Block Processing Technique for Low Power Turbo Decoder Design

Inkyu Lee† Marisa Lopez Vallejo‡ Syed Aon Mujtaba†

†Wireless Systems Research Department

Agere Systems, Murray Hill, NJ 07974

email: {inkyu,mujtaba}@agere.com

‡Dpto. Ingenieria Electronica

ETSI Telecomunicacion, Ciudad Universitaria, Madrid, Spain

email: marisa@die.upm.es

Abstract

In this paper, we apply a block processing technique to the MAP algorithm used in Turbo decoding. This new technique leads to a power-efficient way to access memory, and to a reduced memory size. We introduce the "Very Long Data Word" (VLDW) memory architecture, which leads to a reduction in power consumption for memory access operations. The proposed architecture provides a low power implementation of the turbo decoder.

I. Introduction

In the channel coding area, turbo codes [1] have gained a lot of interest due to the promise they offer in approaching the Shannon limit. It has been shown that turbo codes achieve outstanding bit error rate performance with a relatively simple iterative decoding algorithm. Due to their excellent performance, the third generation wireless mobile system defined in the UMTS standard adopts the turbo codes as the channel coding scheme for high data rate transmission.

The basic element of a turbo decoder is the *Soft Input Soft Output (SISO)* decoding module, which provides soft outputs. These soft values indicate the reliability of the outputs, and are usually represented as log likelihood ratios. The maximum *a posteriori* (MAP) computing algorithm can be used to obtain the required soft outputs. Nevertheless, this algorithm has two serious drawbacks when implemented in hardware: the computational complexity and power consumption. Consequently, several authors have tried to address these problems [2][3][4].

In many applications such as handset terminals, minimizing the power consumption is critical, since battery power is normally limited. In this work, we apply a block processing technique to the conventional MAP algorithm in order to provide an efficient parallel architecture, and show that a significant reduction can be achieved in terms of the memory access power and the memory size.

The following section describes the MAP and Log MAP algorithm for the turbo decoder, and then the block processing technique is proposed in Section III. To illustrate our scheme, the turbo code design defined in the UMTS standard will be used as an example. Section IV discusses a power efficient design which can be combined with the block processing technique. We also introduce the Very Long Data Word (VLDW) architecture which allows a substantial power reduction combined with the proposed structure. Finally Section V summarizes this paper.

II. Conventional MAP algorithm

In this section, we explain the conventional MAP algorithm in the context of the turbo code defined in the UMTS standard. The turbo code in the UMTS standard employs two 8 state constituent encoders, and the MAP algorithm can be used in the decoder to compute the log likelihood values for each user data. Figure 1 (a) shows an 8 state trellis description from time $k-2$ to time k. The number next to the state indicates the

0-7803-7484-3/02/$17.00 ©2002 IEEE.

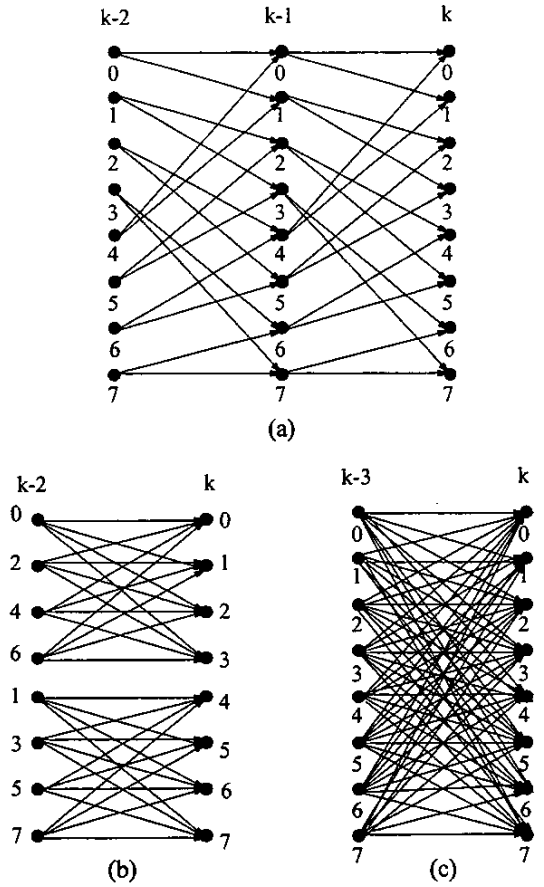

Fig. 1. (a) conventional trellis (b) block processing trellis with $N = 2$ (c) block processing trellis with $N = 3$

state number.

First we define α_i^k and β_i^k as the forward and backward variables at time k in state iΓrespectively. AlsoΓ $\gamma_{i,j}^k$ is defined as the branch metric associated with the transition from state i at time $k-1$ to state j at time k. We assume the branch metric $\gamma_{i,j}^k$ is computed from the signal input y_k. ThenΓassuming that the number of states is LΓthe forward recursion for state j is performed by

$$\alpha_j^k = \sum_{i \in S} \alpha_i^{k-1} \gamma_{i,j}^k \qquad \text{for } j = 0, 1, \cdots L-1$$

where $i \in S$ is a set of states at time $k-1$ which have a valid transition to the state j at time k. In case the input is binaryΓthe set S contains two states.

SimilarlyΓthe backward recursion is carried out by

$$\beta_i^{k-1} = \sum_{j \in S} \beta_j^k \gamma_{i,j}^k \qquad \text{for } i = 0, 1, \cdots L-1$$

where $j \in S$ is a set of states at time k which have a valid transition from the state i at time $k-1$.

Following the forward and backward recursionsΓthe log likelihood value for each user symbol u_k is computed by

$$L(u_k) = \log \left(\frac{\sum_{(i,j) \in S^+} \alpha_i^{k-1} \gamma_{i,j}^k \beta_j^k}{\sum_{(i,j) \in S^-} \alpha_i^{k-1} \gamma_{i,j}^k \beta_j^k} \right)$$

where a state pair $(i,j) \in S^+$ is defined as one that has a transition from state i at time $k-1$ to state j at time k corresponding to the user symbol $u_k = +1$. S^- is defined similarly for $u_k = -1$.

NowΓby substituting $A_j^k = \log \alpha_j^k, B_j^k = \log \beta_j^k$ and $\Gamma_{i,j}^k = \log \gamma_{i,j}^k$ to the above equationsΓwe obtain the Log MAP algorithm. AlsoΓusing a property of $\log(\exp x + \exp y) = \max(x,y) + \log(\exp(-|x-y|) + 1)$Γthe Log MAP algorithm [5] is simply described as

$$A_j^k = \max_{i \in S}{}^*(A_i^{k-1} + \Gamma_{i,j}^k) \qquad \text{for } j = 0, 1, \cdots L-1$$

$$B_i^{k-1} = \max_{j \in S}{}^*(B_j^k + \Gamma_{i,j}^k) \qquad \text{for } i = 0, 1, \cdots L-1$$

where $\max^*(x,y)$ is defined as $\max(x,y) + \log(\exp(-|x-y|) + 1)$.

As an exampleΓin the forward recursionΓA_2^k in state 2 at time k as shown in Figure 1 (a) is computed as

$$A_2^k = \max^*(A_1^{k-1} + \Gamma_{1,2}^k, A_5^{k-1} + \Gamma_{5,2}^k).$$

III. Block processing technique

The block processing technique [6]Γ[7] has been proposed to speed up the convolutional decoder in the ASIC implementation by processing multiple signal inputs at the same time. We extend this idea to a MAP processor in the turbo decoder to help minimize power consumption in its DSP implementation.

By combining multiple trellises from time $k-N$ to time kΓwe can obtain a merged trellis structure with increased parallelism. In Figure 1 (b) and (c)Γmerged trellises for $N = 2$ and 3 are illustrated. In this new trellisΓthe number of states remains the sameΓbut each state takes 2^N inputs instead of two.

In this schemeΓthe state metric α_i^k at time k is computed directly from α_i^{k-N}. SimilarlyΓβ_i^{k-N} is computed directly from β_i^k. The update recursion now becomes

$$A_j^k = \max_{i \in S}{}^*(A_i^{k-N} + \bar{\Gamma}_{i,j}^k) \qquad \text{for } j = 0, 1, \cdots L-1$$

$$B_i^{k-N} = \max_{j \in S}^{*}(B_j^k + \bar{\Gamma}_{i,j}^k) \qquad \text{for } i = 0, 1, \cdots L-1$$

where $\bar{\Gamma}_{i,j}^k$ is defined as the combined branch metric associated with the transition from state i at time $k - N$ to state j at time kΓand is computed in the trellis accordingly.

For exampleΓin the trellis for $N = 2$ΓA_2^k in state 2 at time k is computed as

$$A_2^k = \max^{*}(A_0^{k-2} + \bar{\Gamma}_{0,2}^k, A_2^{k-2} + \bar{\Gamma}_{2,2}^k,$$

$$A_4^{k-2} + \bar{\Gamma}_{4,2}^k, A_6^{k-2} + \bar{\Gamma}_{6,2}^k).$$

Here we can compute $\bar{\Gamma}_{0,2}^k$ as $\bar{\Gamma}_{0,2}^k = \Gamma_{0,1}^{k-1} + \Gamma_{1,2}^k$Γand other branch metrics can be determined in a similar way from the trellis.

The block processing technique can be extended to an arbitrary N. HoweverΓif N is greater than $\log_2 L$Γthen the combined metric $\bar{\Gamma}_{i,j}^k$ is no longer defined uniquely. ThereforeΓwe focus on $N = 2$ and $N = 3$ cases in this workΓas an 8 state encoder is emploied in the UMTS standard.

IV. Power efficient memory access design

In most data dominant applications such as turbo decodersΓdata access and transfer operations normally dominate the whole power consumption [8]. It has been identified [2]Γ[3] that power dissipation associated with the state metric and input signal storage and access dominates the overall power in the turbo decoder. One analysis shows that the input signal transfer and the state metric access contributes 20% and 72% of the total power consumptionΓrespectively [2]. Since the memory power consumption is linearly proportional to the access rate [3]Γminimizing the number of data transfers is an effective way of low power design.

By applying the block processing techniqueΓwe can exploit this new structure to save data transfer operations. Figure 2 compares the input/output relationship in the conventional MAP algorithm and the block processing algorithm. As shown in this figureΓat each time kΓthe conventional MAP decoder takes a signal input y_k and the state metrics $\alpha_i^{k-1}, \beta_i^k$ to generate the updated metrics $\alpha_i^k, \beta_i^{k-1}$. In contrastΓthe block processing MAP decoder takes a block of signal inputs $[y_k, y_{k+1}, \cdots y_{k+N-1}]$ and the state metrics $\alpha_i^{k-N}, \beta_i^k$ to compute the updated state metrics $\alpha_i^k, \beta_i^{k-N}$Γand this operation is carried out at every Nth clock cycle.

The proposed block processing technique can be used not only for memory savingsΓbut also to reduce the power consumption. This is performed in two waysΓ

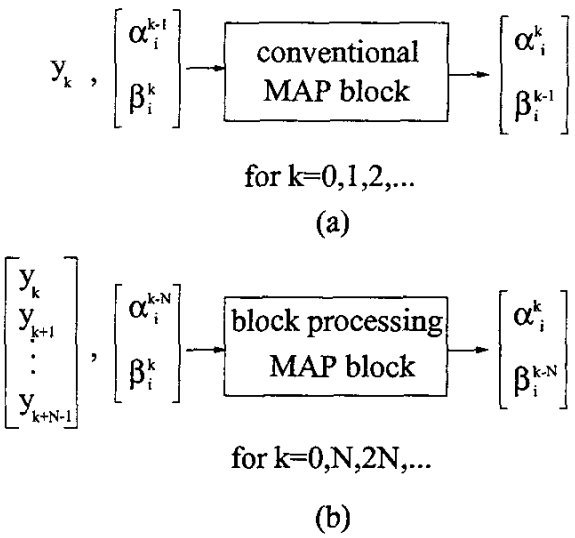

Fig. 2. input/output block diagram (a) the conventional MAP decoder (b) the block processing MAP decoder

both dealing with the major access operations that take place in the decoding process: accessing for input signal y_k and accessing for the state metrics α_i^k, β_i^k.

FirstΓwe can use a multiple word based read/write memory architecture to reduce the data transfer power consumption. This can be achieved because the proposed block processing technique takes an input signal block of length N instead of a single data y_k.

SecondlyΓthe state metrics can be stored and accessed at rate f/N Hz instead of f Hz in the conventional structure. ThereforeΓboth the memory size for the state metric and memory access rate are reduced by N. In what followsΓthese two aspects which lead to a power efficient architecture will be explained in detail.

A. Low power design for input signal access

Here we will explain a new structure to reduce the power consumption when accessing the input signal y_k. To this endΓwe introduce the Very Long Data Word (VLDW) architecture for input signal access.

Since the proposed block processing architecture takes a multiple signal input block instead of the conventional single word memory access operationΓwe can utilize a multiple word base read/write architecture. In this architecture several memory words are concatenated to reduce the number of accesses to the memoryΓand consequently decrease the power consumption. This is illustrated in Figure 3.

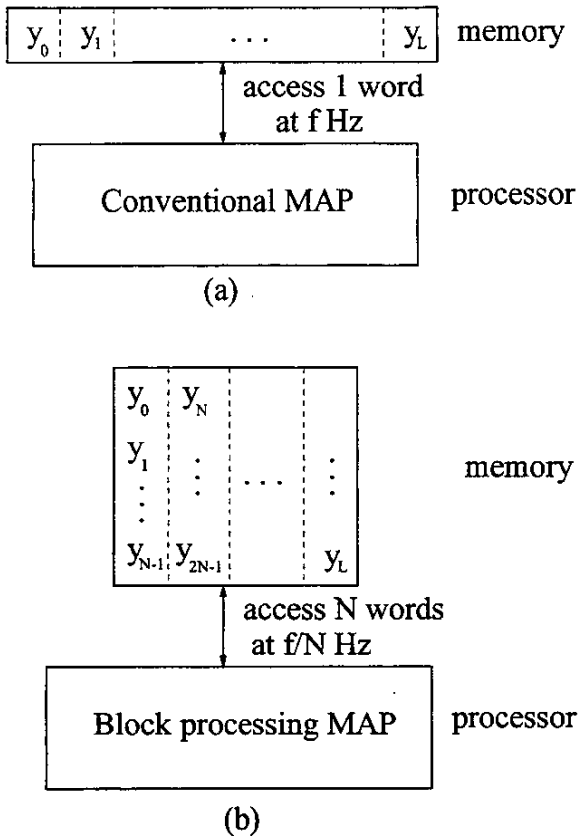

Fig. 3. Memory access architecture for input signal (a) conventional structure (b) the new structure

In the conventional MAP structure, a signal input y_k in memory is accessed at frequency f Hz. In contrast, the block processing schme takes a block of N signal inputs $[y_k, y_{k+1}, \cdots y_{k+N-1}]$ at a reduced frequency f/N Hz. Note that in both cases the total memory size for the input signals is the same.

This VLDW architecture can reduce the power consumption in memory access by N times, since the current discharge rate in the memory cell is reduced by N [8]. Note that the VLDW architecture can be applied to the conventional MAP module by employing a register file of size N on chip. But this scheme is not as efficient as in the block processing structure.

	$N=1$	$N=2$	$N=3$
addition	$8 \cdot 2$	$8 \cdot 8/2$	$8 \cdot 24/3$
max* ops	8	$8 \cdot 3/2$	$8 \cdot 7/3$

TABLE I

THE NUMBER OF OPERATIONS PER DATA SYMBOL

B. Low power design for the state metric memory access

Compared to the input signal access, the state metric storage and access normally take up much more power since at each time instance the whole state metric needs to be stored.

The biggest power savings in the block processing technique are possible in the memory structure associated with the state metric access. As illustrated in section III, we only update the state metric α_i^k, β_i^k for $k = 0, N, 2N, \cdots$. This means that we do not need to store the state metric for other time instances such as $k = 1, 2 \cdots N-1$. This greatly reduces the memory size for the state metric and at the same time this achieves significant power savings since the access rate is reduced by N times accordingly.

Note that in the block processing technique the computation of the log likelihood values should be modified to incorporate the change in trellis description, and this is illustrated in [9].

Table I lists the number of operations required per user data symbol for different N values. The analysis shows that the computation complexity grows with N, but the power consumption for data computation is small compared to that in the data access and transfer operations.

V. CONCLUSION

In this paper, we apply the block processing technique to the MAP algorithm in a turbo decoder. The proposed block processing MAP algorithm results in a much smaller memory size for the state metric and a power efficient memory access architecture. Utilizing the VLDW memory architecture minimizes the number of data transfer operations, which results in substantial power savings. Please note that the proposed technique can be combined with other techniques such as the serial/parallel sliding window technique [3], [10] to further reduce the power consumption.

REFERENCES

[1] A. G. C. Berrou and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. of IEEE International Conferences on Communications '93*, pp. 1064–1070, May 1993.

[2] C. Schurgers, F. Catthoor, and M. Engels, "Energy efficient data transfer and storage organization for a MAP turbo decoder module," in *Proc. of ISLPED '99*, pp. 76–81, 1999.

[3] A. Worm, H. Michel, and N. Wehn, "Power minimization by optimizing data-transfers in turbo-decoders," in *Kleinheubacher Berichte*, pp. 343–350, September 1999.

[4] Z. Wang, H. Suzuki, and K. K. Parhi, "Vlsi implementation issues of turbo decoder design for wireless communications," in *Proc. of IEEE Workshop on Signal Processing Systems*, pp. 503–512, 1999.

[5] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. of IEEE International Conferences on Communications '95*, pp. 1009–1013, 1995.

[6] G. Fettweis and H. Meyr, "Parallel Viterbi algorithm implementation: Breaking the ACS-bottleneck," *IEEE Transactions on Communications*, vol. COM-37, pp. 785–790, August 1989.

[7] H. K. Thapar and J. M. Cioffi, "A block processing method for designing high-speed Viterbi detectors," in *Proc. of International Conference on Communications*, pp. 1096–1100, 1989.

[8] F. Catthoor and et al. *Custom memory management methodology*. Boston, MA: Kluwer Academic Publishers, 1998.

[9] I. Lee, "Modification of the MAP algorithm for memory savings," *submitted to IEEE Transactions on Communications*, February 2002.

[10] A. J. Viterbi, "An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes," *IEEE Journal on Selected Areas in Communications*, pp. 260–264, February 1998.