont, L. Hollevoet, F. Catthoor, L. V. der Perre, H. De Man, and R. Lauwercins, "A scalable 8.7 nJ/bit 75.6 Mb/s parallel concatenated convolutional (turbo-) codec," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2003, pp. 152–153.
- [9] M. Bickerstaff, D. Garrett, T. Prokop, C. Thomas, B. Widdup, G. Zhou, C. Nicol, and R.-H. Yan, "A unified turbo/Viterbi channel decoder for 3GPP mobile wireless in 0.18 μ m CMOS," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2002, pp. 124–125.
- [10] Y. Wu, B. D. Woerner, and W. J. Ebel, "A simple stopping criterion for turbo decoding," *IEEE Commun. Lett.*, vol. 4, no. 4, pp. 258–260, Aug. 2000.
- [11] A. J. Viterbi, "An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 260–264, Feb. 1998.
- [12] D. Garrett, B. Xu, and C. Nicol, "Energy efficient turbo decoding for 3G mobile," in *Proc. IEEE Int. Symp. Low Power Electronics Design (ISLPED)*, 2001, pp. 328–333.
- [13] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. New York: Wiley, 1999.
- [14] R. Gonzales, B. M. Gordon, and M. A. Horowitz, "Supply and threshold voltage scaling for low power CMOS," *IEEE J. Solid-State Circuits*, vol. 32, no. 8, pp. 1210–1216, Aug. 1997.
- [15] G. Maserà, M. Mazza, G. Piccinini, F. Viglione, and M. Zamboni, "Architectural strategies for low-power VLSI turbo decoders," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 3, pp. 279–285, Jun. 2002.
- [16] S. J. Lee, N. R. Shanbhan, and A. C. Singer, "A low-power VLSI architecture for turbo decoding," in *Proc. IEEE Int. Symp. Low Power Electron. Des. (ISLPED)*, 2003, pp. 366–371.
- [17] C. Schurgers, F. Catthoor, and M. Engels, "Memory optimization of MAP turbo decoder algorithms," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 2, pp. 305–312, Apr. 2001.
- [18] M. C. Shin and I. C. Park, "A programmable turbo decoder for multiple third-generation wireless standards," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2003, pp. 154–155.
- [19] J. Kwak, S. M. Park, and K. Lee, "Reverse tracing of forward state metric in log-MAP and MAX-log-MAP decoders," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2003, vol. 2, pp. 25–28.
- [20] Y. Wu, W. J. Ebel, and B. D. Woerner, "Forward computation of backward path metrics for MAP decoders," in *Proc. VTC*, 2000, pp. 2257–2261.
- [21] F. Catthoor, S. Wuytack, E. De Greef, F. Balasa, L. Nachtergaele, and A. Vandecappelle, *Custom Memory Management Methodology: Exploration of Memory Organization for Embedded Multimedia System Design*. Norwell, MA: Kluwer, 1998.
- [22] T. Okuma, Y. Cao, M. Muroyama, and H. Yasuura, "Reducing access energy of on-chip data memory considering active data bitwidth," in *Proc. IEEE Int. Symp. Low Power Electron. Des. (ISLPED)*, 2002, pp. 88–91.
- [23] P. Robertson, E. Villebrun, and P. P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. IEEE Int. Conf. Commun.*, Jun. 1995, pp. 1009–1013.
- [24] Technical Specification Group Radio Access Network, The 3rd generation partnership project (3GPP) 2005 [Online]. Available: <http://www.3gpp.org>
- [25] H. Michel and N. When, "Turbo-decoder quantization for UMTS," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 55–57, Feb. 2001.



Dong-Soo Lee (S'02–M'05) received the B.S. and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2002 and 2004, respectively.

In 2004, he joined Samsung Electronics Co., Ltd., Suwon, Korea, where he has been involved in designing circuits for DTV one-chip-solution. His research interests include low power design, digital signal processing, and system-on-chip designs.



In-Cheol Park (S'88–M'92–SM'02) received the B.S. degree in electronic engineering from Seoul National University, Seoul, Korea, in 1986, the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1988 and 1992, respectively.

Since June 1996, he has been an Assistant Professor and is now a Professor in the Department of Electrical Engineering and Computer Science at KAIST. Prior to joining KAIST, he was with IBM T. J. Watson Research Center, Yorktown, NY, from May 1995 to May 1996, where he researched on high-speed circuit design. His current research interest includes CAD algorithms for high-level synthesis and VLSI architectures for general-purpose microprocessors. He is a senior member of the IEEE.

Prof. Park received the Best Paper Award at ICCD in 1999, and the Best Design Award at ASP-DAC in 1997.