VLSI IMPLEMENTATION CONSIDERATIONS FOR TURBO DECODING USING A LOW LATENCY LOG-MAP

Arun Raghupathy *

Qualcomm Inc., 6455 Lusk Blvd.,
San Diego, CA-92121

K. J Ray Liu

Electrical Engineering Department
Univ. of Maryland, College Park, MD-20742

ABSTRACT

The Soft-Output Viterbi Algorithm (SOVA) and the log-Maximum Aposterior Probability (log-MAP) algorithm are commonly used in turbo decoding. In this paper, we propose to modify the sliding window MAP-algorithm in [5] to reduce the computational delay even further. We compare the simulation performance of this low latency log-MAP algorithm with the sliding window log-MAP. We also estimate the VLSI implementation complexities of the SOVA, the log-MAP and the proposed low latency log-MAP.

1. INTRODUCTION

In turbo decoding, two algorithms have been commonly used, the SOVA [2],[3] and the MAP algorithm. In implementations, the SOVA is preferred because it is less complex when compared to the original MAP. Various approximations [4] have been developed such as the log-MAP to overcome the implementation problems of the MAP. The log-MAP uses the max* operation [5] to update the path metrics unlike the Viterbi Algorithm in which the max operation is used. In additive white Gaussian noise(AWGN) channels, the log-MAP is about 0.5dB better than the SOVA at low SNR (See Figure 2). On fading channels it has been shown [6] that the log-MAP outperforms the normalized SOVA [3] by about 2-3dB. A sliding window log-MAP algorithm that reduces latency and, hence, the path metric storage requirements has been proposed in [5], [7]. The increased complexity of the log-MAP may be justified for fading channels or AWGN channels at very low SNR. However, a comparison of the VLSI implementation complexity of the SOVA and the log-MAP is not available in the literature.

2. IMPLEMENTATION COMPLEXITY

We compared the VLSI implementation complexities of the SOVA and the log-MAP so as to allow system designers to make a fair tradeoff between performance and complexity. Based on a description of the computational modules in a hardware description language, we performed standard cell synthesis followed by automatic layout to estimate the relative VLSI complexities. This was done for both parallel and serial state implementations. Results showed that a parallel state implementation of one turbo-decoding iteration using the log-MAP was about 3 times more expensive in terms of silicon area when compared with the SOVA. A serial state implementation of the log-MAP was shown to be about 1.3 times that of the SOVA (See Table 1). It should be pointed out that multiple turbo iterations can be performed either by reusing the single-turbo-iteration hardware or by cascading multiple such hardware units.

One of the critical components in iterative decoding is the delay in decoding. The decoding delay is made up of computation delay and interleaver delay [8]. In small frame applications [6], [8] (such as for speech) the computation latency can be a significant fraction of the decoding delay. We modified the algorithm in [5] to reduce the computational delay. We used the property that during the acquisition process (see Figure 1(a)), we are interested only in the path metrics at the end of the acquisition phase (i.e. when the path metric values are reliable). We applied a look-ahead like transformation (see Figure 1(b)) to nearly halve the computation latency. We refer to this algorithm as the low-latency log-MAP. An approximation is used to reduce the VLSI implementation complexity of this algorithm (for details of the approximation see [1]). This approximation leads to almost no loss in performance (See the SNR vs. BER plot in Figure 2). A comparison of the VLSI implementation complexities of the approximate low-latency log-MAP was made with the log-MAP (See [1] for details of the architectures used in each case). Again, synthesis and automatic layout was performed starting from a HDL description. Results showed that a parallel state implementation of the approximate low-latency log-MAP was about 1.5 times as expensive in terms of silicon area as the log-MAP. The complexity of a serial state implementation of one turbo decoder iteration of the approximate low-latency log-MAP was about the same as the log-MAP. The VLSI implementation results are summarized in Table 1. The computational latency was shown to be reduced by about 40%. The total latency was reduced by about 12.5% (for an interleaver size of I=256 [6]). For all comparisons the standard 16-state rate 1/2 turbo code in [9] was used.

3. CONCLUSION

We can conclude that improved performance (which comes at the cost of increased complexity) is obtained by using the log-MAP. We attempted to quantify this increase in VLSI implementation complexity to enable the appropriate choice of algorithm for a particular application. In

*This work was supported in part by the NSF NYI Award MIP9457397 and the ONR grant N00014-93-10566, and was done while this author was at University of Maryland, College Park.

0-7803-5123-1/99/ $10.00 © 1999 IEEE

Alg.	$C_{Alg-ITER,p}$ (mm^2)	$C_{Alg-ITER,s}$ (mm^2)
SOVA	22.78	10.45
MAP	68.13	13.18
LL-MAP	100.21	12.56

Table 1: Summary of Area Estimates using Synthesis at 0.8μ for one iteration of the SOVA, Log-MAP and Low-Latency Log-MAP

practice, the nature of the channel (AWGN or fading) has to be taken into account while choosing between SOVA and log-MAP. Amongst log-MAP algorithms, if computational delay is a critical parameter (for example in speech applications large delays cannot be tolerated), then the approximate low latency MAP algorithm that we proposed should be used. In particular, when the frame size is small (for eg. 256), our algorithm reduces the overall latency by 12.5% when compared with the algorithm in [5] at almost no cost in terms of BER (See [1] for details).

4. REFERENCES

[1] A. Raghupathy, "Low Power and High Speed Algorithms and VLSI Architectures for Error Control Coding and Adaptive Video Scaling," *Ph. D Thesis*, Univ. of Maryland, College Park, U.S.A, Dec. 1998.

[2] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. on Info. Theory*, vol. 42, pp. 429–445, Mar. 1996.

[3] L. Papke, P. Robertson, and E. Villebrun, "Improved decoding with the SOVA in a parallel concatenated (turbo-code) scheme," in *IEEE Intl. Conf. on Comm.*, pp. 102–106, June 1996.

[4] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *European Trans. on Telecomm.*, vol. 8, pp. 119–125, Mar.-Apr. 1997.

[5] A. J. Viterbi, "An intuitive justification and simplified implementation of the MAP decoder for convolutional codes," *IEEE J. Sel. Areas in Comm.*, vol. 16, pp. 260–264, Feb. 1998.

[6] M. Valenti and B. Woerner, "Performance of turbo codes in interleaved flat fading channels with estimated channel state information," in *IEEE Vehicular Tech. Conf.*, pp. 66–70, IEEE, May 1998.

[7] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft-output decoding algorithms in iterative decoding of turbo codes," *JPL TDA Progress Report*, vol. 42, pp. 63–87, Feb. 1996.

[8] P. Jung, "Comparison of turbo-code decoders applied to short frame transmission systems," *IEEE J. Sel. Areas in Comm.*, vol. 14, pp. 530–537, Apr. 1996.

[9] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo codes," *IEEE Trans. Comm.*, vol. 44, pp. 1261–1271, Oct. 1996.

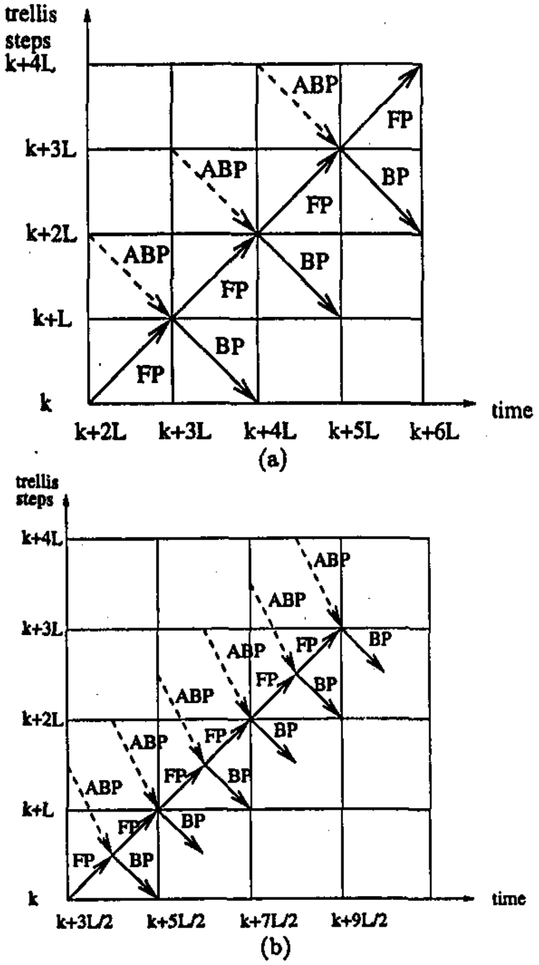

Figure 1: (a) Evolution in time of the original log-MAP algorithm (b) Evolution in time of the modified log-MAP algorithm

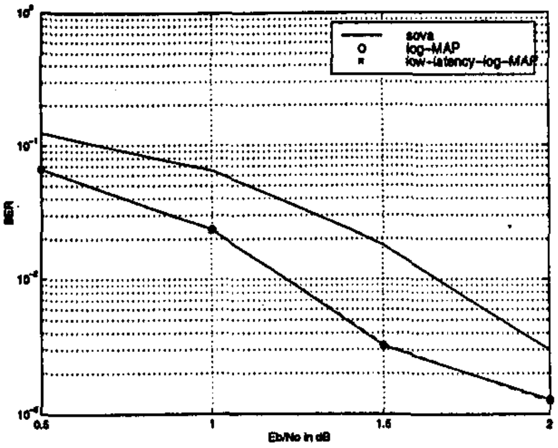

Figure 2: Comparative performance of SOVA, log-MAP and modified log-MAP