Design and Implementation of Low-Energy Turbo Decoders

Jagadeesh Kaza, *Member, IEEE* and Chaitali Chakrabarti, *Member, IEEE*

***Abstract*—Turbo codes have been chosen in the third generation cellular standard for high-throughput data communication. These codes achieve remarkably low bit error rates at the expense of high-computational complexity. Thus for hand held communication devices, designing energy efficient Turbo decoders is of great importance. In this paper, we present a suite of MAP-based Turbo decoding algorithms with energy-quality tradeoffs for additive white Gaussian noise (AWGN) and fading channels. We derive these algorithms by applying approximation techniques such as pruning the trellis, reducing the number of states, scaling the extrinsic information, applying sliding window, and early termination on the MAP-based algorithm. We show that a combination of these techniques can result in energy savings of 53.2%(50.0%) on a general purpose processor and energy savings of 80.66%(80.81%) on a hardware implementation for AWGN (fading) channels if a drop of 0.35 dB in SNR can be tolerated, at a bit error rate (BER) of 10^{-5}. We also propose an adaptive Turbo decoding technique that is suitable for low power operation in noisy environments.**

***Index Terms*—Adaptive turbo decoder, energy quality tradeoffs, low power, very large scale integration (VLSI).**

I. Introduction

TURBO CODING is an attractive channel coding scheme providing near optimal bit error rates (BERs) for data transmission within 0.5 dB of Shannon's limit at BER of 10^{-5} [1]. In fact, Turbo codes have been chosen for the new telemetry coding standard by the Consultative Committee for Space Data Systems (CCSDS) and for medium high-data rate transmission in the new Universal Mobile Telecommunications System (UMTS) third generation mobile communication standards. The superior performance of Turbo codes is due to the combination of parallel concatenated coding, recursive encoding, pseudorandom interleaving and iterative decoding. However, the superior performance comes at the cost of enormous computational complexity. The energy consumption of the coder is also very high compared to convolutional coders (e.g., Viterbi). Thus, third generation wireless devices have to balance the two conflicting requirements of low-energy consumption and high-performance requirements. In this paper, we have addressed this issue by presenting a suite of Turbo decoding algorithms with energy-quality tradeoffs.

The Turbo decoder structure (shown in Fig. 1) has two constituent decoders, denoted by Dec1 and Dec2 that implement *aposteriori* probability, and interleavers/de-interleavers that add randomness to the codes. The decoders generate soft outputs which represent how reliable the outputs are. The Turbo decoder typically runs a fixed number of iterations in order to guarantee minimum performance. Each constituent decoder has three types of soft inputs: x, y_i ($i = 1$ or 2), and the *apriori* information, which is also the extrinsic information provided by the other constituent decoder from the previous step of the decoding process. The soft output generated by each constituent decoder at time k consists of three components: A weighted version of x_k, the previous extrinsic information and a newly generated extrinsic information. The iterations continue with the extrinsic information getting updated and exchanged between the two decoders until a reliable hard decision is made. Each state in the trellis of the decoding algorithm is associated with an α and a β metric. These metrics are obtained through a forward and a backward recursion [see (1) and (2)] and have initial values that are known at the beginning (for α) or end (for β) of the block

$$\alpha_{k+1} = F_\alpha\left(\alpha_k, \lambda_k^{\text{int}}, \lambda_k^1, \lambda_k^2\right) \tag{1}$$

$$\beta_{k-1} = F_\beta\left(\beta_k, \lambda_k^{\text{int}}, \lambda_k^1, \lambda_k^2\right) \tag{2}$$

where λ_k^{int} denotes the *a priori* information used by the next decoder and λ_k denote the log likelihood ratios (the soft output for each decoded bit is determined from the log likelihood ratio) of the received symbols.

There are two main algorithms for Turbo decoding: The maximum *aposteriori* algorithm (MAP) [2] and the soft output Viterbi algorithm (SOVA) [3]. The SOVA based decoder is computationally less complex since it chooses the branch in the trellis with the highest probability and discards the other. MAP, on the other hand, does not reject any path and calculates the probabilities of each point in the trellis. In order to reduce the computational complexity of MAP, it is generally implemented in the logarithmic domain and the corresponding algorithm is referred to as the log-MAP algorithm. In this paper, we have implemented a version of log-MAP (referred to as Max log-MAP) where some of the additions are approximated by max operations [4]. It is worth mentioning that SOVA is a suboptimal algorithm and has inferior performance compared to log-MAP.

In recent years, several architectures have been designed for Turbo decoders [5]–[12]. The first of these was a baseline Turbo decoder implementation for deep space transmission applications [5]. There the impact on the VLSI complexity due to system parameters like the number of states, number of iterations and code rate have been evaluated. Area efficient decoding schemes such as pipelined interleaving, segmented

Manuscript received December 19, 2002; revised December 19, 2003. This work was supported by the Center for Low Power Electronics under EEC 9523338.

J. Kaza is with Texas Instruments Incorporated, Dallas, TX 75243 USA (e-mail: jagadeeshk@ti.com).

C. Chakrabarti is with the Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: chaitali@asu.edu).

Digital Object Identifier 10.1109/TVLSI.2004.832942

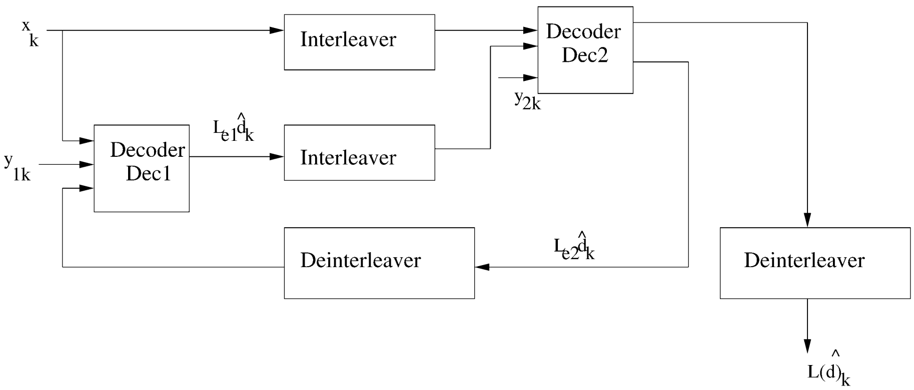

Fig. 1. Block Diagram of a Turbo Decoder.

sliding window approach and partial storage of state metrics approach have been reported in [6]. A 4-stage pipelined low power Turbo decoder has been proposed in [7] that resulted in 70 mW power dissipation for 1 Mb/s data rate. This architecture avoids complex operations such as exponent and logarithmic operations. Memory optimization of MAP turbo decoder algorithms using IMECs data transfer storage exploration methodology has been reported in [8] that resulted in a decrease in decoding energy (factor 2.5) for a rate $1/2$, 16 state sliding window SISO Turbo decoder with an interleaver size of 1000. Partitioning techniques to reduce the memory power consumption of Turbo decoders have been proposed in [9] that resulted in reduction of 70% with an area overhead of 23% for a rate $1/2$, 4-state code. More recently, a low power implementation for a Turbo decoder implemented on additive white Gaussian noise (AWGN) channel that employed approximation techniques like stopping criteria, quantization and reduction in states has been proposed in [10]. The ASIC architecture resulted in 70% reduction in power consumption over a fixed six iteration, 8-state baseline Turbo decoder at 2 dB of SNR for a rate $1/3$ code with block size of 840 bits.

All of the existing work on Turbo decoder architectures is for AWGN channels. However, in many real world situations such as radio, satellite and mobile channels, transmission errors are mainly caused by variations in the received signal strength referred to as *fading*. The AWGN channel model fails to accurately describe the dominant sources of noise in such channels. While Turbo decoders for fading channels has been described in [11], power consumption issues have not been addressed. In this paper, we address the issue of low energy Turbo decoder design for both AWGN and fading channels. We used the Rician mathematical model to describe the first order effects of fading.

Our approach to reducing the energy consumption of the Turbo decoder is by judicious use of approximation algorithms that reduce the computational complexity of the decoding algorithms (log-MAP/SOVA) with a minimal decrease in the BER. This is an extension of our earlier work that was presented in [12]. The approximation techniques that we studied reduce the number of computations by pruning the trellis, scaling the extrinsic information, terminating the iterations early, applying sliding window and reducing the number of states in the trellis. In each case we compared the performance with respect to BER and energy. We also investigated the effects of different interleavers on Turbo decoders. Our analysis showed that scaling of extrinsic information and pruning techniques result in the largest reduction in energy for both AWGN and fading channels. Our main contributions are as follows.

- Presented a suite of Turbo decoding algorithms with energy-quality tradeoffs. These algorithms are obtained by applying one or more approximation techniques to log-MAP based decoder.
- Proposed energy efficient algorithms for AWGN/fading environments by carrying out a systematic algorithmic exploration (using a combination of the approximation techniques) that resulted in 50% energy savings over the log-MAP based Turbo decoder, with a loss of 0.35 dB in SNR for a BER of 10^{-5} and block size of 2048 bits for AWGN and fading channels on a general purpose processor.
- Implemented the energy efficient algorithms in hardware (synthesized using Synopsys). Achieved 80.66% (80.89%) energy savings over log-MAP based Turbo decoder for AWGN (fading) channels.
- Proposed a low power adaptive Turbo decoding technique based on which the user can dynamically choose the required low power decoding algorithm based on the SNR and energy savings required.

We have no knowledge of any work on Turbo decoding that presents a comprehensive analysis of the implementation (general purpose as well as hardware) of energy efficient Turbo decoders for both AWGN and fading channels. As part of the hardware implementation, we have synthesized two energy-efficient algorithms that resulted in 0.35 dB loss in SNR at a BER of 10^{-5}. While the synthesized implementations result in larger energy savings compared to the general purpose processor implementation (81% versus 50%), they are not optimized implementations. Clearly, greater energy savings can be obtained if architectural and circuit-level energy reduction techniques are applied to the hardware implementation.

The rest of the paper is organized as follows. Section II gives a brief description of the different energy reduction techniques. Section III describes the proposed energy efficient algorithms and their synthesis results. Section IV describes the adaptive Turbo decoding technique and Section V concludes the paper.

TABLE I
DROP IN SNR AND ENERGY SAVINGS CORRESPONDING TO DIFFERENT STOPPING CRITERIA FOR LOG-MAP(SOVA)

Stopping Criterion	AWGN		Fading	
	Drop in SNR	% Energy Savings	Drop in SNR	% Energy Savings
CE	0.2(0.21)	11.2(12.1)	0.24(0.25)	13.2(14.3)
LLR	0.34(0.39)	17.5(18.2)	0.31(0.33)	17.9(19.2)
CRC	0.31(0.32)	14.3(15.2)	0.35(0.36)	21.2(20.4)

II. ENERGY REDUCTION TECHNIQUES

Although SOVA and log-MAP are low-complexity algorithms compared to MAP, they still require a large number of computations. In the rest of this section, we describe existing techniques to reduce the number of computations and their effect on energy consumption, when implemented on a general purpose processor (GPP). The experimental setup that was used to calculate the power consumption (and energy) on a GPP is based on Wattch [13], a framework for architectural level power analysis. All the simulations were performed on rate $1/2$, 8 state Turbo code with 8 PSK modulation, interleaver size 2048 bits, constraint length $(k) = 5$ and rician factor $\mathbf{K} = 5$. The quantitative quality results were obtained in terms of BER measurements. The quantitative estimates for energy values were obtained using Wattch. The processor used in Wattch is representative of a state of the art general purpose processor and had the following parameters: Issue width 4, window size 16, number of virtual registers 32 and number of physical registers 16, datapath width 64, L-1 data cache size 128:32:4 (128 K is the data cache size, 32 K is the line size and 4 is the set associativity). L-2 data cache 1024:64:1, L-1 instruction cache 512:32:1 and L-2 instruction cache similar to the L-2 data cache.

A. Reduction in the Number of States

In this method, the state trellis is reduced to a smaller structure and the search is performed on the reduced trellis [14]. The advantage of this approach is that the designer could tradeoff complexity with performance by selecting the degree of state reduction. The key concept in this method is to construct a survivor map on a reduced state trellis during the backward recursion and use it during the forward recursion.

This method has been applied to MAP and SOVA for AWGN and fading channels. The number of states were reduced from eight to four and the drop in SNR and energy savings were calculated. For AWGN channels, the energy savings and loss in SNR were 24.5% (25.2%) and 0.51 dB (0.58 dB) for log-MAP(SOVA) at a BER of 10^{-5}. For fading channels, reduction of states from eight to four had only a slight effect on the performance. The resulting energy savings was 14% (16%), with a loss of 0.3 dB (0.27 dB) in SNR for log-MAP (SOVA). Further reduction to two states resulted in significant performance degradation ($\approx$0.63 dB) for log-MAP for fading channels where as, for AWGN channels, the drop in SNR was $\approx$0.56 dB.

B. Reduction in the Number of Paths

The *T* and *M* algorithms [15] are pruning algorithms that reduce the number of paths of a classical Viterbi algorithm. The *M* algorithm retains the largest M_t sufficient statistics while the *T* algorithm retains sufficient statistics exceeding a threshold p_t. We chose to apply the *T* algorithm to the trellises in log-MAP and SOVA since the *T* algorithm has superior performance (compared to the *M* algorithm) though it introduces some manageable delay in decoding.

For AWGN channels, the energy savings in log-MAP(SOVA) were $\approx 25\%$ at the expense of 0.48 dB (0.63 dB) loss in SNR for a BER of 10^{-5}. For fading channels, the energy savings in log-MAP(SOVA) were 29% (30.5%) at the expense of 0.61 dB (0.68 dB) loss in SNR for a BER of 10^{-5}. Thus for fading channels, reduction in paths results in greater performance degradation and consequently larger energy savings. This is because in the presence of multipath in fading channels, it is possible that both the transmitted path and the alternate path are on the list of possible contenders, thus resulting in forced decision rejection.

C. Effects of Stopping Criteria

Since early termination of the iterative decoding helps conserve power, the effects of the following stopping criteria, namely, cross entropy (CE) [16], sign change in the loglikelihood and extrinsic informations (LLR) [17] and adaptive iteration based on cyclic redundancy check (CRC) [18], have been studied. The CE criterion has been devised based on the cross entropy between the distributions of the estimates at the outputs of the decoders at each iteration. The LLR method that we used is slightly different from what has been used in [17]. In our method, the difference of the absolute values of each extrinsic information and each log likelihood information from the output of the current and the previous decoding blocks is compared with a preset bound to check if all of them are larger than the bound. Also, the number of 1's from the output of the current and previous decoding blocks are compared against a preset bound. The decoding converges if the above conditions are true. The advantage of our LLR method is that it could be implemented using simple logic. In the CRC method in [18], cyclic redundancy checking is used to adaptively terminate the iteration of each frame and dynamic voltage scheduling (DVS) is used on top of this to further reduce the power consumption. In this paper, we do not use DVS.

Table I shows the drop in SNR (in dB) and the energy savings for log-MAP (SOVA) for a block size of 2 K and BER of $10^{-5}$for different stopping criteria. For AWGN channels, while LLR and CE are the best stopping criteria in terms of reduced energy consumption and performance, CRC (because of its faster convergence on fading channels) and CE are the best for fading channels. Figs. 2 and 3 show the BER versus SNR for AWGN and fading channels when all the three stopping criteria are applied on log-MAP. For AWGN channels, as SNR increases to 2 dB, the plots converge, whereas for fading channels, CE clearly outweighs the other two in terms of

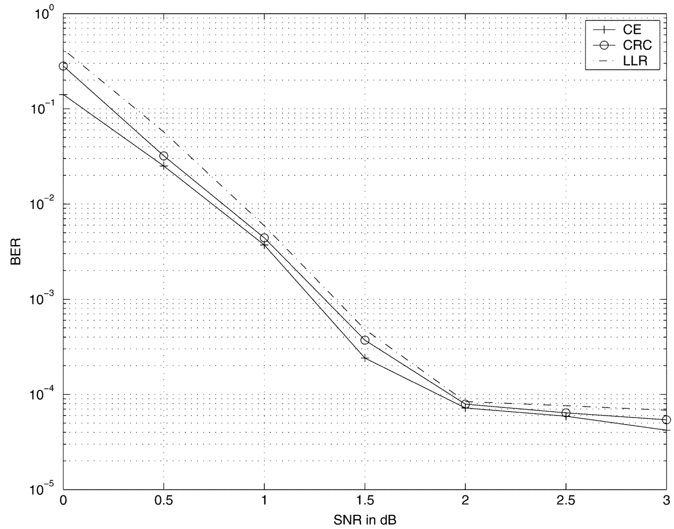

Fig. 2. BER versus SNR for various stopping criteria for log-MAP on AWGN channels, $\text{Block size} = 2048$ bits.

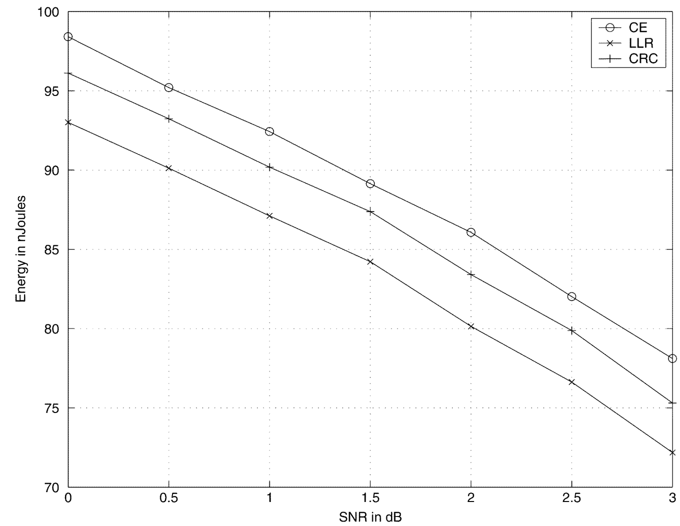

Fig. 4. Energy estimates of log-MAP for various stopping criteria on AWGN channels, $\text{Block size} = 2048$ bits.

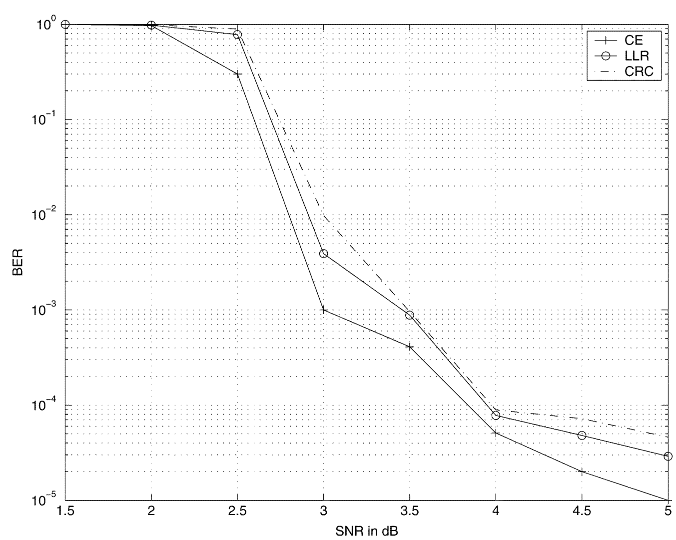

Fig. 3. BER versus SNR for various stopping criteria for log-MAP on fading channels, Rician factor $K = 5, \text{Block size} = 2048$ bits.

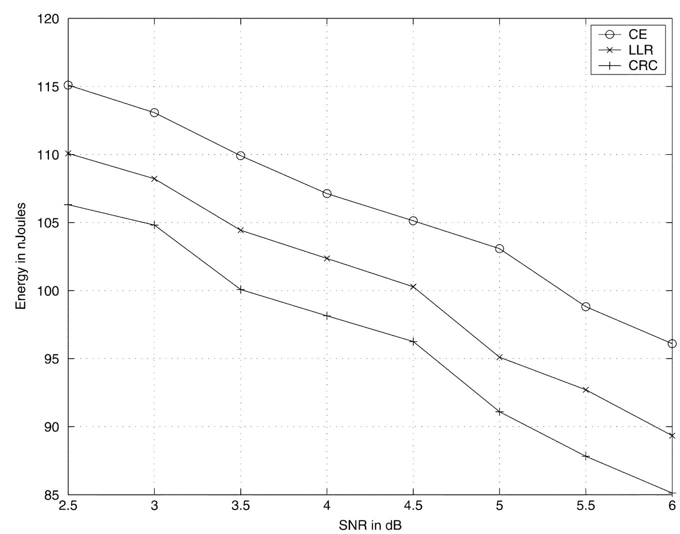

Fig. 5. Energy estimates of log-MAP for various stopping criteria on fading channels, Rician factor $K = 5, \text{Block size} = 2048$ bits.

performance as SNR increases. Figs. 4 and 5 show the energy estimates for AWGN and fading channels when all the three stopping criteria are applied on log-MAP. For fading channels, while LLR is close to CRC for smaller values of SNR, CRC clearly outweighs LLR at higher values of SNR in terms of energy consumption. For AWGN channels, CRC and LLR are close to each other for all values of SNR.

D. Sliding Window Techniques

Sliding window is a very effective technique to reduce the number of data accesses. By using smaller block size in conjunction with sliding window, the memory required to store the state and the branch metrics can be reduced. In addition, the delay can be reduced if the decoding blocks are synchronized properly.

In this technique, the data block is divided into several windows. The processing is done for each window separately. After the calculation of the forward path metrics for a window is completed, the backward recursion is started, and the soft output for the bits in the window are computed. The forward path metrics for only one window have to be stored (instead of the whole block) and the memory required to store the forward metrics is proportional to the window size (and not to the block size as in a regular log-MAP decoding algorithm). Note, however, that backward path metrics have to be calculated before the recursion begins. For a window size W, the backward recursion should start at the $W + P$ stage in order to construct the reliable backward path metrics for the Wth stage. The number of stages, P, over which the reliable backward path metrics are built is called the *re-usability factor*. In our implementation, the first iteration makes use of P stages, the second iteration $P + 1$ stages, and so on.

We investigated the effect of different re-usability factors on the performance and energy. We observed that in terms of performance, the sliding window with a reusability factor $P = 1$

TABLE II
DROP IN SNR AND ENERGY SAVINGS CORRESPONDING TO DIFFERENT INTERLEAVERS FOR LOG-MAP

Interleaver type	AWGN		Fading	
	Drop in SNR	% Energy Savings	Drop in SNR	% Energy Savings
Block	0.11	9.8	0.13	9.9
Convolutional	0.09	8.9	0.12	9.1
Random	0.12	11.4	0.17	11.9
Code matched	0.05	7.4	0.1	8.3

is better than other alternatives for both AWGN and fading channels. For instance, with re-usability factor of 1, we achieve 10%(11%) energy savings on log-MAP(SOVA) at the expense of 0.34 dB(0.49 dB) loss in SNR for a BER of 10^{-5} on AWGN channels. For fading channels, on the other hand, we achieve 11.8%(12.2%) energy savings on log-MAP(SOVA) at the expense of 0.3 dB(0.43 dB) loss in SNR for a BER of 10^{-5}. When sliding window with same re-usability factor (but higher than 1) for all iterations was used, we observed that while the energy savings were higher, the loss in SNR was disproportionately high.

E. Effect of Interleavers

The role of the interleaver is to break low weight input sequences and hence increase the code free Hamming distance or to reduce the number of codewords with small distances in the code distance spectrum. Thus, the structure of interleaver affects the performance at high SNRs. We investigated the effect of interleavers on the performance and energy consumption of Turbo decoders on AWGN and fading channels. The interleavers studied included block interleavers, convolutional interleavers, random interleavers and code matched interleavers.

Table II gives the drop in SNR in dB and energy savings for AWGN and fading channels for different interleavers. All results are with respect to a code-matched interleaver that has input patterns with weights as a multiple of 2. We observed that code-matched interleavers are the best in terms of performance. Wattch results indicate that in terms of energy consumption, S-random interleavers are the best for values of $\mathrm{SNR} > 2$ dB, where as random interleavers are the best for lesser values of SNR. Though the interleaver size also affects the performance and energy consumption of a Turbo decoder, we did not consider the effect of size as it is more predominant at lower values of SNR (≈0–2 dB).

F. Scaling of Extrinsic Information

The performance of Log-MAP and SOVA can be greatly enhanced by scaling the extrinsic information. At the end of each half iteration, if the extrinsic information is scaled by a factor of 0.63 (in case of AWGN channels) and 0.72 (in case of fading channels), before passing it to the next iteration (information exchanged between the decoder 1 and decoder 2), performance loss is reduced to 0.08 dB on AWGN channels and 0.1 dB on fading channels. Wattch results indicate energy savings of 32%(33.2%) for AWGN(fading) channels for a BER of 10^{-5} for a log-MAP based decoder.

The effect of scaling factor is to combat correlation effects in the iterative procedure. Since, practical Turbo codes rely on finite blocks, extrinsic and intrinsic pieces of information become more and more correlated with successive iterations. Thus SE results in improved performance by reducing the number of iterations (since data becomes more and more uncorrelated after every iteration). One can think of the scaling factor as a tradeoff between the need to provide the decoders with full-scale extrinsic and intrinsic information and the concern to minimize correlation effects [19].

TABLE III
DROP IN SNR AND ENERGY (SAVINGS) FOR AWGN AND FADING CHANNELS FOR COMBINATION OF SE + (RS OR RP OR SC) ON LOG-MAP BASED TURBO DECODER AT $\mathrm{BER} = 10^{-5}$

Approximation	% Energy Savings		Drop in SNR in dB	
	AWGN	Fading	AWGN	Fading
SE+RS	42.1	37.6	0.28	0.16
SE+RP	40.3	47.8	0.24	0.33
SE+SC	38.9	39.2	0.15	0.17

TABLE IV
DROP IN SNR AND ENERGY (SAVINGS) FOR AWGN AND FADING CHANNELS FOR DIFFERENT APPROXIMATIONS ON LOG-MAP BASED TURBO DECODER AT $\mathrm{BER} = 10^{-5}$

Approximation	% Energy Savings		Drop in SNR in dB	
	AWGN	Fading	AWGN	Fading
Loop Transformations	8.5	8.5	0.0	0.0
Sliding Window	11.8	10.0	0.3	0.34
Reduction in States	24.5	14.0	0.51	0.20
Reduction in Paths	22.0	29.0	0.48	0.61
Stopping Criterion	17.5	21.2	0.34	0.35
Effect of Interleaver	11.4	11.9	0.12	0.17
Scaling of Extrinsic Information	32.0	33.2	0.08	0.1

Another significant advantage is that this technique can be combined with other approximations like reduction in states (RS), reduction in paths (RP) and stopping criteria (SC) to help offset the quality degradation when RS, RP, and SC are applied individually. In Table III, we show the drop in SNR in dB and energy savings of log-MAP based Turbo decoder on AWGN and fading channels when SE is applied in combination with RS, RP, and SC. For instance, if reduction in states is used along with scaling of extrinsic information on a log-MAP based Turbo decoder, Wattch results indicate energy savings of 42.1% (37.6%) for AWGN(fading) channels at the cost of 0.28 dB (0.16 dB) drop in SNR for a BER of 10^{-5}. This validates our hypothesis that the choice of appropriate scaling factor greatly improves the performance of Turbo decoder.

G. Loop Transformations

Data loop transformations are high-level transformations (and not really approximations) and can be applied on top of all the above approximations. In [20], loop and data flow transformations have been applied to the sliding window MAP algorithm to reduce both the memory size and the number of

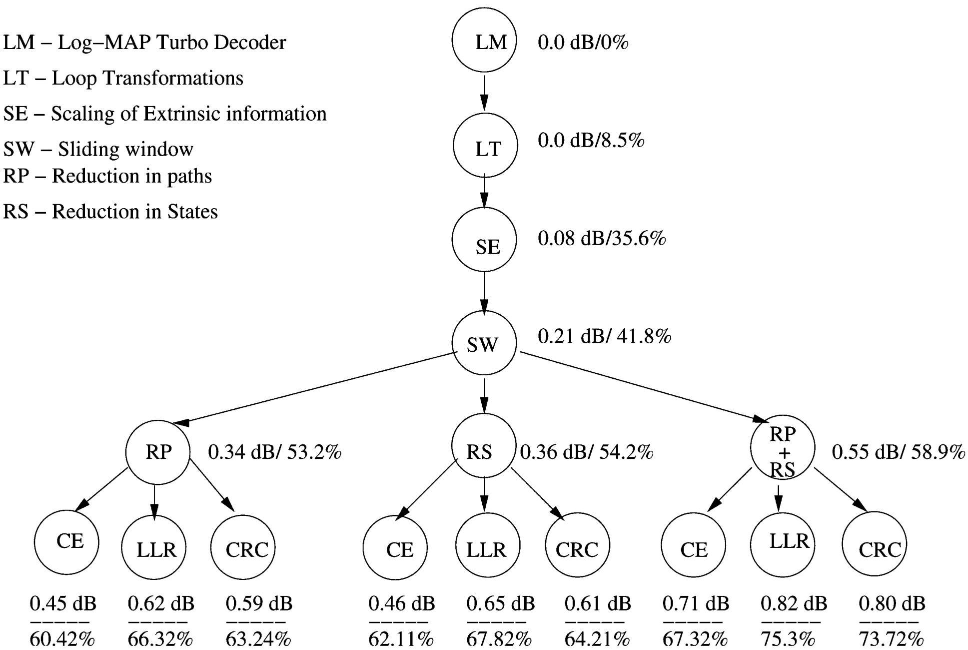

Fig. 6. Drop in SNR and energy savings in log-MAP based Turbo decoder for AWGN channels, $\text{Block size} = 2048$ bits, $\text{BER} = 10^{-5}$.

accesses. We applied global loop unrolling (by a factor of 2) on the *C*-code of log-MAP and SOVA and found that it resulted in approximately 8.5% energy savings for both AWGN and fading channels.

III. Results

A. Comparison Between AWGN and Fading Channels

In Table IV, we compare the drop in SNR in dB and energy savings of log-MAP based Turbo decoder on AWGN and fading channels when the approximations listed in Section II are applied individually. The results are for a block size of 2048 bits and a BER of 10^{-5}, with Rician factor $K = 5$. In terms of stopping criteria, we compared the best case energy savings in AWGN (that is LLR) with that in fading (that is CRC). In terms of effect of interleaver, we compared the results using code-matched interleavers that have input patterns with weights as a multiple of 2. We observed that each approach results in different energy savings and different drops in SNR for the two types of channels. Table IV. shows that scaling of extrinsic information is the clear winner in terms of high-energy savings and minimal loss of quality for both AWGN and fading channels. The largest variation is due to reduction in number of states and reduction in number of paths. For instance, for fading channels, there is significantly more degradation when the number of paths is reduced. This could be because, in fading channels, erroneous decoding is caused by disturbances that cause many values of *alpha* or *beta* at a few levels to have roughly similar values, instead of the usual wide spread of values. Similarly, for AWGN channels, there is more degradation when the number of states is reduced.

B. Quality-Energy Tradeoffs

Figs. 6 and 7 describe the energy savings and loss in SNR in dB for a BER of 10^{-5} for a combination of approximations for a log-MAP based decoder that uses the baseline code-matched interleaver (which has input patterns with weights as a multiple of two and gives no performance loss for AWGN and fading channels). Note that only a small subset of combinations are shown in these two figures. A similar quality energy tradeoff study can be done for SOVA based decoders. As the number of approximations increase, the energy savings and loss in SNR also increases. For instance, for fading channels (see Fig. 7) a combination of loop transforms (LT) and scaling of extrinsic information (SE) results in 0.1 dB loss in SNR and 37.3% energy savings. Now, if sliding window optimizations are applied on top of LT and FPE, the energy savings increase to 42.3% with 0.27 dB loss in SNR.

In case of AWGN channels (Fig. 6), the largest energy savings (75.3%) are obtained for a combination of LT, SE, SW, RS+RP, LLR resulting in a 0.82 dB drop in SNR. In case of fading channels (Fig. 7), the largest energy savings (90.2%) are obtained with a combination of LT, SE, SW, RS + RP, CRC resulting in a 0.99 dB drop in SNR. Though a loss of 0.99 dB would mean a clear performance penalty, such a combination of algorithms could be used when there is severe drain on the battery power. On the other hand, a combination of LM, LT, and SE should be chosen for low-power and high-performance oriented applications.

In the derivation of Figs. 6 and 7, we find that the drop in SNR is not additive for successive approximations. For instance, Fig. 7 shows that when we apply only sliding window technique or only scaling of extrinsic information technique separately on the log-MAP based Turbo decoder, the drops in SNR are 0.34 dB and 0.08 dB respectively. But when we apply both the techniques simultaneously, the drop in SNR was found to be only 0.27 dB $(<0.34 + 0.08)$.

Based on an analysis similar to that presented in Fig. 6 and 7, different combination of algorithms can be chosen given the permissible drop in SNR. For instance, if the user desires to

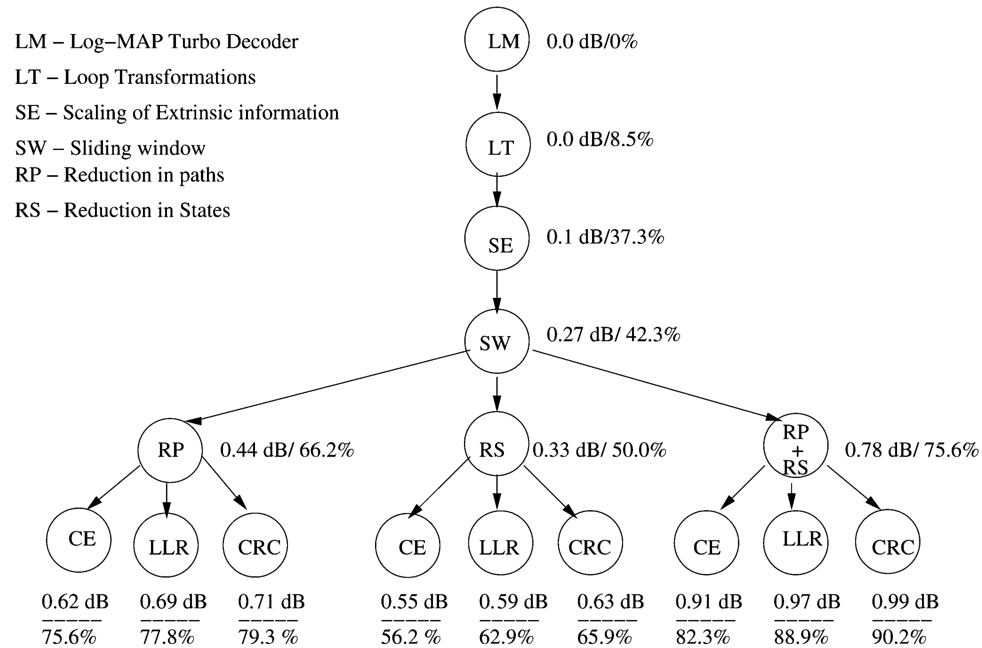

Fig. 7. Drop in SNR and energy savings in log-MAP based Turbo decoder for Fading channels, $\text{Block size} = 2048$ bits, Rician factor $K = 5, \text{BER} = 10^{-5}$.

TABLE V
CHARACTERISTICS OF EEA AND EEF BASED TURBO DECODER SYSTEMS. DROP IN $\text{SNR} < 0.35$ dB

Algorithm	Throughput	Latency	Power	Area (datapath+control+memory)
EEA	6 Mbps	4.39 msec	6.63 mW	1.07 sq.mm (0.5+0.205+0.37)
EEF	6 Mbps	5.48 msec	7.74 mW	1.12 sq.mm (0.58+0.23+0.39)

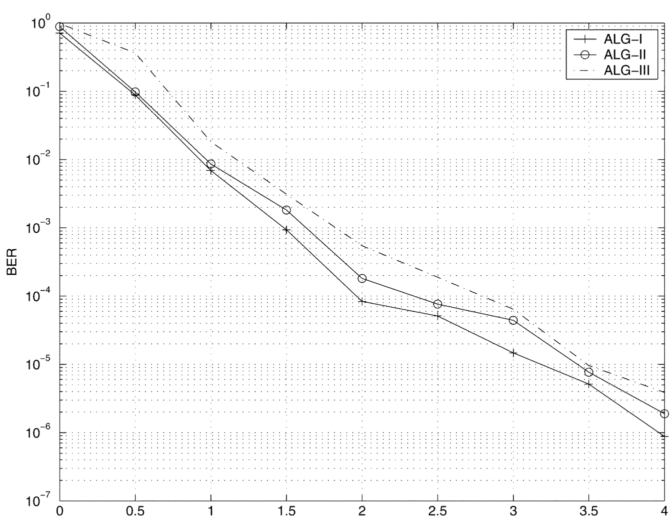

Fig. 8. Example illustrating the adaptive Turbo decoding technique.

have less than 0.1 dB drop in SNR on both AWGN and fading channels, a combination of LT and SE can be chosen. Another point to note is that the individual approximations (e.g., SC, SW, RS(RP)) can be applied in any order. While different orderings result in different amounts of drop in SNR, the maximum variation in drop in SNR is around 0.1 dB.

C. Energy-Efficient Algorithms

We next describe two algorithms with reasonably good performance ($\approx$0.35 dB drop in SNR) and high energy savings ($>$50%) for AWGN and fading channels. Such algorithms would be useful in battery operated hand-held devices. We refer to them as the energy efficient AWGN (EEA) and energy efficient rading (EEF) algorithms. The EEA algorithm uses a combination of loop unrolling, scaling of extrinsic information, sliding window and reduction in paths. The EEF algorithm uses a combination of loop unrolling, scaling of extrinsic information, sliding window, and reduction in states. We have also evaluated the performance of EEA and EEF algorithms for two different block sizes, one of 128 bits (suitable for voice applications) and the other of 2-K bits (suitable for data applications). We found that the EEA and EEF have higher energy savings for large block sizes ($\approx$24%) with lesser drop in SNR ($\approx$0.3 dB) when compared to smaller block sizes.

D. Synthesis Results

We also synthesized two Turbo decoder systems using Synopsys for a 0.13-μm CMOS technology and a supply voltage of 1.8 V. The first decoder was based on the EEA algorithm (our low power version of log-MAP for AWGN channels) as a building block, while the second one was based on the EEF algorithm (our low-power version of log-MAP for fading channels). Both the Turbo decoder systems were parameterized with the interleaver block length, width of the branch metric and the convergence length. Though the internal decoder building blocks differed slightly, the overall implementation was the same for both the systems. The datapath of both the architectures consisted of adders and multiplexers. Proper buffering was done in each stage to avoid hardware conflicts between extrinsic data

TABLE VI
COMPARISON OF THE POWER CONSUMPTIONS OF GPP AND HARDWARE IMPLEMENTATIONS. NOTATION: $A + B + C$, WHERE A, B, AND C ARE THE POWER CONSUMPTION OF THE CONTROL BLOCK, THE DATA PATH AND THE MEMORY, RESPECTIVELY

Configuration	Log-MAP(AWGN)	EEA	Log-MAP(Fading)	EEF
GPP	50.69 mW	22.7 mW	58.4 mW	29.2mW
	(17.00+23.00+11.69)	(6.59+8.69+5.42)	(21.03+24.78+12.59)	(9.87+11.55+7.88)
Hardware	34.29 mW	6.63 mW	40.34mW	7.74mW
	(9.98+18.78+5.53)	(1.45+3.89+1.29)	(11.86+22.14+6.34)	(1.67+4.76+1.31)

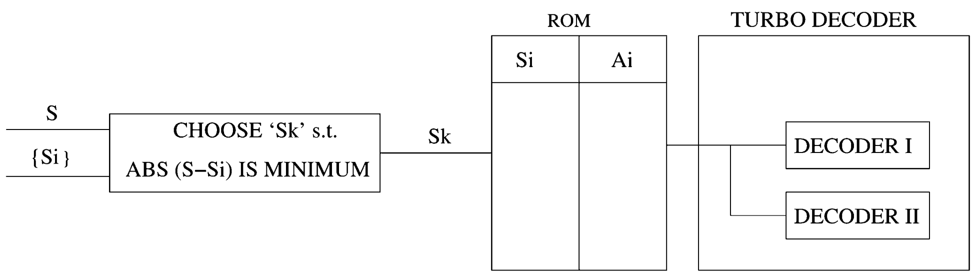

Fig. 9. Adaptive Turbo decoder.

TABLE VII
OPTIMAL CONFIGURATIONS FOR AWGN CHANNELS

Operating SNR	Optimal Configuration	Energy Savings
2.15 dB	LT+SE	35.6%
2.30 dB	LT+SE+SW	41.8%
2.45 dB	LT+SE+SW+RP+CE	60.42%
2.60 dB	LT+SE+SW+RS+CRC	64.21%
2.75 dB	LT+SE+SW+RS+LLR+Convolutional Interleaver	73.2%
2.90 dB	LT+SE+SW+RS+RP+LLR+Block Interleaver	78.9%

TABLE VIII
OPTIMAL CONFIGURATIONS FOR FADING CHANNELS.

Operating SNR	Optimal Configuration	Energy Savings
2.15 dB	LT+SE	37.3%
2.30 dB	LT+SE+SW	42.3%
2.45 dB	LT+SE+SW+RP	66.2%
2.60 dB	LT+SE+SW+RP+Random-Interleaver	72.3%
2.75 dB	LT+SE+SW+RP+CRC	79.3%
2.90 dB	LT+SE+SW+RP+RS+Convolutional-Interleaver	81.2%

from the previous iterations and the new channel data. Synopsys Design Compiler was used to characterize the soft input soft output blocks for power consumption, area and delay.

Table V shows the characteristics of the EEA and the EEF based Turbo decoder systems for a BER of 10^{-5} and a block size of 2 K. This is compatible with the third generation mobile systems [21] that require a throughput of 2 Mb/s and interleaver sizes between 46 and 5114 bits. The power consumption of the EEA based decoder is less than that of the EEF based decoder, a trend that is similar to the corresponding GPP implementation. In both the decoders, the datapath components occupy the maximum area followed by memory and control.

Table VI compares the power consumption of the EEA and the EEF algorithms implemented in hardware and GPP with the power consumption of the log-MAP based Turbo decoder. We observe, as expected, that the hardware implementation has significantly smaller power consumption than the GPP implementation for both EEA and EEF algorithms. In fact, the GPP implementation results in 54.2% (50%) power savings for EEA (EEF), compared to 80.66% (80.81%) power savings obtained by the hardware implementation. Table VI also gives the break up of the power consumption with respect to control, datapath and memory. For instance, the power consumption for control, datapath and memory blocks are 17.00 mW, 23.00 mW, and 11.69 mW, respectively, for Log-MAP based Turbo decoder (AWGN channel) implemented on a GPP. We find that the maximum power savings in EEA and EEF come from reduction in data path and memory power consumption for both GPP and hardware implementations. Note that the power consumption numbers do not include any post annotation capacitance and clock distribution.

In a real world scenario, a Turbo decoder is likely to encounter a combination of an AWGN and a fading channel. Hence, a dedicated Turbo decoder should use EEA or EEF algorithms depending on the channel conditions. If EEA and EEF algorithms are derived from the exact same combination of algorithms, a Turbo decoder can be configured to use either EEA or EEF based on the channel conditions.

IV. Adaptive Turbo Decoding

In a typical noisy environment with lot of ISI and multipath, significant energy savings can be obtained by varying the decoding algorithm dynamically based on the channel characteristics. In order to adapt the decoder to the channel characteristics, an estimate of the SNR of the signal transmitted across the channel is needed. This is done in an AWGN channel by measuring the response of the matched filter output and computing the ratio of the signal power to the noise power. In fading channels, a square pulse of known amplitude is transmitted as a pilot signal on a different channel. By time averaging the pilot signal and attenuating it with the known fading gain, the signal power and hence the SNR can be estimated.

We explain the principle of the adaptive decoding technique with the help of Fig. 9. Let the decoder be currently implementing ALG-I in order to achieve a BER of 10^{-5} at an SNR of 2.5 dB. When the SNR changes to 3 dB, both ALG-I and ALG-II achieve a BER of 10^{-5}. However, since ALG-II offers higher energy savings, the decoder would switch to ALG-II. Fig. 8 gives the block diagram of the proposed adaptive Turbo decoder. The target BER is 10^{-5} in compliance with the third generation mobile systems [21]. Let S be the SNR of the transmitted signal. For each Si, the SNR at which a BER of 10^{-5} is achieved, the approximations that resulted in the maximum energy savings, Ai, is stored in the ROM. During decoding, the SNR of the transmitted signal S is compared with all the Si's stored in the ROM and the largest $\mathrm{SNR} < S$ is chosen for the ROM table look up. This is because at a specific BER, as the SNR increases, the energy savings increase (see Tables VII and VIII). In this way, the optimal Turbo decoding configuration is chosen for the current channel conditions. This process can be repeated for every 10 pilot signals.

The size of the ROM clearly depends on the number of Si entries. To reduce the size of the ROM, we consider Si's greater than 2 dB since for AWGN channels a BER of 10^{-5} (of the order of 10^{-5}) can be achieved using the Log-MAP algorithm only if the SNR is greater than 2 dB. We also consider 0.15 dB increments in SNR though this can be reduced at the expense of larger ROM size. Tables VII and VIII show all the optimal algorithm configurations that result in a BER of 10^{-5} for AWGN and fading channels for 0.15 dB increments of SNR. The maximum SNR is limited to 2.90 dB since the optimal energy configuration corresponding to 2.90 dB is the one that can offer maximum energy savings with our approach as shown in Figs. 6 and 7. Note that the optimal energy configurations for fading and AWGN channels are different.

We now illustrate how we choose the optimal configuration for a given operating SNR. Let us assume that the operating SNR is 2.60 dB, which means that we are interested in all configurations that can operate at less than 2.60 dB SNR, while satisfying the BER requirements. For AWGN channels, the contenders are $\mathrm{LT} + \mathrm{SE} + \mathrm{SW} + \mathrm{RS} + \mathrm{CRC}$ and $\mathrm{LT} + \mathrm{SE} + \mathrm{SW} + \mathrm{RP} + \mathrm{CRC}$. Since the former configuration results in higher energy savings for the same drop in SNR we choose that. However, it is important to note that the optimal configurations differ based on the ordering and the target BER for a given block size and constraint length. Also, when CRC is used in the adaptive decoding technique, the header information needs to be transmitted to decode the information precisely.

The above analysis is very useful in channels that have extremely random noisy behaviors. In such situations, adapting the Turbo decoding to the channel properties helps in reducing the power consumption. However, in such a scenario, the time and energy penalties of reconfiguring a Turbo decoder have to be taken into account. The penalties of reconfiguring the Turbo decoder would be minimal, if this logic is implemented in a programmable processor (for instance, a DSP) than in an ASIC implementation. This is because in a programmable implementation, the only penalty is the time taken to load a new instruction set from a different location in memory. In fact, the adaptive decoder should continue to operate on the previous algorithm while the new algorithm is being loaded. The energy penalty is the memory access power consumption to load a new instruction set corresponding to the new algorithm. This energy penalty would be less compared to the total amount of energy savings in a programmable implementation.

V. Conclusion

In this paper, we showed the effect of applying different approximation techniques on the BER and energy consumption of a Turbo decoder on AWGN and fading channels. We proposed a suite of algorithms with energy-quality tradeoffs that can be utilized to choose an algorithm that matches the application requirements. We synthesized two of the algorithms (one for AWGN and one for fading channels) that have a degradation <0.35 dB. The resulting implementation consumes 6.63 mW (7.74 mW) for AWGN(fading) which is 70.79%(73.49%) lower than the corresponding GPP implementations. We also proposed a low power adaptive Turbo decoding technique for noisy channels.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding," in *Proc. 1993 Int. Conf. Communications*, pp. 1064–1070.

[2] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.

[3] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. Global Telecommunications Conf.*, vol. 3, 1989, pp. 1680–1686.

[4] P. Robertson, E. Villeburn, and P. Hoeher, "A comparison of optimal and suboptimal decoding algorithms operating in the log domain," in *Int. Conf. Communications*, vol. 2, 1995, pp. 1009–1013.

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

[6] Z. Wang, Z. Chi, and K. K. Parhi, "Area efficient high-speed decoding schemes for turbo decoders," *IEEE Trans. VLSI Syst.*, vol. 10, pp. 902–912, Dec. 2002.

[7] S. Hong and W. E. Stark, "Design and implementation of a low complexity VLSI turbo-code decoder architecture for low energy mobile wireless communications," *J. VLSI Signal Processing Systems*, vol. 24, pp. 43–57, 2000.

[8] C. Schurgers, F. Catthoor, and M. Engels, "Memory optimization of MAP turbo decoder algorithms," *IEEE Trans. VLSI Syst.*, vol. 10, pp. 305–312, Apr. 2001.

[9] G. Masera, M. Mazza, G. Piccinini, F. Viglione, and M. Zamboni, "Architectural strategies for low-power VLSI turbo decoders," *IEEE Trans. VLSI Syst.*, vol. 10, pp. 279–285, June 2002.

[10] D. Garrett, B. Xu, and C. Nicol, “Energy efficient turbo decoding for 3-G mobile,” in *Proc.2001 Int. Symp. Low Power Electronic Design*, pp. 328–333.

[11] E. K. Hall and S. G. Wilson, “Design and analysis of turbo codes on Rayleigh fading channels,” *IEEE J. Select. Areas Commun.*, vol. 16, pp. 160–174, Feb. 1998.

[12] J. Kaza and C. Chakrabarti, “Energy efficient turbo decoder,” in *Proc. 2002 Int. Conf. Acoustics, Speech and Signal Processing*, May 2002, pp. 3093–3096.

[13] D. Brooks, V. Tiwari, and M. Martonosi, “Wattch: A frame work for architectural power analysis and optimizations,” in *Proc. 27th Int. Symp. Computer Architecture*, 2000, pp. 83–94.

[14] G. Colvalope, G. Ferrari, and R. Raheli, “Reduced-state BCJR-type algorithms,” in *Proc. IEEE Int. Conf. Communications*, June 2000, pp. 460–464.

[15] S. Simmons, “Breadth-first trellis decoding with adaptive effort,” *IEEE Trans. Commun.*, vol. 38, pp. 3–12, Jan. 1990.

[16] J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of binary block and convolutional codes,” *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, Mar. 1996.

[17] Z. Wang, H. Suzuki, and K. K. Parhi, “VLSI implementation issues of turbo decoder design for wireless applications,” in *Proc. IEEE Workshop Signal Processing Systems*, 1999, pp. 503–512.

[18] O. Y. Leung, C. Y. Tsui, and R. S. Cheng, “Reducing power consumption of turbo-code decoder using adaptive iteration with variable supply voltage,” *IEEE Trans. VLSI Syst.*, vol. 9, pp. 34–41, Feb. 2001.

[19] C. Berrou, Ecole Nationale Superieure des Telecommunications de Bretagne, Brest, France, private communication.

[20] F. Maessen, A. Giulietti, B. Bougard, V. Derudder, L. Van der Perre, F. Catthoor, and M. Engels, “Memory power reduction for high-speed implementation of turbo codes,” in *Proc. IEEE Workshop Signal Processing Systems*, Sept. 2001, pp. 16–24.

[21] Third Generation Partnership Project [Online]. Available: http://www.3gpp.org

Jagadeesh Kaza received the B.E. degree (Hons.) in electrical and electronics engineering from the Birla Institute of Technology and Science, Pilani, India, in 2000, and the M.S. degree in electrical engineering from Arizona State University, Tempe.

Since Otcober 2002, he has been working as a member of the technical staff at Texas Instruments Inc., Dallas. His research interests are in the area of low-power mixed-signal design and VLSI signal-processing.

Chaitali Chakrabarti received the B.Tech. degree in electronics and electrical communication engineering from the Indian Institute of Technology, Kharagpur, India, in 1984, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, in 1986 and 1990, respectively.

Since August 1990, she has been with the Department of Electrical Engineering, Arizona State University, Tempe, where she is now a Professor. Her research interests are in the areas of low-power systems design including memory optimization, high level synthesisand compilation, and VLSI architectures and algorithms for signal processing, image processing, and communications.

Dr. Chakrabarti is a member of the Center for Low Power Electronics, the Consortium for Embedded and Inter-Networking Technologies and Connection One. She received the Research Initiation Award from the National Science Foundation in 1993, a Best Teacher Award from the College of Engineering and Applied Sciences, ASU, in 1994, and the Outstanding Educator Award from the IEEE Phoenix Section in 2001. She has served on the program committees of ICASSP, ISCAS, SIPS, ISLPED, and DAC. She is currently an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the *Journal of VLSI Signal Processing Systems*.