Algorithm and Circuit Co-Design for a Low-Power Sequential Decoder *

S. K. Singh, P. Thiennviboon, R. O. Ozdag, S. Tugsinavisut, P. A. Beerel and K. M. Chugg
Department of Electrical Engineering – Systems
University of Southern California
Los Angeles, CA 90089-2565

Abstract

The design of a sequential decoder for a convolutional code in a mobile radio environment with the objective of minimizing the average decoding effort and hence the energy consumption is presented. An efficient normalization scheme and a corresponding fast, low-energy architecture is proposed. We present the chip layout and Powermill simulation results in 0.5 μ HP14B CMOS technology. For sufficiently high SNR, the proposed design performs close to that of the optimal Viterbi decoder. The average energy consumption per decoded bit for the proposed architecture is compared against a baseline Viterbi decoder design for a simple AWGN channel. The AWGN performance characteristics are translated into average energy consumption per decoded bit for a nominal cellular system mobile unit accounting for fading and mobility. For realistic operational scenarios, the proposed architecture consumes only 7.2% as much energy as the Viterbi baseline.

1 Introduction

Power consumption in portable, battery-powered communication devices is increasingly important. In contrast to traditional design methodologies, which often aim to meet latency and fidelity requirements under the worst-case channel conditions, we suggest the approach of minimizing average computational complexity, which translates to lower average energy consumption. Traditional design techniques have led to fixed-complexity algorithms (e.g., the Viterbi Algorithm) that do a fixed amount of effort, hence energy consumption, regardless of the prevailing SNR and/or fidelity requirements. Properties of such algorithms that are traditionally viewed as advantages are regular structure, optimal decoding, and deterministic processing delay.

In some applications, such as mobile radio, the operating characteristics varying dramatically and an algorithm that minimize the average complexity may yield significant reduction in energy consumption. Although it is well-known that sequential decoding techniques require less average complexity than the Viterbi Algorithm, large variations in the required decoding effort and associated delay have made these approaches most applicable to non-real-time applications (e.g., the Pioneer deep space probe). Increasingly, real-time applications include large buffers for the purposes of interleaving, rate control, and/or network scheduling. In such systems the the problem of buffer overflow is alleviated through flow control.

In this paper we develop a novel decoder for a convolutional code based on the Fano algorithm for application in a mobile radio transceiver. We present a chip layout and system simulation results that indicate that substantial reductions in energy consumption relative to the Viterbi Algorithm can be obtained with only minor degradations in decoded the bit error rate (BER).

We characterized the effort statistics of the modified Fano algorithms for various operating environments via computer simulation. These effort statistics are translated to energy consumption per decoded bit statistics for the proposed circuit architecture based on high-level architectural analysis and energy-consumption models for both a standard Viterbi architecture and our proposed Fano architecture. This process yields a set of decoded error rate curves parameterized by energy consumption. In particular, our codesign procedure results in a hard-decision decoder for a constraint length 7 (128 states), rate 1/2, convolutional code performing within 1.0 dB of SNR relative to the Viterbi Algorithm. High-level architectural energy estimates suggest that for reasonably low decoded bit error rates (approximately 10^{-3}) we consume significantly less average energy than a comparable Viterbi chip.

A process is described by which the energy consumption and BER characteristics for the simple AWGN channel model can be translated into the analogous characteristics for an interleaved fading channel. The results are then utilized to predict the average energy consumption of the Fano and Viterbi decoders for a typical mobile user in a cell-

*This work supported in part by the National Science Foundation (NCR-CCR-9726391).

0-7803-5700-0/99/$10.00©1999 IEEE

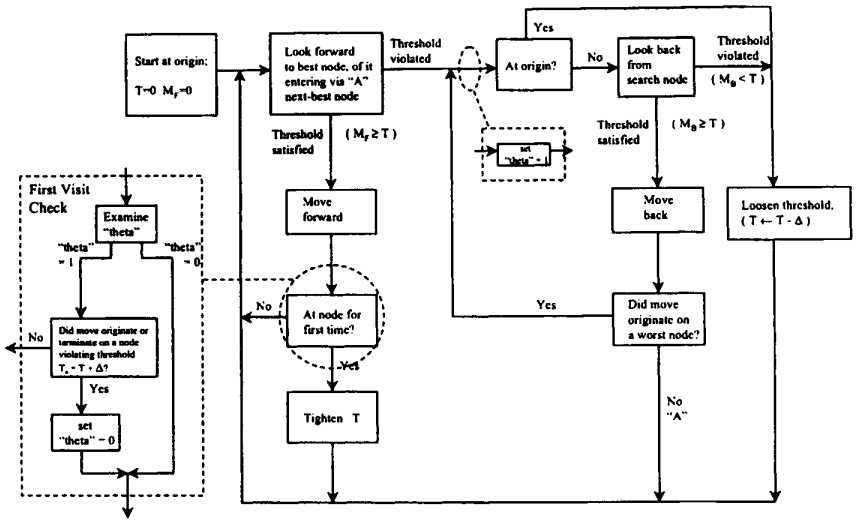

Figure 1. Flow-chart of Fano Algorithm.

based system accounting for both cell location and shadowing. The results indicate that our Fano design consumes only 7.2% of the energy per decoded bit than a comparable Viterbi design.

2 Background on the Fano algorithm

The Fano algorithm is a tree-search algorithm that achieves good performance with low average complexity. A tree comprises nodes and branches, associated with each branch is a *branch metric* (or weight, or cost). A path is a sequence of nodes connected by branches with the path metric obtained as the sum of the corresponding branch metrics. An optimal tree-search algorithm determines the complete path (i.e., from the root to leaf) with minimum path metric, while a good (suboptimal) tree-search algorithm finds a path with metric close to this minimum.

The Fano algorithm searches through the tree sequentially, always moving from one node to a neighboring node until a leaf node is reached. The Fano algorithm is a depth-first tree-search algorithm [1], meaning that it attempts to search as few paths as possible to obtain a good path. Thus, the metric of a path being considered is compared against a threshold T. The relation between T and the metric is determined by the statistics of the branch metrics (i.e., the underlying model) and the results of partial path exploration. The latter is reflected by dynamically adjusting the threshold to minimize the number of paths explored.

The key steps of the algorithm involve deciding which way to move (i.e., forward, or deeper, into the tree or backward) and threshold adjustment. Intuitively, it moves forward only when the partial path to that node has a path weight that is greater than T. If no forward branches satisfy this threshold condition, the algorithm backtracks and searches for other partial paths that satisfy the threshold test. If all such partial paths are exhausted, it will *loosen* the threshold and continue. In addition, if the current partial path metric is significantly above the threshold, it may *tighten* the threshold. Threshold tightening prevents always backtracking to the root node at the cost of potentially missing the optimal path. Moreover, a maximum *traceback depth limit* is often imposed to limit worst-case complexity. The details of the Fano algorithm are illustrated in the flow chart depicted in Fig. 1 and a more detailed explanation can be found in [5].

The decoding of a convolutional code with known channel parameters can be viewed as a tree-search problem with the optimal solution provided by the Viterbi algorithm, a breadth-first, fixed-complexity algorithm. The Fano algorithm is known to perform near-optimal decoding of convolutional codes with significantly lower average complexity than the Viterbi algorithm.

3 Low-Power Architecture Design

This section describes the efficient normalization scheme used to optimize the algorithm, our architecture at the register-transfer level, and statistics of our chip layout.

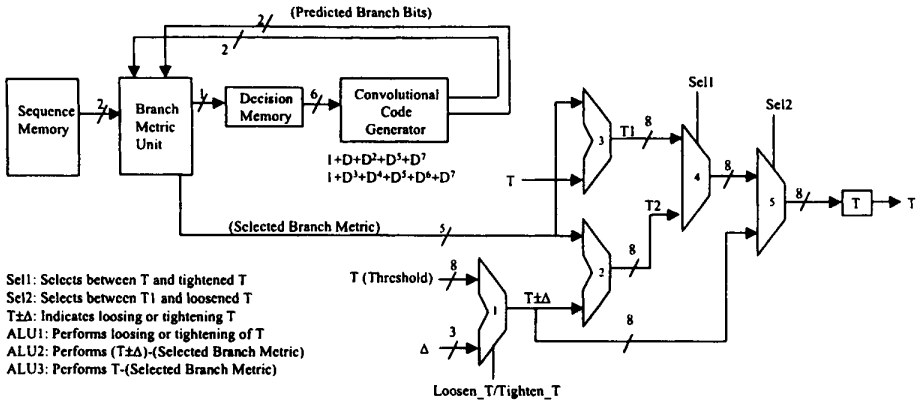

Figure 2. RTL architecture of our Fano design.

3.1 Normalization and its benefits

The basic idea behind normalization is to change the point of reference (e.g., from the origin of the tree to a current node under consideration). Normalization is often necessary to prevent hardware overflow/underflow. Interestingly, in traditional communication algorithms, such as the Viterbi algorithm, normalization often yields significant performance and area overhead that hardware designers generally avoid by using slightly larger bit-widths and modulo arithmetic [2]. In contrast, we show that using normalization in the Fano algorithm can yield a smaller, faster, and more energy efficient design.

In particular, we normalize our variables in such a way as to make the current node's metric always equal to zero. This is equivalent to subtracting the current node's metric from every variable in the algorithm, which does not change the overall behavioral algorithm. The advantages of this type of normalization in the Fano algorithm are as follows. 1) Additions involving the current metric (i.e., during the threshold check) are removed and comparisons with the current metric (i.e., during the first visit check and threshold tightening steps) reduce to a 1-bit sign check. 2) The normalization of the next threshold (subtracting the current node's metric from it) can be done by the ALU that compares the threshold with the next metric, and thus consumes negligible additional energy. 3) Lastly, the normalization enables us to work with numbers with smaller magnitudes that can be represented with fewer bits.

3.2 Register-Transfer-Level Design

Our register-transfer-level architecture is illustrated in Fig. 2. At each clock cycle, the best and next best branch metrics are both calculated using data that is stored in memory. (See [5] for more details regarding the branch metric computation.) The threshold check unit compares the error metric with the current threshold to determine if a forward move can be performed and simultaneously *speculatively* calculates two normalized next thresholds, the first assuming a forward move will be taken and the second assuming the threshold must be loosened (by subtracting Δ from T).

Based on the above results, either the move will be made and the pre-computed threshold will be stored or the threshold T will be loosened, all in one clock cycle. Additional clock cycles are needed to complete tightening the threshold if (i) a forward move is made, (ii) the first visit check is passed, and (iii) the pre-computed tightened threshold is not in the range of Δ. Fortunately, with reasonable choices of Δ, computer simulations suggest that these additional cycles of tightening are rarely needed. Similar speculative execution allows us to perform a look/move back in one clock cycle.

3.3 Chip Implementation

We used automatic placement and routing tools with a combination of synthesized and manually laid-out components in the 0.5μ HP14B CMOS process. The layout, illustrated in Fig. 3, has an area of 1.2mm by 1.8mm. Powermill was used to estimate the performance of the design. At 1.5V power supply the design successfully operated at 15 MHz and at 3.3V it successfully operated at 100MHz.

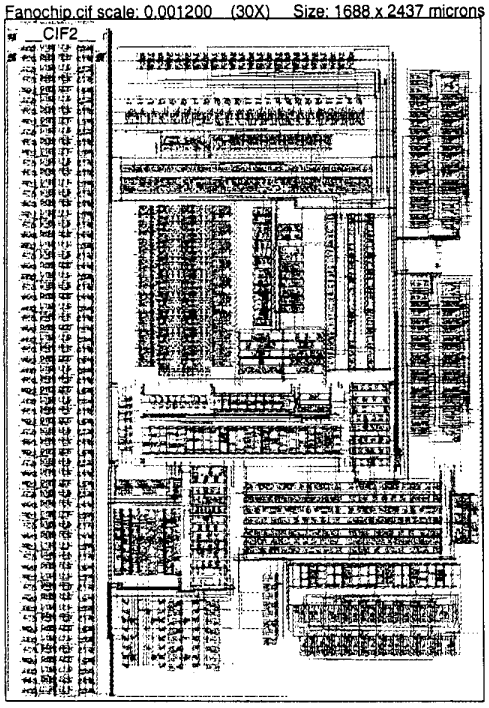

Figure 3. Layout of Fano design.

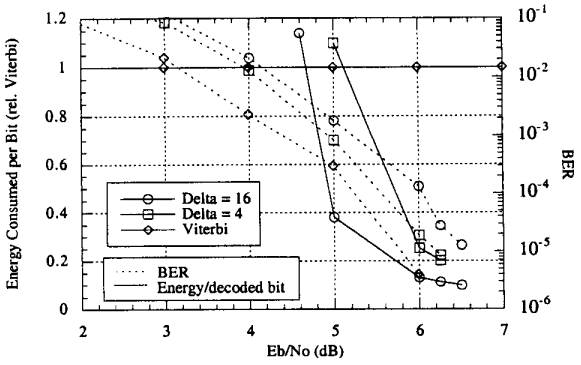

Figure 4. Energy per decoded bit and BER versus Eb/No in an AWGN channel.

4 Energy Consumption Analysis

To quantify the potential energy advantages of our design, we compared the energy of our architecture with a standard architecture for a radix-2 Viterbi algorithm using the same (7,1,2) code [2]. We simulated the critical path of this architecture in the same HP 0.5 μ technology and estimated that it can run at approximately 10 MHz at 1.5 Volts. Because we didn't have access to the chip itself, we estimated its energy consumption using the high-level capacitive models described in [3].

To make a fair comparison, we used the same capacitive models to estimate the energy consumption for our architecture. Then, to account for the fact that our algorithm is a variable complexity algorithm and thus takes a variable number of clock cycles, we used computer simulations to identify the average number of clock cycles / bit required by our algorithm and calculated the energy consumption using a weighted sum of available power supply voltages (we assumed that the values from 1.5V to 3.3V were available at 0.3V steps) which would equalize our average decoded-bit throughput.

Fig. 4 summarizes the comparison of our design and the reference Viterbi design. The left y-axis shows the energy consumed per decoded bit $\mathcal{E}$ and the right y-axis characterizes the decoded BER, with both plotted against the channel SNR (E_b/N_0) in dB. This corresponds to BPSK signaling over a memoryless AWGN channel with a packet length of 128 information bits. Two sets of curves are shown for the Fano algorithm, corresponding to $\Delta = 4, 16$. Decreasing Δ yields improved BER at the expense of greater average complexity. Reducing the maximum traceback limit has similar effects. For the curves shown, the traceback limit was set to 40 which was found to be a good compromise. The BER degradation for the Fano algorithm relative to the Viterbi algorithm is approximately 0.5 dB and 1.0 dB in E_b/N_0 over the range of interest for the $\Delta = 4$ and $\Delta = 16$ options, respectively.

While the $\mathcal{E}$ for the Viterbi design is a constant for all SNR values, $\mathcal{E}$ for the Fano design varies dramatically over the SNR range of interest. The decoding effort for the Fano algorithm is dominated by rare events which require many backward moves to decode a packet. The probability of such costly events increases as E_b/N_0 decreases. The effect on the $\mathcal{E}$ is exacerbated by the variation in supply voltage to alleviate the variation in decoding delay – *i.e.*, at lower E_b/N_0, not only is more effort expended, but each move consumes more energy. In summary, for this AWGN channel, according to approximate energy analysis, the Fano algorithm compares favorably for a desired decoded BER of less than 5.5×10^{-4} and 3.4×10^{-3} for the $\Delta = 4$ and $\Delta = 16$ options, respectively (these correspond to 1.8×10^{-4} and 4.5×10^{-4}, respectively for the Viterbi baseline).

The value of $\mathcal{E}$ for the Powermill simulation of the chip was also computed based on the same effort statistics (*i.e.*, the shape of the $\mathcal{E}$ curves will be the same as those shown in Fig. 4). For an E_b/N_0 of 5 dB, $\mathcal{E}$ is 1.5 nJ and 4.3 nJ for $\Delta = 16$ and $\Delta = 4$, respectively. At $E_b/N_0 = 6$ dB, $\mathcal{E}$ is 0.5 nJ and 1.0 nJ for $\Delta = 16$ and $\Delta = 4$, respectively. For $E_b/N_0 > 7$ dB, the energy per decoded bit is less than 0.36

nJ for $\Delta = 16$ and 0.5 nJ for $\Delta = 4$.

5 Impact of Mobile System Characteristics

Mobile radio channels are not accurately modeled by simple AWGN [4]. Instead, short-term multipath fading results in an E_b/N_0 that is random. Similarly, as the mobile unit traverses a cell-based system, one can expect significant variations in the low-term average received SNR due to path loss and shadowing. In general, random variations in the SNR will tend to make the Fano algorithm less desirable because the average complexity is dominated by the worst case SNR (*e.g.*, deep fades). However, this effect is offset by the inclusion of fade margins and the potential to eliminate decoding packets that would result in an excessive decoded BER. In this section we translate the AWGN channel results illustrated in Fig. 4 to account for a nominal mobile radio system.

A Rayleigh fading channel with sufficient interleaving and perfect channel state information at the receiver front-end is assumed. Three orders of equal power diversity with maximum ratio combining are assumed (*e.g.*, a Rake receiver (CDMA) or sequence detector (TDMA)). Thus, the input to the decoder is a sequence of independent bits, with a raw channel error rate of $\bar{p}$. Without additional soft-decision information provided to the decoder, this is indistinguishable from an AWGN channel with channel error rate equal to $\bar{p}$. Thus, translation from the the AWGN results of Fig. 4 to the fading channel can be accomplished by this correspondence on $\bar{p}$ (*i.e.*, a given $\bar{p}$ corresponds to different E_b/N_0 for the AWGN and fading channels). This yields curves analogous to those of Fig. 4 for the (short-term) average E_b/N_0.

Next, we averaged over the effects of shadowing, and cell location. For this purpose, we used an r^{-4} path-loss model and log-normal shadowing with shadowing deviation 8 dB. The curves of BER and $\mathcal{E}$ obtained by averaging over short-term fading were fit numerically and averaged over the log-normal shadowing for a fixed path-loss predicted SNR, $[E_b/N_0]_{dB,PL}$. In this process, it was assumed that the receiver had some means to estimate the prevailing SNR averaged over short-term fading. Also, it was assumed that there is some required BER, BER_{req}, that must be maintained for the source decoder to operate properly. Thus, we assume a system that does not decode if the prevailing average SNR is such that the decoded BER will exceed a cut-off BER, $\text{BER}_{c/o}$. To allow for estimation error in the process, we assumed that $\text{BER}_{c/o} = 5\text{BER}_{req}$. This process yields $\mathcal{E}$ as a function of $[E_b/N_0]_{dB,PL}$.

Finally, we averaged over cell-location using the path-loss model for a fixed outage probability at the cell boundary. Specifically, we assumed that the user was equally likely to be anywhere in a circular cell, with the outage due to shadowing at the cell boundary set to $P_{out}(R)$. Fixing

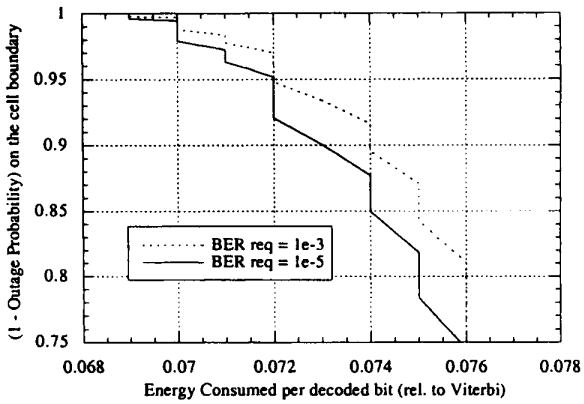

Figure 5. Average energy consumption for a mobile system.

$P_{out}(R)$ allows $\mathcal{E}([E_b/N_0]_{dB,PL})$ to be translated to $\mathcal{E}(r)$ where r is the distance from the base-station normalized to the cell radius. In this step, it is assumed that forward link power control is not used, thus providing an increasing SNR margin as the mobile unit moves inward from the cell boundary.

The final results of this analysis are plotted in Fig. 5 for the $\Delta = 16$ option. As $P_{out}(R)$ is reduced, $\mathcal{E}$ significantly reduces for the Fano decoder. For $P_{out}(R) = 0.95$, the improvement relative to the Viterbi baseline design 0.072; meaning that the proposed design consumes 13.8 times less energy than the comparable Viterbi decoder. Note that, because of the large SNR margin typically allocated to meet the coverage requirements, much of the time the mobile unit observes a very high SNR. Thus, under the assumptions described above, the Fano algorithm performs little backtracking for most of the packets decoded. This may be seen since, if only forward moves are taken (*i.e.*, no noise), our design consumes 0.067 times the energy of the Viterbi decoder.[1] This also explains why there is little difference between the two values of BER_{req} shown. Note that the piecewise linear nature of the curves is an artifact of the discrete voltage scaling levels assumed. Based on the Powermill simulations, therefore, our design would consume less than 0.5 nJ per decoded bit on average in such an operational scenario.

Finally, the degradation in E_b/N_0 can be accounted for in this comparison by considering the serviceable cell radius. Specifically, the performance degradation will result in a smaller cell radius relative to that associated with a Viterbi

[1] This relative energy assumes both designs are running at 1.5V despite the fact that at this supply voltage with no noise our design has 50% higher throughput than the Viterbi algorithm. To equalize throughputs, we can operate the Viterbi design at 1.7V which translates to our design consuming 0.052 times the energy of the Viterbi design.

decoder. Based on the r^{-4} path-loss model assumed, the 0.5 dB and 1.0 dB degradations in E_b/N_0 for the $\Delta = 4$ and $\Delta = 16$ versions, respectively, translate to a reduction in the radius of coverage of 3% and 6%, respectively.

6 Concluding Remarks

Using normalization and speculative data execution, we built a design which executes one forward or backward move of the Fano algorithm in one clock cycle with relatively low energy consumption. Compared to a baseline architecture for the Viterbi algorithm, our design was found to operate with a performance degradation of less than 1 dB in SNR and to have significantly lower energy consumption for an AWGN channel operating with a decoded BER less than approximately 10^{-3}. These results were translated to account for variations in the operating SNR and required BER for a nominal cell-based mobile communication system. The results suggest that, our Fano-based design can provide an average energy reduction of approximately 14 relative to the baseline Viterbi decoder in such systems for reasonable outage probabilities.

The results suggest several future research directions. For example, we are currently working on an asynchronous implementation of this design in which the typical on-chip operations are further optimized, promising significantly higher speeds and energy efficiency. In addition, the tradeoff between varying the supply voltage to approximately equalize the average decoding delay and building external buffers is another interesting direction for future study.

Acknowledgments

We would like to acknowledge R. Chokkalingam, S. Kim, and the students in the University of Southern California EE577b classes of Fall'98 and Spring'99 for their numerous contributions to this project.

References

[1] J. B. Anderson and S. Mohan. Sequential coding algorithms: A survey cost analysis. *IEEE Trans. on Communications*, COM-32:169–176, Feb. 1984.

[2] P. J. Black. *Algorithms and Architecturees for High-Speed Viterbi Decoding*. PhD thesis, Stanford University, 1993.

[3] A. P. Chandrakasan and R. W. Brodersen. *Low Power Digital Design*. Kluwer Academic Publishers, 1995.

[4] T. S. Rappaport. *Wireless Communications*. Prentice Hall, 1996.

[5] J. M. Wozencraft and I. M. Jacobs. *Principles of Communication Engineering*. John Wiley and Sons, 1965.