Low-Power Hybrid Turbo Decoding Based on Reverse Calculation

Hye-Mi Choi, Ji-Hoon Kim, and In-Cheol Park
Department of Electrical Engineering and Computer Science
Korea Advanced Institute of Science and Technology (KAIST)
373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea
{hmchoi, jhkim, icpark}@ics.kaist.ac.kr

Abstract—As **turbo decoding is a highly memory-intensive algorithm consuming large power, a major issue to be solved in practical implementation is to reduce power consumption. This paper presents an efficient reverse calculation method to lower the power consumption by reducing the number of memory accesses required in turbo decoding. The reverse calculation method is proposed for the Max-log-MAP algorithm, and it is combined with a scaling technique to achieve a new decoding algorithm, called hybrid log-MAP, that results in a similar BER performance to the log-MAP algorithm. For the W-CDMA standard, experimental results show that 80% of memory accesses are reduced through the proposed reverse calculation method. A hybrid log-MAP turbo decoder based on the proposed reverse calculation reduces power consumption and memory size by 34.4% and 39.2%, respectively.**

I. INTRODUCTION

Turbo coding introduced in 1993 [1] has been reported that the turbo code can approach the Shannon's limit within a few tenths dB with moderate complexity. Having this remarkable performance, the turbo code has found its way into many standardized systems such as W-CDMA and CDMA2000. As turbo decoding is a highly memory-intensive algorithm, however, a significant amount of power is consumed for memory accesses, resulting in a power bottleneck that limits the incorporation of turbo decoders in commercial hand-held systems. To overcome this problem, several low-power techniques have been proposed [2]-[6]. One of the efficient methods is the reverse calculation that replaces memory accesses with additional computations. Y. Wu *et al.* first proposed the reverse calculation of state metrics [4], but did not describe implementation issues because their reverse calculation requires impractically large lookup tables. In [5], a loser storing method for the Max-log-MAP algorithm was proposed. In the loser storing method, only the losers are stored into a memory while the winners are recovered by applying the reverse calculation. A practical reverse calculation method was proposed for the log-MAP algorithm along with its implementation in [6]. However, the implementation is relatively complex and requires a large logic overhead.

This paper proposes an efficient reverse calculation method to reduce not only memory power consumption but also memory size. The proposed reverse calculation method is based on the Max-log-MAP algorithm, and it is combined with a scaling technique to achieve a new decoding algorithm called hybrid log-MAP. Compared to the loser storing method that always accesses the memory to decide the position and the value of the loser, the proposed reverse calculation method stores and accesses the loser selectively only when the loser cannot be recovered by the reverse calculation.

II. DECODING ALGORITHMS FOR TURBO CODES

Turbo coding requires SISO decoders to generate extrinsic information and the log-likelihood ratio. Either the maximum a posteriori (MAP) algorithm [7] or the soft output Viterbi algorithm (SOVA) [8] can be used for SISO decoding. As MAP-based turbo decoders provide much better performance than SOVA-based turbo decoders, we focus on the MAP-based algorithm in this work.

A. Max-log-MAP algorithm

The Max-log-MAP decoder decides whether an information bit $u_k = 0$ or 1 depending on the log-likelihood ratio $L_R(u_k)$ which is defined below.

$$L_R(u_k) = \max_{U^1}[\alpha_{k-1}(s_{k-1}) + \gamma_k(s_{k-1} \to s_k) + \beta_k(s_k)] - \max_{U^0}[\alpha_{k-1}(s_{k-1}) + \gamma_k(s_{k-1} \to s_k) + \beta_k(s_k)] \tag{1}$$

where s_k is an encoder state at time *k*, U^1 is the set of all possible state transitions $(s_{k-1} \to s_k)$ corresponding to the case of $u_k = 1$, and U^0 is similarly defined. If $L_R(u_k)$ is greater than or equal to zero, u_k is decided to 1. Otherwise, u_k is decided to 0. In (1), $\alpha_{k-1}(s_{k-1})$, $\beta_k(s_k)$, and $\gamma_k(s_{k-1} \to s_k)$ are the forward, backward, and branch metrics, respectively. The metrics are calculated as

$$\alpha_k(s_k) = \max_{S_{k-1}}[\alpha_{k-1}(s_{k-1}) + \gamma_k(s_{k-1} \to s_k)] \tag{2}$$

$$\beta_{k-1}(s_{k-1}) = \max_{S_k}[\beta_k(s_k) + \gamma_k(s_{k-1} \to s_k)] \tag{3}$$

$$\gamma_k(s_{k-1} \to s_k) = \frac{1}{2}x_k^s(L_e(u_k) + L_C y_k^s) + \frac{L_C}{2}x_k^p y_k^p \tag{4}$$

Equation (4) is derived with assuming that the code is transmitted through an AWGN channel with a noise variance of σ^2, where x_k and y_k represent the transmitted codeword

0-7803-9390-2/06/$20.00 ©2006 IEEE

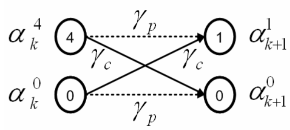

Figure 1. One butterfly pair.

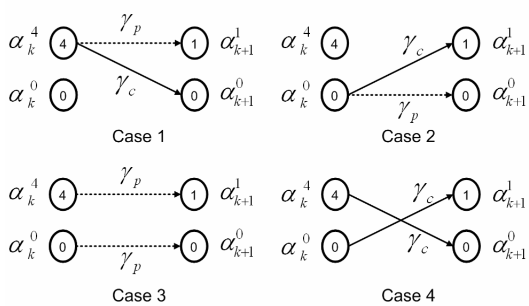

Figure 2. Four cases of forward metric calculation.

associated with this transition, and the codeword received from the channel, respectively. Also, superscripts p and s in (4) denote the parity bits and the systematic bits, respectively, $L_e(u_k)$ is the extrinsic information received from the other SISO decoder, and $L_C = 2/\sigma^2$.

B. *Log-MAP algorithm*

As some approximations are used in the max functions of (1), (2), and (3), there is performance degradation in the Max-log-MAP algorithm compared to the MAP algorithm. It can be improved by using the max* function [9] which is defined as

$$\max{}^*(x, y) = \max(x, y) + \ln(1 + e^{-|x-y|}) \tag{5}$$

To calculate each metric in log-MAP algorithm, only max functions in (1), (2), (3) are replaced with max* function.

III. PROPOSED REVERSE CALCULATION FOR MAX-LOG-MAP DECODING

Since the calculation directions of the two metrics are different, it is impossible to calculate both metrics simultaneously. If we can calculate the forward metrics in the reverse direction, there is no need to access the memory any more, saving power consumption. Assuming that forward metrics are calculated prior to backward metrics, we will present in this paper an efficient method to reversely calculate the forward metrics.

In a binary system, two branch transitions appear as a butterfly pair. Let us consider one butterfly pair shown in Fig. 1. Even though there are four branches in a butterfly pair, there exist only two distinguishable branch metric values for a butterfly pair [4]. The branch metric values, γ_p and γ_c, are always different except the case of $\gamma_p = \gamma_c = 0$. In the Max-log-MAP decoding, α_{k+1}^0 and α_{k+1}^1 are calculated from α_k^0 and α_k^4 as

$$\begin{aligned} \alpha_{k+1}^1 &= \max(\alpha_k^4 + \gamma_p, \alpha_k^0 + \gamma_c) \\ \alpha_{k+1}^0 &= \max(\alpha_k^4 + \gamma_c, \alpha_k^0 + \gamma_p) \end{aligned} \tag{6}$$

If α_k^0 and α_k^4 can be calculated from α_{k+1}^0 and α_{k+1}^1 in the same direction as backward metrics, the memory access can be substituted by calculations. From Fig. 1 and (6), the decision of α_{k+1}^0 and α_{k+1}^1 can be categorized into 4 cases as shown in Fig. 2.

1) Case 1

Because the values of α_{k+1}^0 and α_{k+1}^1 are determined by α_k^4, below equation is satisfied .

$$\begin{aligned} \alpha_{k+1}^1 &= \alpha_k^4 + \gamma_p \\ \alpha_{k+1}^0 &= \alpha_k^4 + \gamma_c \end{aligned} \tag{7}$$

From (7), we can see that if α_{k+1}^0 and α_{k+1}^1 come from α_k^4, $\alpha_{k+1}^1 - \gamma_p$ and $\alpha_{k+1}^0 - \gamma_c$ are equal to the value of α_k^4, as shown below.

$$\alpha_{k+1}^1 - \gamma_p = \alpha_{k+1}^0 - \gamma_c = \alpha_k^4 \tag{8}$$

2) Case 2

This case is very similar to case 1, and we can see that if α_{k+1}^0 and α_{k+1}^1 come from α_k^0, $\alpha_{k+1}^1 - \gamma_c$ and $\alpha_{k+1}^0 - \gamma_p$ are equal to the value of α_k^0.

$$\alpha_{k+1}^1 - \gamma_c = \alpha_{k+1}^0 - \gamma_p = \alpha_k^0 \tag{9}$$

3) Case 3

The values of α_{k+1}^0 and α_{k+1}^1 come from α_k^0 and α_k^4 as follows.

$$\begin{aligned} \alpha_{k+1}^1 &= \alpha_k^4 + \gamma_p \\ \alpha_{k+1}^0 &= \alpha_k^0 + \gamma_p \end{aligned} \tag{10}$$

From (6) and (10), the followings are derived.

$$\begin{aligned} \alpha_{k+1}^1 &> \alpha_k^0 + \gamma_c \\ \alpha_{k+1}^0 &> \alpha_k^4 + \gamma_c \end{aligned} \tag{11}$$

From (10) and (11), the following expressions come out.

$$\begin{aligned} \alpha_{k+1}^1 - \gamma_c &> \alpha_{k+1}^0 - \gamma_p = \alpha_k^0 \\ \alpha_{k+1}^0 - \gamma_c &> \alpha_{k+1}^1 - \gamma_p = \alpha_k^4 \end{aligned} \tag{12}$$

From (12), we can see that between two paths connected to each α_k, the smaller one is the value of α_k.

4) Case 4

This case is analogous to case 3, and we can derive the following expressions.

$$\begin{aligned} \alpha_{k+1}^1 - \gamma_p &> \alpha_{k+1}^0 - \gamma_c = \alpha_k^4 \\ \alpha_{k+1}^0 - \gamma_p &> \alpha_{k+1}^1 - \gamma_c = \alpha_k^0 \end{aligned} \tag{13}$$

From (13), the smaller $\alpha_{k+1} - \gamma$ is the value of α_k as in case 3.

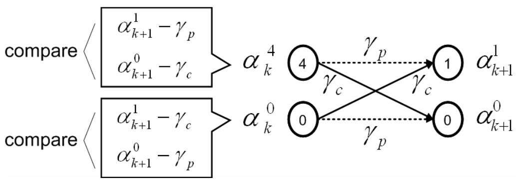

Figure 3. The strategy for reverse calculation.

A. *Strategy for Reverse Calculation*

Among the four cases, case 1 and case 2 can be satisfied simultaneously if and only if $\gamma_p = \gamma_c = 0$. To avoid such a situation, although the situation is extremely rare, a zero-valued branch metric is replaced by a value of minimal non-zero magnitude. The replacement makes the four cases unique and experimental results show that the performance influence induced by the replacement is ignorable. Looking at the above four cases, we can calculate the forward metrics in the reverse direction, as is summarized below.

- When we calculate α_{k+1} from α_k, the only α_k value which never wins in the max competitions of (6) is saved into the memory. If two values for a max function are equal, the loser in the other max function is saved in the memory. Later when we need α_k, we calculate α_k from α_{k+1} as described below.
- First, we calculate two $\alpha_{k+1}-\gamma$ values on the paths connected to a state of α_k, as shown in Fig. 3.
- If these values are equal, then they are the value of α_k as indicated in (8) and (9). And the value of α_k of the other state in the butterfly pair has to be loaded from the memory. The latter α_k must be in the memory because it was the loser in the max competition.
- If these values are different in both the states of a butterfly pair, each state α_k takes the smaller value of two $\alpha_{k+1}-\gamma$ values, as shown in (12) and (13). Since in this case the two α_k's can be reversely calculated from the two α_{k+1}'s belonging to the butterfly pair, there is no need to store any α_k's into the memory.

This strategy stores, in the worst case, only one α_k in the memory for a butterfly pair, instead of two α_k's required in the conventional turbo decoding. Therefore the required memory size is reduced to a half of the original memory size, and the number of memory accesses is reduced drastically. Compared to the loser storing method [5], the proposed strategy requires memory accesses only when the loser cannot be recovered by the reverse calculation, reducing the number of memory accesses significantly than the loser storing method.

IV. HYBRID LOG-MAP ALGORITHM

In this section, we propose a new decoding algorithm called hybrid log-MAP, which gives almost the same performance as the conventional log-MAP algorithm. The reverse calculation method proposed for the Max-log-MAP decoding is also applied to the hybrid log-MAP algorithm.

TABLE I. SPECIFICATION OF THE PROPOSED HYBRID LOG-MAP DECODER

Application	W-CDMA
Block size	1024
Sliding Window size	32
The number of Iteration	8 (Fixed)
Scaling factor	0.875
Quantization	Received input : (4, 2) Extrinsic Information : (6,2) Branch metric : (7,2) State metric : (9,2)

The reverse calculation method for the Max-log-MAP decoding can be applied to the log-MAP decoding if all the max* functions used for the forward metric calculation in the log-MAP algorithm are replaced with the max functions. This replacement makes the forward metric calculation equal in both of the log-MAP and Max-log-MAP algorithms. The replaced max function leads to performance degradation in the log-MAP algorithm. Since the backward metric calculation and $L_R(u_k)$ calculation are still based on the max* function even if the forward metric calculation is replaced with the max function, the performance of the hybrid log-MAP decoding will be placed between the conventional log-MAP and Max-log-MAP decodings. In addition, more performance improvement is expected in the proposed hybrid log-MAP decoding if it is associated with the scaling technique due to the features of the Max-log-MAP decoding [10], [11].

V. EXPERIMENTAL RESULTS

Based on the proposed hybrid log-MAP decoding, a turbo decoder of which specification is presented in Table I was described in Verilog-HDL and synthesized using a 0.18 μm standard-cell library and compiled SRAM memories. In Table I, (q, f) denotes a quantization scheme that uses q bits in total and f bits to represent the fractional part.

Table II shows the memory specification of the proposed decoder, comparing with the conventional decoder and the loser storing method [5]. Employing the proposed reverse calculation scheme reduces the state memory size by 50%. Therefore, the total memory size is reduced by 39.2%. The proposed reverse calculation scheme reduces the memory accesses by about 80% because there is no need to access the memory if both the metrics in a butterfly can be reversely calculated, while the loser storing method reduces the memory accesses by 50% because one of two metrics in a butterfly should be loaded.

The power consumption of the proposed SISO decoder in 1.2dB SNR is presented in Table III, compared to the conventional decoder and the loser storing method. As the SISO logic complexity of the hybrid log-MAP decoder is comparable to that of the loser storing method, the less total power consumption is expected due to the less power consumption in the state memory.

The proposed decoder is designed with 21820 gates, while the conventional decoder requires 18004 gates when they are synthesized under the constraint of the same

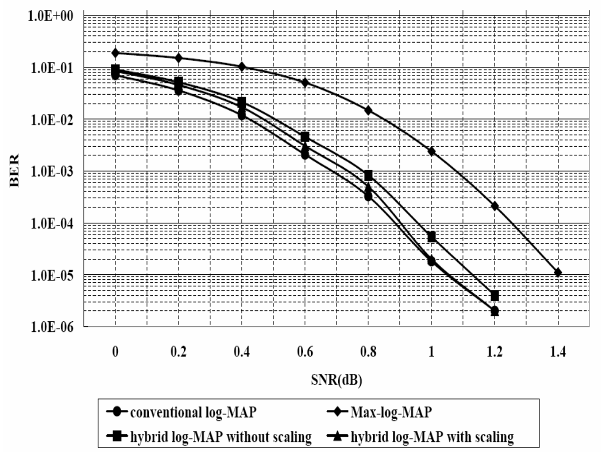

Figure 4. Performance comparison of four algorithms.

TABLE II. MEMORY SIZE COMPARISON OF A SISO DECODER FOR SINGLE-PORT SRAM.

	Conventional log-MAP decoder	Loser Storing Method [5]	Proposed Hybrid log-MAP decoder
Branch memory	4 banks, 32*10 bits/ bank	4 banks, 32*10 bits/ bank	4 banks, 32*10 bits/ bank
	1280 bits	1280 bits	1280 bits
Alpha state memory	2 banks, 32*(9*8) bits/bank	8 banks, 32*9 bits/ bank	8 banks, 32*9 bits/bank
	4608 bits	2304 bits	2304 bits
Flag memory	None	4 banks, 32bits/bank	None
		128 bits	
Total	5888 bits	3712 bits	3584 bits
	100%	63.0%	60.8%

TABLE III. POWER COMPARISON OF SISO DECODERS

	Conventional log-MAP decoder	Loser Storing Method [5]	Proposed hybrid log-MAP decoder
Branch memory	274.74 µW/MHz	274.74 µW/MHz	274.74 µW/MHz
Alpha State memory	573.34 µW/MHz	261.80 µW/MHz	146.59 µW/MHz
SISO logic	348.47 µW/MHz	N.A	364.43 µW/MHz
Total	1196.55 µW/MHz	N.A	785.76 µW/MHz
	100%	N.A	65.60%

operation frequency of 200MHz. The logic complexity of the proposed decoder is increased by 21.2%. The critical path delay is 4.74ns. As a result, the proposed decoder can operate at approximately 200MHz, which is enough to meet the W-CDMA standard specification of 2Mbps.

The BER performances of the Max-log-MAP, conventional log-MAP and hybrid log-MAP algorithms are presented in Fig. 4. The extrinsic information scaling improves the performance of the hybrid log-MAP decoding. The hybrid log-MAP with the scaling factor shows almost the same performance as that of the conventional log-MAP algorithm. In the worst case, the performance of the proposed decoder associated with scaling is only 0.05dB lower than the conventional log-MAP decoder.

VI. CONCLUSION

This paper has presented a new efficient reverse calculation method to reduce power consumption by minimizing memory accesses required in turbo decoding, and has modified the log-MAP algorithm to apply the proposed reverse calculation. The hybrid log-MAP algorithm shows almost the same performance as the log-MAP algorithm. Experimental results show that about 80% forward metrics can be reversely calculated in the W-CDMA standard, power consumption and memory size are reduced by approximately 35% and 40%, respectively, compared to the conventional log-MAP turbo decoder.

ACKNOWLEDGMENTS

This work was supported by Institute of Information Technology Assessment through the ITRC and by the Korea Science and Engineering Foundation through MICROS center and by IC Design Education Center (IDEC).

REFERENCES

[1] C. Berrou, A. Glavieux and P. Thitimajshima, “Near shannon limit error correcting coding and decoding: Turbo codes,” in Proc. Of In. Conf. Commun., pp.1064-1070, May 1993.

[2] C. Schurgers, F. Catthoor and M. Engels, “Memory optimization of MAP turbo decoder algorithms,” IEEE Trans. VLSI Syst., vol.9, no.2, pp.305-312, April 2001.

[3] M. C. Shin and I. C. Park, “A programmable turbo decoder for multiple third-generation wireless standards,” IEEE Int. Solid-State Circuits Conf. (ISSCC), pp.154-155, Feb. 2003.

[4] 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.

[5] 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 Int. Symp. On Circuits and Systems, vol.2, pp.25-28, May 2003.

[6] D. S. Lee, and I. C. Park, “Low-power log-MAP turbo decoding based on reduced metric memory access,” in Proc. Of IEEE Int. Symp. On Circuits and Systems, pp.3167-3170, May 2005.

[7] P. McAdam, L. Welch, and C. Weber, “MAP bit decoding of convolutional codes,” Int. Symp. on inform. Theory, p.91, 1972.

[8] J. Hagenauer and P. Hobber, “A Viterbi algorithm with soft-decision outputs and its applications,” IEEE GLOBECOM, pp.47.1.1-47.1.7, Nov.1989.

[9] P. Robertson, E. Villebrun, and P. Hoeher, “A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain,” Proc. Of ICC, pp.1009-1013, June 1995.

[10] J. Vogt and A. Finger, “Improving the max-log-MAP turbo decoders,” Electronics Letters, Volume 36, Issue 23, pp.1937-1939, Nov. 2000.

[11] H. Claussen, H. R. Karimi, and B. Mulgrew, “Improved max-log-MAP turbo decoding using maximum mutual information combining,” IEEE Int. Symp. on Personal, Indoor and Mobile Radio Comm. (PIMRC), Volume 1, 7-10, pp.424-428, 2003.