LOW-POWER LOG-MAP TURBO DECODING BASED ON REDUCED METRIC MEMORY ACCESS

*Dong-Soo Lee and In-Cheol Park**

SOC R&D Center, System LSI, Samsung Electronics, Korea
*Department of Electrical Engineering and Computer Science, KAIST, Korea
E-mail address: dongsoo3.lee@samsung.com

ABSTRACT

Due to the powerful error correcting performance, turbo codes have been adopted in many wireless communication standards. 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 of backward metrics which can be implemented with simple computational complexity. Simulation results show that the proposed method applied to W-CDMA standard reduces the access rate of the backward metric memory by 90% without degrading error correcting performance. A prototype turbo decoder based on the proposed reverse calculation achieves 30% power reduction compared to the conventional decoder.

1. INTRODUCTION

Since turbo coding was introduced by Berrou *et al.* 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 third-generation mobile radio systems such as W-CDMA and CDMA 2000, and various studies have focused on their practical implementations [2][3]. A turbo decoder consists of two decoding components 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.

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 greatly. It has been reported that the memory access power accounts for more than 50% of the entire power consumption [4]. A complex address generation algorithm for the interleaving is implemented on-the-fly instead of storing the interleaver addresses in a table [5]. The partial metric storage method proposed in [4] replaces some parts of the metric memory to 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 to up to a quarter of the memory size.

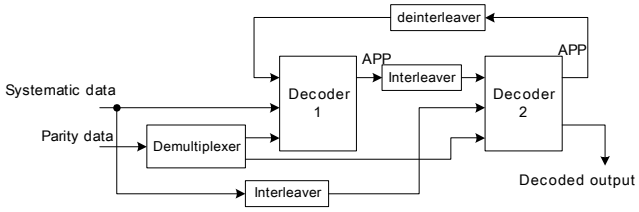

Figure 1. Structure of a turbo decoder

Reverse calculation of state metrics is another efficient method to reduce memory accesses as reported in [6][7], which demonstrated 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 need to access a memory is greater than that of the corresponding computation, which is usually valid for today's deep submicron technology [8]. However, the quantization and singular matrix problems are not solved in [6], and the modifications introduced to solve these problems are not efficient in terms of real applications [7].

In this paper, we propose a new approximate reverse calculation for backward metrics. In the process of backward metrics calculation, about 10% of the calculated metrics corresponding to singular matrix calculations are written into a memory and the others are not saved because they can be recovered by the reverse calculation. When backward metrics are needed, they are read from memory or recovered by the proposed reverse calculation.

2. LOG-MAP ALGORITHM

The turbo decoding structure consists of two soft-input, soft-out (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, the output for the k^{th} symbol is expressed in log-likelihood ratio (LLR) form as

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

, where s_k represents a state of the encoder at time k, and $s_k \to s_{k+1}$ is the state transition from state s_k to state s_{k+1}, and $s0$ and $s1$ denote the set of all the possible state transitions associated with message bit 0 and 1, respectively.

To simplify the calculation of α and β metrics, the Jacobian logarithm is applied to produce the following equations (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 :

$$\ln\left[\alpha_k(s_k)\right] = \overline{\alpha}_k(s_k) = \max_{s_{k-1}\in A}{}^{*}\left[\overline{\alpha}_{k-1}(s_{k-1}) + \overline{\gamma}_k(s_{k-1} \to s_k)\right] \quad (2)$$

$$\ln\left[\beta_k(s_k)\right] = \overline{\beta}_k(s_k) = \max_{s_{k+1}\in B}{}^{*}\left[\overline{\beta}_{k+1}(s_{k+1}) + \overline{\gamma}_{k+1}(s_k \to s_{k+1})\right] \quad (3)$$

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

$$\ln\left[\gamma_k(s_k \to s_{k+1})\right] = \overline{\gamma}_k(s_k \to s_{k+1}) = \ln\left[P(\mathbf{y}_k \mid \mathbf{x}_k)\cdot P(u_k)\right] \quad (5)$$

, where u_k is the input bit necessary to cause 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

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

As indicated in equations (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 equation (6). In this paper, we assume that β metrics are calculated prior to α metrics.

3. APPROXIMATE REVERSE CALCULATION

A turbo encoder with BPSK modulation can be represented by a trellis that has butterfly pairs when the first and the last shift registers are connected in both of the feedback and feed-forward polynomials. This is a valid condition for a good RSC encoder [6]. In W-CDMA, four butterfly pairs shown in Fig. 2 are constructed as

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

The first pair is represented as

$$\begin{aligned}\overline{\beta}_k^0 &= \ln\left(e^{\overline{\beta}_{k+1}^0 + \overline{\gamma}_{k,-1,-1}} + e^{\overline{\beta}_{k+1}^1 + \overline{\gamma}_{k,1,1}}\right) \\ \overline{\beta}_k^4 &= \ln\left(e^{\overline{\beta}_{k+1}^0 + \overline{\gamma}_{k,1,1}} + e^{\overline{\beta}_{k+1}^1 + \overline{\gamma}_{k,-1,-1}}\right)\end{aligned} \quad (8)$$

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

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

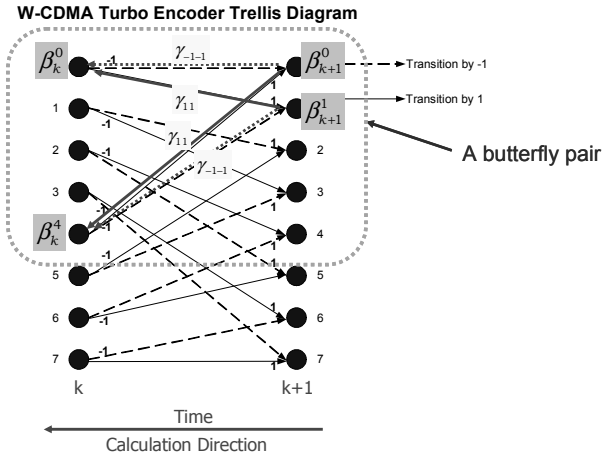

Figure 2. Butterfly pairs in a W-CDMA turbo encoder trellis diagram

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

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

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

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

When $e^{|x-y|} < 2$, the second term, $\ln\left(e^{|x-y|} - 1\right)$, is on the steep curve as shown in Fig. 3, requiring an impractically large lookup table. On the other hand, when $e^{|x-y|} >> 2$, the second term can be approximated to $|x-y|$. By applying equation (11), $\overline{\beta}_{k+1}^0$ is rearranged as

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

Based on the graph of Fig. 3, the calculation of $\overline{\beta}_{k+1}^0$ can be classified into the following two cases.

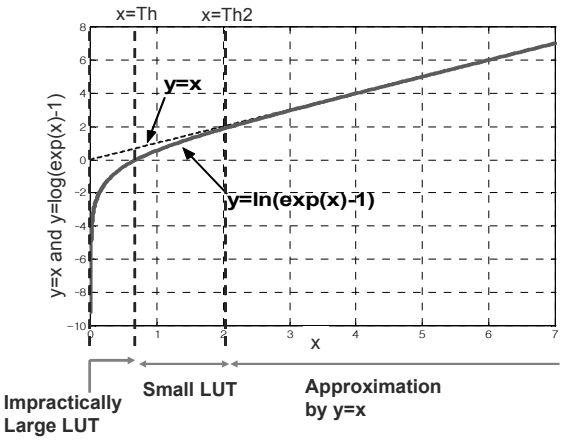

Figure 3. Approximation of ln(exp(x)-1)

1) $\left|\overline{\beta}_k^0 - \overline{\beta}_k^4 + 2\overline{\gamma}_{k,-1,-1}\right| < \ln 2$ *or* $\left|4\overline{\gamma}_{k,-1,-1}\right| < \ln 2$.

In this case, as $\ln\left(e^{|\bullet|}-1\right)$ is on the steep curve requiring a large-sized lookup table, it is difficult to apply the reverse calculation for $\overline{\beta}_{k+1}^0$. 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 $\overline{\beta}_{k+1}^0$ is used to compute $\overline{\beta}_k$ metrics and stored in the memory. The stored $\overline{\beta}_{k+1}^0$ is retrieved from the memory when it is needed to compute LLR values.

2) $\left|\overline{\beta}_k^0 - \overline{\beta}_k^4 + 2\overline{\gamma}_{k,-1,-1}\right| \geq \ln 2$ *and* $\left|4\overline{\gamma}_{k,-1,-1}\right| \geq \ln 2$.

In this case, $\overline{\beta}_{k+1}^0$ metric is calculated only for computing $\overline{\beta}_k$ metric but not stored into the memory. When $\overline{\beta}_{k+1}^0$ metric is required to compute the output LLR value, the reverse calculation of equation (12) is used to approximate the value. If the absolute value in this condition is between *Th*, a quantized value of $\ln 2$, and *Th*2, the logarithm is approximated by referring a small lookup table, as shown in Fig. 3. If the absolute value is equal or greater than *Th*2, the logarithm value is approximated by the value, i.e., $\ln\left(e^x - 1\right) \approx x$.

For the remaining $\overline{\beta}_{k+1}$ 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. Since only partial $\overline{\beta}$ 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. An approximation flag is used for a time index to record whether approximation is possible at the time index. If the corresponding approximation flag is set, a backward metric is extracted by the reverse calculation. The number of approximation flags is equal to the sliding window size, and each flag has the bit width of the number of states at a time index.

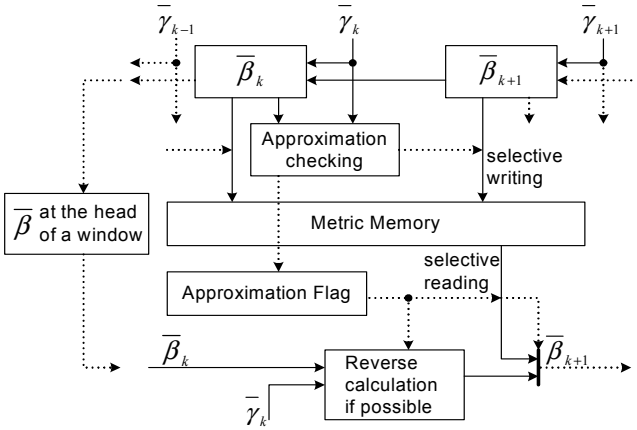

Figure 4. Proposed decoding procedure for backward metrics

4. MEMORY OPTIMIZING

The proposed decoding procedure for backward metrics is shown in Fig. 4. As the positions where the approximation can be applied are random, 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 optimized by investigating the pattern of approximation successes, the case that the two absolute values are equal or greater than *Th*.

The optimal memory structure depends on the number of states and the butterfly pairs. 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.

To determine the optimized structure, simulations are conducted with the quantized scheme of [3]. In the simulations, *Th* and *Th*2 are set to 0.75 and 2.0, respectively, because the values result in a negligible degradation of error correcting performance that is with in the quantizing error. Fig. 5 shows the rate of approximation success plotted for each memory structure, which is obtained with 8 fixed iterations. As indicated in the Fig. 5, more memory banks result in a higher rate of approximation success. The rate of approximation success improves according to the number of decoding iterations, but not rapidly. Table. 1 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 and β metric quantization to 9 bits.

5. EXPERIMENTAL RESULTS

The proposed log-MAP decoder was described in Verilog-HDL and synthesized by a 0.25μm standard-cell library and compiled SRAM memories. Design Compiler and DesignPower of Synopsys were used for the synthesis and power estimation, respectively. The proposed decoder is compared with a conventional log-MAP decoder, as summarized in Table. 2.

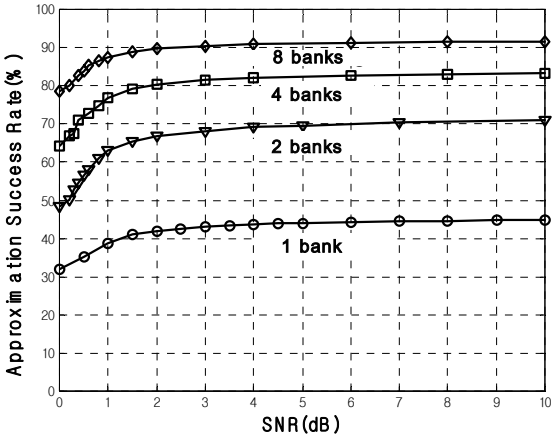

Figure 5. Approximation success rate versus SNR(dB)

Note that the rate of the β memory access power to the total power consumption is significantly reduced. In the proposed structure, therefore, more attention is paid to the power and delay optimization of logic modules. The SISO module of a conventional decoder consists of 18705 gates, while the module increases to 24850 gates in the proposed decoder. In the proposed decoder, the metric memory is partitioned to 8-banked memories with the same size and registers are inserted to hide the control delay overhead. Both of the decoders achieves the critical delay of 10.43ns. As a result, the proposed log-MAP decoder can be operated at approximately 95MHz, which meets the W-CDMA standard specification of 2Mbps. In the favorable situation associated with high SNR and a large number of iterations, the proposed decoding procedure consumes less power because of the improved rate of approximation success, while the conventional decoder consumes a fixed power.

TABLE 1. Power comparison of β memory access measured at 1MHz for 2dB SNR and 8 iterations

Decoder	β memory configuration	Power of (read+write) for 72 bits	Memory access rate	Memory power
Conventional	(32x8x9)	818.7 μW	1.00	818.7 μW
Proposed	(32x8x9)	818.7 μW	0.57	466.7 μW
	(32x4x9) x 2	858.9 μW	0.29	249.1 μW
	(32x2x9) x 4	939.4 μW	0.18	169.1 μW
	(32x1x9) x 8	1100.2 μW	0.10	110.0 μW

TABLE 2. Power comparison of turbo decoders per MHz

Component	Conventional Log-MAP decoder	Proposed Log-MAP decoder
Branch Memory	375.6 μW (21.7%)	375.6 μW (31.2%)
Beta Memory	818.7 μW (47.3%)	110.0 μW (9.1%)
SISO Module (+flags)	536.8 μW (31.0%)	719.8 μW (59.7%)
Total	1731.1 μW	1205.4 μW
Normalized Power	100%	69.6%

6. CONCLUSION

This paper has presented an approximate reverse calculation to reduce the backward metric memory accesses required in turbo decoding. We save only a small portion of the backward metrics, not all the backward metrics, that cannot be computed by using the proposed approximate reverse calculation. The other backward metrics are not saved but recovered by using the proposed reverse calculation when they are needed in the forwarding process of LLR values. Experimental results show that in the W-CDMA standard 90% of backward metric memory accesses can be substituted by the reverse calculations if the metric memory is organized suitably. At the expense of small logic overhead for the decision, about 30% power consumption is reduced in a MAP decoder.

7. ACKNOWLEDGEMENT

This research was supported in part by University IT Research Center Project and by Korea Science Engineering Foundation through MICROS center.

8. REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error correcting coding and decoding: Turbo codes," in *Proc. ICC*, pp. 1064-1070, May 1993.

[2] G. Masera, G. Piccinini, M. R. Roch, and M. Zamboni, "VLSI architectures for turbo-codes", *IEEE Trans. VLSI Syst.,* vol. 7, Sep. 1999, pp. 369-379.

[3] Z. Wang, H. Suzuki, and K. K. Parhi, "VLSI implementation issues of turbo decoder design for wireless applications," in *Proc. IEEE SiPS,* pp. 503-512, 1999.

[4] C. Schurgers, F. Catthoor, and M. Engels, "Memory optimization of MAP turbo decoder algorithms," *IEEE Trans. VLSI Syst.,* vol. 9, April 2001, pp. 305-312.

[5] D. Garrett, Bing Xu, and Chris Nicol, "Energy efficient turbo decoding for 3G mobile", in *Proc. of IEEE ISLPED,* pp. 328-333, 2001.

[6] Y. Wu, W. J. Ebel, and B. D. Woerner, "Forward computation of backward path metrics for MAP decoders," in *Proc. of VTC,* pp. 2257-2261, 2000.

[7] J. Kwak, S. M. Park, and K. Lee, "Reverse tracing of forward state metric in log-MAP and MAX-log-MAP decoders," in *Proc. of IEEE ISCAS,* pp. 25-28, 2003.

[8] F. Catthoor et al., *Custom Memory Management Methodology: Exploration of Memory Organization for Embedded Multimedia System Design.* Norwell. MA: Kluwer, 1998.