

 Open access • Proceedings Article • DOI:10.1109/VETEC.1993.510966

Soft-output Viterbi decoding: VLSI implementation issues — [Source link](#)

O.J. Joeressen, M. Vaupel, H. Meyr

Institutions: RWTH Aachen University

Published on: 18 May 1993 - Vehicular Technology Conference

Topics: Soft output Viterbi algorithm, Viterbi decoder, Viterbi algorithm, Very-large-scale integration and VHDL

Related papers:

- [Optimal decoding of linear codes for minimizing symbol error rate \(Corresp.\)](#)
- [A Viterbi algorithm with soft-decision outputs and its applications](#)
- [Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1](#)
- [Optimal decoding of linear codes for minimizing symbol error rate](#)
- [A low complexity soft-output Viterbi decoder architecture](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/soft-output-viterbi-decoding-vlsi-implementation-issues-1g9tmxzjlf>

Soft-Output Viterbi Decoding: VLSI Implementation Issues

Olaf J. Joeressen, Martin Vaupel, Heinrich Meyr

Aachen Univ. of Technology (RWTH), Lab. for Integrated Systems for Signal Processing
IS2-611810, Templergraben 55, 5100 Aachen, Germany
Phone: +49-241-807632, Fax: +49-241-807631, email: joeresse@ert.rwth-aachen.de

Abstract

During the last few years decoding algorithms that make not only use of soft quantized inputs but also deliver soft decision outputs have attracted considerable attention due to additional coding gains obtainable in concatenated systems. A prominent member of this class of algorithms is the Soft-Output Viterbi Algorithm (SOVA). This paper is concerned with implementation parameter effects in SOVA decoding that are related to considerable variations in area consumption of VLSI implementations. Namely the quantization of the reliability values inside the survivor memory unit, the depth of reliability updating and the effect of a simplified update rule for the reliability values are investigated. Results of extensive simulations are presented. Area estimates obtained by logic synthesis from VHDL descriptions are given to show how these parameters translate into the area consumption of VLSI implementations.

1 Introduction

It is well known that parameters such as wordlengths, tap count, etc. have a high impact on VLSI implementations because area consumption and throughput are often directly dependent on such parameters. Thus system designers have to attempt the optimization of such parameters in order to obtain cost effective and competitive solutions. While in certain cases good estimates may be found analytically, simulation of the parameter effects is often indispensable. This is especially the case if nonlinear modifications of the original algorithm are applied for the sake of implementation advantages. This paper is concerned with implementation parameter effects in decoding with the Soft-Output Viterbi Algorithm (SOVA).

The SOVA, a modification of the Viterbi algorithm [1], was developed by Hagenauer, Höher and Huber [2,3]. The algorithm allows not only for finding the most likely path sequence in a finite state markov-chain (like the Viterbi algorithm) but in addition delivers a reliability value for each decoded bit. This reliability value can then be used to improve transmission systems if either concatenated decoding or coding over channels with intersymbol interference (ISI) is the applied transmission scheme. In these schemes the signal is first decoded or equalized and subsequently an *outer* decoder completes processing. The *inner* decoder/equalizer usually outputs hard decisions sometimes augmented by erasure information. Thus soft decision decoding cannot be used in the outer decoder.

Given that the inner processing step is performed by means of the Viterbi algorithm, using SOVA leads to additional coding gains in the above mentioned schemes, since the outer decoder can use the reliability information by means of soft decision decoding [4]-[6].

For standard (hard deciding) Viterbi decoding the effect of the implementation parameters are well known and simulation as well as analytical methods have been used for their evaluation [7,8]. For SOVA decoding, the publications available [2,5] present only few results on implementation parameter effects. The aim of this paper is to clarify the effect of the most important parameters in order to be able to actually design a system employing SOVA decoding as the inner processing step.

Predominant algorithm parameters for VLSI implementations of SOVA decoders are the quantization of the soft decision values and the *update depth* which affect the quality of the soft decision values. Their optimization is indispensable in order to obtain a cost effective VLSI implementation [9]. In section 6, area estimates obtained by logic synthesis from VHDL descriptions are given which show how these parameters translate into the area consumption of VLSI implementations. Furthermore the effect of a simplified formula for likelihood updating is investigated.

Several performance criteria have been proposed [2,10]. Below we present simulation results on channel capacity and cutoff rate which allow for the determination of the maximally obtainable SNR gains. Since these gains may not be obtainable in real transmission systems, bit error rate simulations for concatenated Viterbi decoding with an outer code of constraint length three and rate 1/2 were carried out [4]. Sufficient interleaving was applied to remove the statistical dependencies between consecutive output values of the inner decoder. Please note that the outer decoder is used to evaluate the quality of the reliability estimate only, since the low overall code-rate of 1/4 limits the applicability of this scheme. The assumed modulation scheme is BPSK over a channel with additive white gaussian noise.

2 The Soft Output Viterbi Algorithm

In this section we will briefly introduce SOVA. Please refer to [2] for a thorough description.

The Viterbi algorithm finds the most likely sequence of state transitions of a finite state markov-chain by assigning a transition metric to each state transition (a branch in the trellis) and by selecting the path (state sequence) with the best sum of the transition metrics in each decoding cycle and for all states. These decisions are taken in the so called add compare select unit (ACSU) which accumulates the transition metrics recursively and outputs the decision about the best path for each state and each trellis cycle. These decisions are then processed in the survivor memory unit (SMU) of the decoder which keeps track of the history of decisions. Consequently the content of the SMU allows for the reconstruction of the

paths that are associated with the states. The problem of finding the most likely path through the trellis can then be solved by tracing the paths back in time until they have all merged into one path.

Similar to Hagenauer's notation we write the probabilities of the competing paths at state s as

$$\text{Prob}\{\text{path } m, \text{state } s\} \sim e^{-\Gamma^{(m,s)}} \quad (1)$$

where $\Gamma^{(m,s)}$ is the sum metric of the m 'th path that ends at state s . For rate 1/2 codes the decision is taken between two competing paths $\Gamma^{(1,s)}$ and $\Gamma^{(2,s)}$. Without loss of generality we assume path 1 to be the one which survives. The probability of selecting the wrong path as the survivor of state s at time k is then

$$P_{s,k} = \frac{e^{-\Gamma^{(2,s)}}}{e^{-\Gamma^{(1,s)}} + e^{-\Gamma^{(2,s)}}} = \frac{1}{1 + e^{\Delta_s}} \leq \frac{1}{2} \quad (2)$$

$$\text{with } \Delta_s = \Gamma^{(2,s)} - \Gamma^{(1,s)} \geq 0$$

In this notation the Viterbi algorithm has made an error with probability $P_{s,k}$ on all information symbols where competing paths that end in state s are labeled by different symbols. Given that the symbols at time $k-j$ differ we can thus update the loglikelihood ratio $\hat{L}_{s,k-j}$ of the symbol (of the survivor of state s at time $k-j$) to be erroneous [2].

$$\hat{L}_{s,k-j} := f(\hat{L}_{s,k-j}, \Delta_s) := \frac{1}{\alpha} \log \frac{1 + e^{(\alpha \hat{L}_{s,k-j} + \Delta_s)}}{e^{\Delta_s} + e^{\alpha \hat{L}_{s,k-j}}} \quad (3)$$

Where α is a constant which depends on code, modulation and SNR. The update operation is required for all the survivors that are associated with the states.

In figure 1 an example of an update operation in the trellis diagram for the survivor associated with state 1 is given (constraint length $K = 3$ and $S = 4$ states).

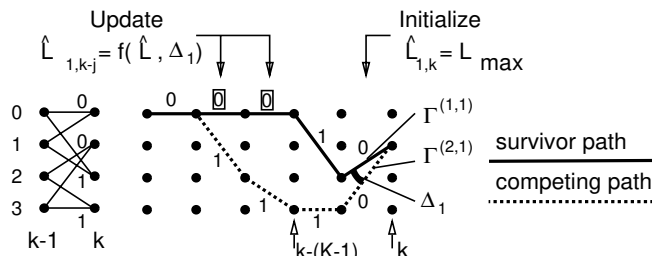


Fig. 1: SOVA Example

The path with metric $\Gamma^{(1,1)}$ is selected as the survivor for state 1. The information symbols of the survivor differing from the symbols of the competing path are framed. Only the likelihood values associated with these symbols (upon which a *relevant decision* is made) are updated. The likelihood value $\hat{L}_{1,k}$ of the newly chosen symbol is initialized to a value L_{max} that corresponds to the highest possible reliability of the symbol.

3 Effect of the 'Minimum' Rule

If update processing is implemented according to (3) the likelihood value is dependent on the SNR whereas the

decisions between the competing paths are independent, as long as the SNR is slowly varying. Consequently the metric computation unit must include an estimator for the SNR which stands in contrast to standard Viterbi decoder implementations. In [2] an approximation to (3) was proposed. This 'Minimum' rule is given by:

$$f(\hat{L}_{s,k-j}, \Delta_s) = \text{Min} \left(\hat{L}_{s,k-j}, \frac{\Delta_s}{\alpha} \right) \quad (4)$$

We refer to Δ_s/α as Δ'_s below. In contrast to the values Δ_s the values Δ'_s are independent of the SNR [2]. It turns out that for slowly varying SNR the squared euclidean distance between received signal and reference point can be used as metric increment for each trellis step. The difference between the competing metrics $\Gamma^{(m,s)}$ is then equivalent to the likelihood value Δ'_s except for a constant factor. The simplified update rule of equation (4) yields implementation advantages and does not require the estimation of the SNR.

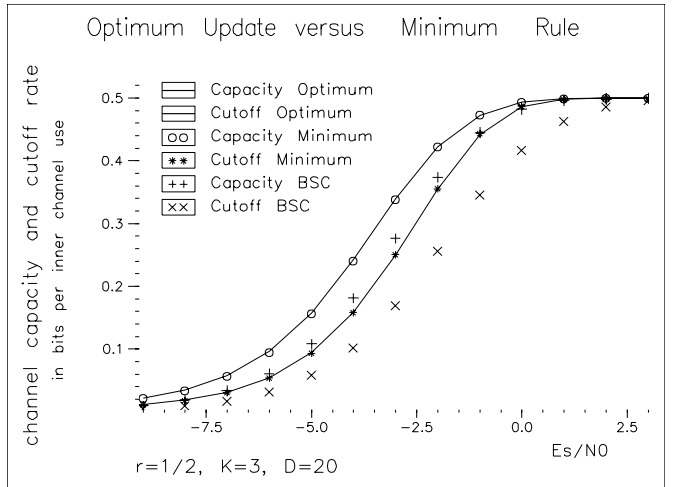


Fig. 2: Capacity and Cutoff Rate with 'Minimum' Rule

Figure 2 summarizes results of simulations of the channel capacity and cutoff rate. Results are drawn for the optimum update rule (3) and the 'Minimum' rule as well as for a hard deciding decoder that may be interpreted as a binary symmetric inner channel (BSC). Neither in capacity simulations nor in BER simulations have performance losses been observed for codes of rate 1/2 and 2/3. Thus the 'Minimum' rule is the method of choice and all other results in this paper include the application of the 'Minimum' rule.

4 Effect of limited update depth

Several approaches to reduce the complexity of the SOVA have been suggested by Hagenauer and Höher [5,2]. We found reasonable results only for approaches which update either all surviving paths or only the globally best survivor. Implementation aspects for these cases have been discussed in [9]. We found performance degradations for concatenated systems even compared to systems with a hard deciding inner decoder for all approaches that rely on updating only the survivors of some states.

It is known that the survivor depth D of a Viterbi decoder may be set to roughly five times the constraint length

K without significant performance losses [7,8]. While the original formulation of the SOVA requires the path comparison and update operation for the depth of the SMU, limiting the update depth seems to be an attractive way of reducing the computational complexity of the SOVA. Simulations of the channel capacity [10] suggest that choosing roughly twice the constraint length for the update depth is close to optimum. We did extensive simulations for various constraint lengths in concatenated decoding systems. Figure 3 shows simulation results on this topic. Note that the abscissa of figure 3 is scaled in multiples of the constraint length K .

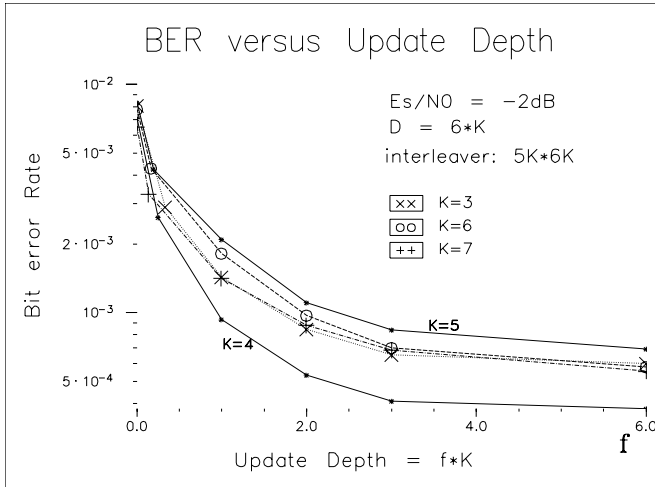


Fig. 3: BER versus Update Depth ($r=1/2$)

Two facts become clear from figure 3. First, choosing three times the constraint length as update depth is in fact close to optimum. Second, the performance degrades gracefully if further shortening of the update depth is applied. Thus system designers may use this parameter to trade complexity versus performance.

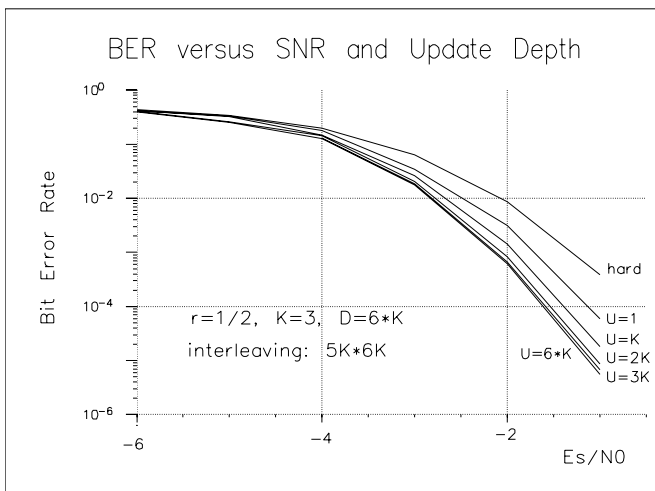


Fig. 4: BER versus SNR and Update Depth ($r=1/2$)

Figure 4 shows a typical performance result with respect to SNR and update depth U for a code with rate $1/2$. For codes with rate $2/3$ the required update depth is higher. Figure 5 shows that at least $U = 4 * K$ is

required for reasonable results. This corresponds to the required survivor depth for rate $2/3$ codes which is roughly $D = 8 * K$.

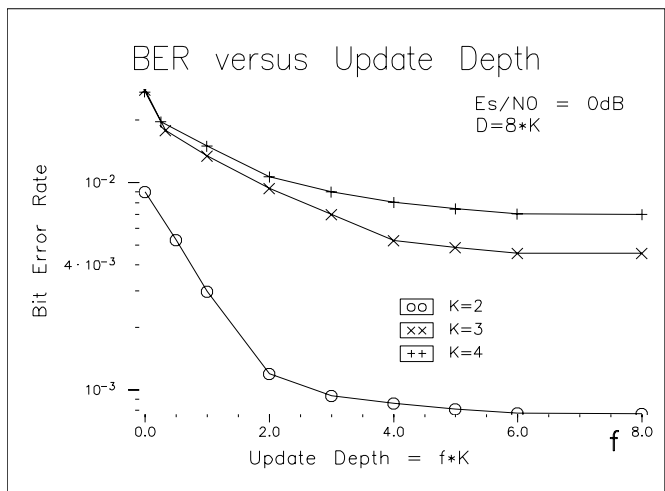


Fig. 5: BER versus Update Depth ($r=2/3$)

5 Quantization of the likelihood values

The quantization of the likelihood values has a big impact on the implementation if a VLSI implementation is required [9] because it affects the complexity of the update circuits and the storage requirements. To quantize efficiently we need to know the distribution of the update values along the final survivor path since only these metrics affect the output of the SOVA decoder.

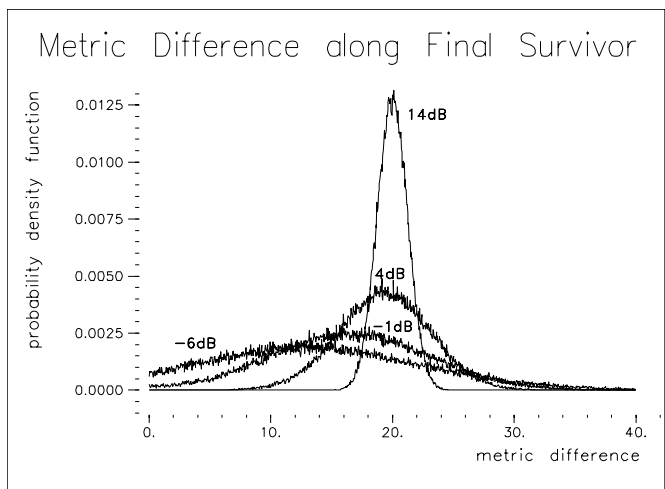


Fig. 6: PDF of Metric differences for varying SNR

Due to the application of the minimum rule we assume the path metrics to be computed as squared euclidean distances. Consequently we expect the metric difference between the competing paths along the final survivor path to be close to four times the free distance d_{free} for high SNR because the shortest error path dominates in this situation (and the euclidean distance between the possible signal point is two in BPSK). However, we are interested in the actual distribution and the distribution

for low SNR as well. A representative result is shown in figure 6 for a code with rate 1/2 and free distance five.

It is interesting that the distribution of the update values does not center around $4 * d_{free}$ for low SNR but at lower values. The distribution suggests quantizing linearly in an interval $[0, k * 4 * d_{free}]$ with k not much bigger than one. Our simulation results led to $k = 1$ as a nearly optimum choice. Choosing $k < 1$ led to performance degradations. Figure 7 shows typical results for BER and cutoff rate simulation for a rate 1/2 code, varying k and two and three bit quantization.

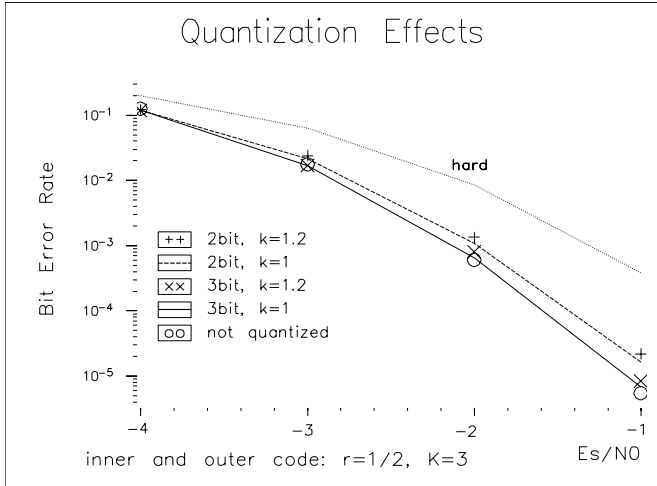


Fig. 7: BER for varying Likelihood Quantizations

It is obvious that three bit quantization is sufficient and even two bit quantization is close to optimum which was verified in cutoff rate simulations as well.

6 Synthesis Results

We did some synthesis examples using VHDL descriptions of the survivor memory units and the SYNOPSIS Design CompilerTM to estimate the area of a register exchange SOVA-SMU. The target library was the $1\mu\text{m}$ semi-custom library from European Silicon Structures (ES2). The area estimates below are accumulated cell area multiplied by a factor of 2.5 which accounts for wiring overhead. We did the examples for $n_s = 2, 3$ and 4 and forced optimization for smallest design. For this constraint the operation speed of a complete decoder is still limited by the add compare select unit (ACSU).

Tab. 1: Area for Register Exchange SOVA-SMUs

U	area ($n_s = 2$)		area ($n_s = 3$)		area ($n_s = 4$)	
1	10.3	mm ²	13.7	mm ²	17.2	mm ²
$2 * K$	12.8	mm ²	17.5	mm ²	22.1	mm ²
$5 * K$	17	mm ²	23.7	mm ²	30.3	mm ²

Tab. 2: Area estimates for Two-Step SOVA-SMUs

U	area ($n_s = 2$)		area ($n_s = 3$)		area ($n_s = 4$)	
1	10.7	mm ²	10.8	mm ²	13.3	mm ²
$2 * K$	12	mm ²	12.9	mm ²	15.6	mm ²
$5 * K$	16.1	mm ²	16.48	mm ²	19.3	mm ²

Table 1 gives an overview on the synthesis results for a straightforward register exchange implementation of the

SOVA-SMU while table 2 gives figures for the two-step formulation of the algorithm [9,10]. The figures are given for constraint length K set to five and survivor depth set to $6 * K$. They include the effort to normalize the metric difference and to reduce their wordlength to n_s . It becomes clear from the tables, that choosing parameters off the optimum point results in a considerable area penalty. Furthermore the two-step architecture provides better results. However, the figures represent results for a high throughput parallel architectures. At lower throughput the register exchange architecture will provide better results since it requires less storage.

7 Conclusion

In this paper implementation parameter effects for Soft-Output Viterbi Decoders for rate 1/2 and rate 2/3 codes have been investigated. We have shown by computer simulation, that an update length of two or three times the constraint length is sufficient to achieve nearly optimum performance for rate 1/2 codes. For rate 2/3 codes approximately four times the constraint length is sufficient. Furthermore the required quantization of the likelihood values was determined and it was pointed out that the effects of the application of the minimum rule for update processing are negligible. Finally area estimates for high-speed VLSI implementations were given.

References

- [1] G. Forney, "The Viterbi Algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, March 1973.
- [2] J. Hagenauer and P. Höher, "A Viterbi Algorithm with Soft Outputs and Its Application," in *Proceedings of the IEEE Global Communications Conference GLOBECOM*, pp. 47.1.1–47.1.7, Nov. 1989.
- [3] J. Huber and A. Ruppel, "Zuverlässigkeitsschätzung für die Ausgangssymbole von Trellis-Decodern," *Archiv für Elektronik und Übertragung (AEÜ)*, vol. 44, pp. 8–21, Jan. 1990. in German.
- [4] J. Hagenauer and P. Höher, "Concatenated Viterbi-decoding," in *Fourth Swedish-Soviet International Workshop on Information Theory*, (Gotland, Sweden), pp. 29–33, Lund: Studentlitteratur, Aug. 1989.
- [5] P. Höher, "TCM on Frequency-Selective Fading Channels: a Comparison of Soft-Output Probabilistic Equalizers," in *Proceedings of the IEEE Global Communications Conference GLOBECOM*, pp. 401.4.1–401.4.6, 1990.
- [6] T. Woerz and J. Hagenauer, "Iterative decoding for multilevel codes using reliability information," in *Proceedings of the IEEE Global Communications Conference GLOBECOM*, pp. 1779–1784, December 1992.
- [7] J. A. Heller and I. M. Jacobs, "Viterbi Decoding for Satellite and Space Communication," *IEEE Transactions Communications*, vol. COM-19, pp. 835–848, Oct. 1971.
- [8] F. Hemmati and D. Costello, "Truncation Error Probability in Viterbi Decoding," *IEEE Transactions Communications*, pp. 530–532, May 1977.
- [9] O. Joeressen, M. Vaupel, and H. Meyr, "VLSI Architectures for Soft-Output Viterbi Decoding," in *Proceedings of the Int. Conf. on Application Specific Array Processors*, (Oakland), pp. 373–384, August 1992.
- [10] J. Hagenauer and P. Höher, "private communication, spring 1992."