A Viterbi Decoder with Efficient Memory Management

Chanho Lee

This paper proposes a new architecture for a Viterbi decoder with an efficient memory management scheme. The trace-back operation is eliminated in the architecture and the memory storing intermediate decision information can be removed. The elimination of the trace-back operation also reduces the number of operation cycles needed to determine decision bits. The memory size of the proposed scheme is reduced to 1/(5×constraint length) of that of the register exchange scheme, and the throughput is increased up to twice that of the trace-back scheme. A Viterbi decoder complying with the IS-95 reverse link specification is designed to verify the proposed architecture. The decoder has a code rate of 1/3, a constraint length of 9, and a trace-forward depth of 45.

Keywords: Viterbi decoder, memory management, VLSI.

Manuscript received Jan. 15, 2003; revised Nov. 6, 2003.

This work was supported by the Soongsil University Research Fund.

Chanho Lee (phone: +82 2 820 0710, email: chanho@e.ssu.ac.kr) is with the Department of Electronic Engineering, Soongsil University, Seoul, Korea.

I. Introduction

Channel coding has been an important issue in communication systems. Channel coding has the ability to detect and correct errors caused by noise on the channel. The bit-error-rate (BER) can be reduced without increasing the signal power since the transmitted data carry redundancies that are used to detect and correct errors. This coding skill is useful in transmission on finite power channels such as general switched telephone networks. The convolutional coding method has been preferred to the block coding method in noisy channel environments. Data are encoded using the relationship between the current bit and the past bits in the convolutional coding methods.

The Viterbi algorithm is a representative decoding method for convolutional coding. It is widely used in communication and signal processing to achieve low-error-rate data transmission. The Viterbi decoding method uses the maximum likelihood decoding (MLD) algorithm, which finds the most likely pattern from the received data, and is known as the optimum decoding method [1]. When erroneous L-bit data are received, the closest codeword is selected using the MLD algorithm. The pattern indicates the state of the encoder in the transmitter. Branch metrics (BMs) and path metrics (PMs) are calculated for MLD in the Viterbi algorithm. The BM represents the similarity of the states involved in a state transition for one depth of a trellis diagram. The PM is the summation of BMs. The soft decision scheme, in which the Euclidean distance is calculated, is preferred to the hard decision scheme, in which the Hamming distance is calculated, since the BER of the soft decision scheme is better by 2 dB than that of the hard decision scheme in spite of the complexity in implementation [2], [3].

There are two common implementation methods to determine the survivor path: register exchange [4] and trace-back [5]. The register exchange method obtains the decoding data using the multiplexers and dual port memory containing all decision values through the trace-forward depth without the trace-back process. The register exchange method needs the same number of multiplexers and dual port memory as the number of states multiplied by the survivor path depth, and they are activated every cycle to update the data in memory. It results in considerable power consumption by memory. Therefore, the trace-back method is preferred. In the trace-back method, the decision bits from the add-compare-select (ACS) units are simply stored in survivor path memory in sequence and are used for trace-back after the survivor path is determined. The trace-back of the survivor path memory is required to determine the decision bit while the decision bit is determined as soon as the survivor path is determined in the register exchange method. Since the next starting state is determined after obtaining the decoding data, the trace-back scheme cannot begin the next decoding process until the decoded data is determined through the trace-back process [3], [6].

However, all intermediate data of the decoding sequence are not necessarily stored to determine the decision bit for the continuous stream of input data. We need only the decision bits of the first stage and the final PMs of the states. The modified architecture stores only the first decision bits in memory and discards other decision bits after the survivor path is determined at each stage.

I propose a new implementation architecture, the modified register exchange (MRE) method. The final decoding bit is determined as soon as the trace forward is finished using the algorithm of the conventional register exchange method, and the amount of memory is greatly reduced by eliminating the survivor path memory except for the first trace-forward stage.

II. Modified Register Exchange Method

In order to understand the operation of the modified register exchange scheme, it is better to review the operation of the trace-back scheme and the register exchange scheme. Figure 1(a) shows a simple Trellis diagram with four states: S0-S3. For simplicity, only the survived paths are shown at each stage. The initial state is S1 and the final PM value of the survivor path is 1. The survivor path is shown in gray thick line. The state transition with input bit '0' and '1' are denoted by a solid line and dotted line, respectively. We can see that the decoded bit is '1', and the next initial state is S3.

Figure 1(b) shows how the decoded bit and the next initial state are obtained by the trace-back operation. Select bits '1' for the upper branch and '0' for the lower branch are stored in 4×4 bit memory at each stage. The final PM values determine the survivor branch, and the survivor path is traced back using the select bits. The destination is S3 at t8, and it will be the initial state for the next trace-forward operation. Once we reach S3 at t8, we can know that the decoded bit is '1'. The next decoding

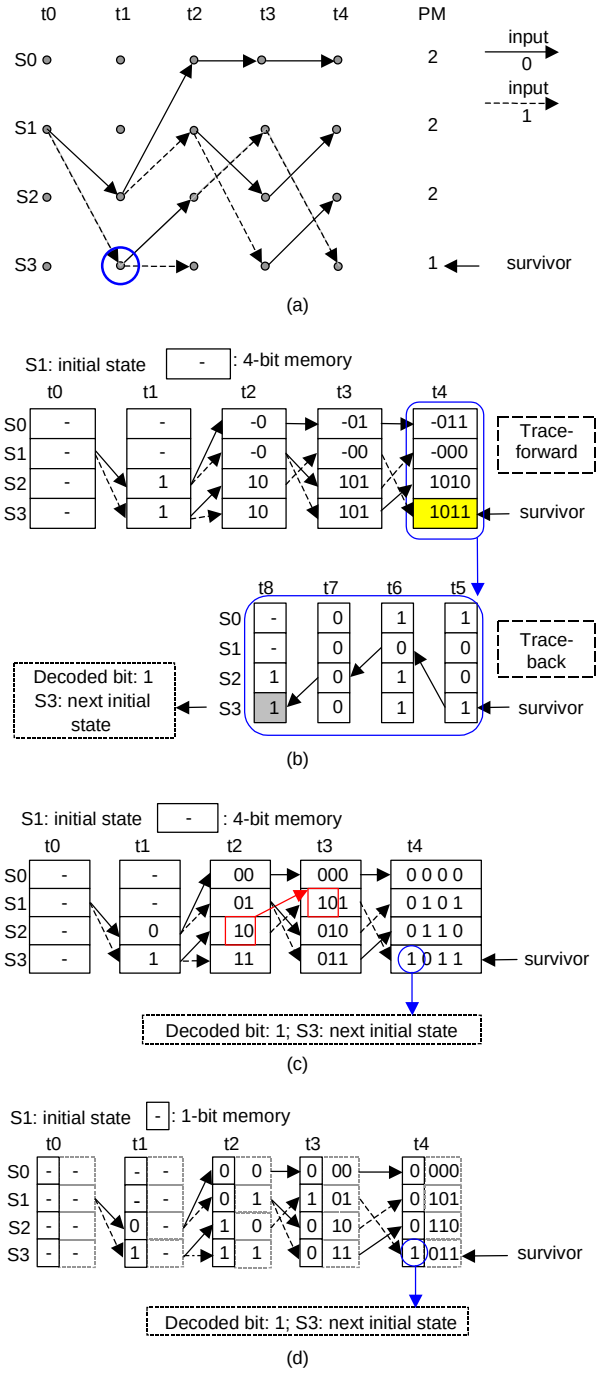

Fig. 1. Decoding process of three Viterbi decoding methods: (a) a simple Trellis diagram, (b) trace-back method, (c) register exchange method, and (d) modified register exchange method.

cycle starts with the initial state S3 and a new input bit.

Figure 1(c) shows the decoding process by the register exchange method, which also requires 4×4 bit memory. Partially decoded bits are stored here instead of the select bits. They move in memory according to the survivor branch (as shown in squares at t2 and t3) and a new decoded bit is added at each stage. Therefore, all the contents of the memory are changed at every stage. After the survivor path length, the survivor branch is determined by the final PM values. Since the chosen memory location contains the tentative decoded bits according to the survivor path, the trace-back operation is not necessary. The MSB is the final decoded bit and the state corresponding to the smallest PM value is the next initial state. If no more input bits are received, the other tentative decoded bits in the memory are the final decoded bits. The latency to obtain the decoded bit is reduced by half at the expense of the increased power consumption to update all the contents of the memory at every stage.

If the input bits are received continuously or the number of input bits is larger than the survivor path length, the decoding process must be repeated after one decoded bit is obtained. The tentative decoded bits are discarded except the MSB, and the tentative decoded bits and PM values are calculated again with the new initial state and an updated set of input bits. "1011" in the survivor path memory in Fig. 1(c) is discarded after the MSB '1' is determined to be the final decoded bit. S3 is the new initial state to start a new trace-forward operation, that is, "011" is not necessary to determine the final decoded bit and the next initial state and is not used in the next decoding process. Therefore, we do not have to store them in memory. Figure 1(d) shows the decoding process by the register exchange method with 4×1 bit memory. Only MSBs are stored in memory here, and other values denoted by gray color are discarded. After the survivor path depth, all the situations are the same as those of the register exchange method except that the tentative decoded bits that are not necessary to determine the final decoded bit and the next initial state are not available. The amount of memory is greatly reduced and the power consumption for the access to memory is almost the same as that of the trace-back method. The trace-back operation is not necessary, either, and the latency for decoding is the same as that of the register exchange method. The MRE method has the advantages of both the trace-back method and the register exchange method.

The tentative decoded bits except the MSBs in the register exchange method are used only at the last decoding process when the last input symbol is received. Therefore, we have the same information for the decision process in both the register exchange and the MRE method, and the BER performance is not affected. Figure 2 shows the BER curves obtained by simulation using the trace-back method and the MRE method when K=3 and the code rate is 1/2. The MRE method does not show any degradation of BER performance compared with that of the trace-back method.

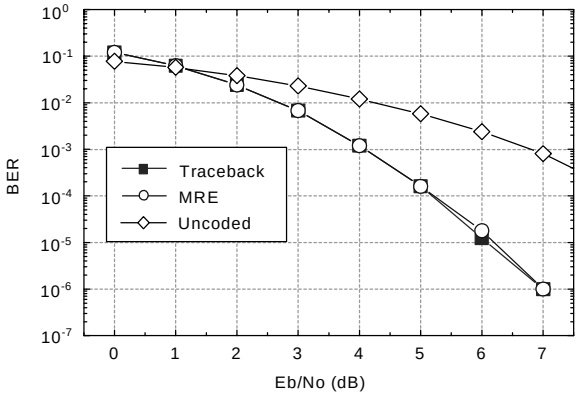

Fig. 2. BER characteristics of Viterbi decoders (K=3, code rate=1/2) using the trace-back and the MRE method.

Table 1. Comparison of required memory size and latency of three decoding methods.

Item \ Method	Trace back	Register exchange	Proposed MRE
Input buffer	L × q × n = 405	L × q × n = 405	L × q × n = 405
PM memory	S × p = 2,560	S × p = 2,560	S × p = 2,560
Trace-back memory	L × S × d = 11,520		
Decision memory		L × S × d = 11,520	S × k × d = 256
Total memory (bit)	14,485 (100%)	14,485 (100%)	3,221 (22.2%)
Latency (# of clocks)	L+L = 10K = 90	L = 5K = 45	L = 5K = 45

(Code rate 1/3, K=9, quantization bits: 3)
L: survivor path length (= 5×K),
K: Constraint length,
S: # of state (= 2^{K-1}),
d: decision bit,
k, n: code rate = k/n (k=1, n=3)
q: # of quantization bits (=3)
p: # of path metric bits (= 10)

Table 1 shows the comparison of the required memory size and the required latency, and Fig. 3 shows the block diagram of a decoder with the MRE method and necessary memory units. The numbers are calculated for the IS-95 reverse-link specification and the fully parallel ACS operations are assumed. The trace-back decoder needs as many as 11,520 bits of trace-back memory, which is 45 (survivor path length) × 256 (state) × 1 (decision bit) for the trace-back operation. However, the MRE method needs only 256 bits of the decision memory that

store the decoded bits for the corresponding states of the first trace-forward stage. The total memory size for the MRE method is only 3,221 bits including the input buffer memory and the temporary PM memory while the other methods need 14,485 bits. The input buffer memory contains the recent 45 input symbols. The memory size is reduced to about 22% of that of the conventional methods.

The discarded tentative decoded bits are used in the last decoding process in the register exchange method as we mentioned above. Since they are not stored in the MRE method, (5K–1) times the decoding processes are necessary. If the input symbols are received continuously (e.g., streaming data) or the data packet size is much larger than the survivor path length, the throughput for the MRE method is almost the same as that for the register exchange method. However, it is worse than those of other methods if the data packet size is around the survivor path length. This is the weak point of the MRE method, although the MRE method still requires much smaller memory size than the others. Table 2 shows the decoding efficiency which is defined as the ratio of the input data size and the required number of cycles to complete the decoding. The decoding efficiency is proportional to the throughput of the decoder. The modified register exchange method requires m times the decoding process for m input symbols while the trace-back and the register exchange methods require (m–5K) times the decoding process, since one decoding process is enough for the last 5K input symbols. Since the MRE method requires a 5K cycle whatever the input data packet size, the decoding efficiency is the same for all m values while it decreases as m increases for the other methods. The MRE method has a comparable decoding efficiency with the trace-back method when m is about twice the survivor path length (L) and with the register exchange method when m is about 5L. Therefore, the MRE method has its merits when applied to Viterbi decoders in practical cases. For example, the decoding efficiencies of the MRE method and the trace-back method are 0.022 and 0.015 when m=192 (the packet size of IS-95 specification), respectively.

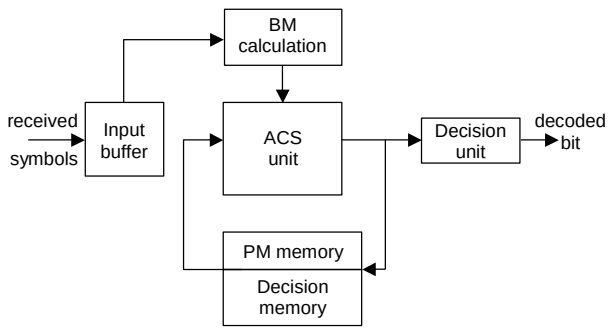

Fig. 3. Block diagram of the MRE register exchange method.

Table 2. Comparison of decoding efficiency according to the decoding data size.

Item \ Decoding method		Trace-back decoding	Register exchange	Proposed MRE
Decoding efficiency		$\frac{m}{5K(m-5K+1)}$	$\frac{m}{5K(m-5K+1)}$	$\frac{m}{5K \cdot m} = \frac{1}{5K}$
m	45	0.5	1	0.022
	100	0.020	0.040	0.022
	192	0.015	0.029	0.022
	1,000	0.012	0.023	0.022

* m: the number of input symbol
K: constraint length (=9)

III. Application to Block Decoding

The MRE method can be applied to find a merging state of block decoding algorithms. The block decoding scheme is often employed when high throughput is required [7]. Although many modified block decoding schemes have been reported [8]-[11], the basic idea is to decode input symbols by a block size in one decoding process, or moreover, a decoded bit is obtained in one trace-forward step while the classic decoding schemes decode one input symbol in one decoding process for a continuous input stream. Since a decoding process takes as many cycles as the trace-forward length (L), the block decoding scheme greatly enhances the throughput.

The trace-forward length is larger than the survivor path length by a decoding block size (D) in block decoding schemes as shown in Fig. 4. In order to obtain decoded bits, we need a merging state which is the starting state of the trace-back for the decoding block. A block decoder using the MRE scheme has MRE memory to store the decoded bits of the merging state. The size of the MRE memory is 2^{K-1} bits. After the trace-forward of decoding block length D, the decoded bits corresponding to the merging states are stored in the MRE memory, and they are shuffled according to the MRE scheme. After the trace-forward length (D+M), the survivor path is determined, and the merging state is known immediately using the MRE method. The decoded bits are obtained in reverse order. Most of the conventional block decoding methods require the trace-back operation over the length of M to find a merging state, which results in additional latency. Although the "look-ahead trace-back" method does not require the trace-back operation to find a merging state [12], the required size of memory is huge, and it is not attractive.

Table 3 shows the memory size, latency, and number of

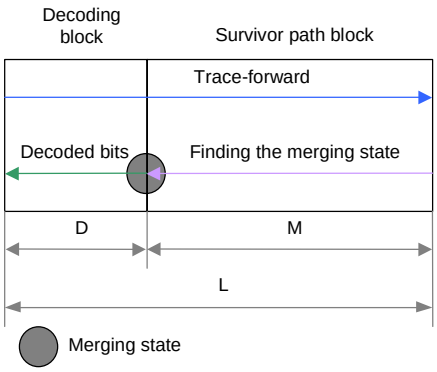

Fig. 4. The operation of a basic block decoding scheme.

Table 3. The comparison of hardware complexity and latency of block decoders (K=9, L=46).

	k-pointer even [8] (k=3)	Hybrid even [11] (k1=k2=2)	Look-ahead trace-back [12]	Proposed MRE (5-pointer)
Total memory (bits)	36,398 (100%)	25,134 (69%)	112,640 (334%)	24,855 (68%)
Latency	3L	2L	L	3L/2
# of ACSU	256	256	256	256
# of TBU	2	1*	N/A	0
# of Decoding Unit	1	1*	N/A	1

* TBU and decoding unit must operate twice as fast as other units

major operation units of several Viterbi decoders. The proposed structure is based on the k-pointer method [8], and the MRE method is applied. The memory size is reduced by 32% and the latency is decreased by 50%. Although the hybrid-even method has a similar memory size, it also has a larger latency, and the TBU and the decoding unit must operate twice as fast as the other operation units [11]. The look-ahead trace-back method does not require the trace-back operation or the smallest latency [12]. However, the memory size is about 5 times that of the proposed decoder. If the 12-pointer scheme is applied to the proposed structure, the latency is reduced to 13L/11 while the total memory size is almost the same as that of the 3-pointer method. The MRE method can also be applied to the hybrid-even method, which results in better memory management and latency. It cannot be applied to the look-ahead trace-back method. Many other techniques to improve the performance, such as radix-n, can be used with the MRE method.

IV. Design and Implementation

A Viterbi decoder complying with the IS-95 reverse-link specification is designed to verify the operation of the modified register exchange scheme. It has a code rate of 1/3, the constraint length K=9 (the number of states is 256), and the generation polynomial (577, 663, 711) is used. Figure 5 shows the block diagrams of the IS-95 reverse-link convolutional encoder.

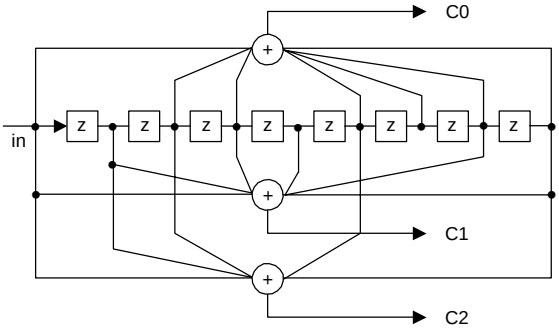

Fig. 5. Block diagram of the IS-95 reverse-link convolutional encoder.

The operation of the designed Viterbi decoder is verified by the simulation using Synopsys tools, and the decoder is synthesized using a 0.35 μm CMOS technology. Table 4 summarizes the synthesis results compared with a reported one [13].

Table 4. Summary of the synthesizes results.

	Ref. [13]	Ours
Technology	0.25 μm CMOS cell library	0.35 μm CMOS cell library
Area (mm^2)	2.4×2.4	4.6×4.6
Operation frequency	640 MHz	70 MHz
Throughput	20 Mbps	70 Mbps

V. Conclusions

I have proposed a new implementation method of the Viterbi decoder, the modified register exchange method. The memory for tentative decoded bits that are not necessary to determine the starting state and the final decode bit is eliminated. It greatly reduces the memory size. The final decoded bit is known as soon as the survivor path is determined by rearranging the position of the decoded bits at the first trace-forward stage. It results in the elimination of the trace-back operation. Therefore, the MRE method has the advantages of reduced memory size

and no trace-back operation. The more attractive advantage of the modified register exchange method is its applicability to other decoding architecture, such as the block decoding scheme, pipelined architecture, etc. A Viterbi decoder complying with the IS-95 reverse-link specification is designed to verify the proposed implementation method. It is synthesized using a 0.35 μm CMOS technology and cell library.

References

[1] G. Forney, "Convolutional Codes II. Maximum-Likelihood Decoding," *Information and Control*, vol. 25, no. 3, July 1974, pp. 222-266.

[2] K. Hu, W. Lin, and M. Caldwell, "A Viterbi Decoder Memory Management System Using Forward Traceback and All-Path Traceback," *Proc. of the IEEE 1999 Int'l Conf. Consumer Electronics*, 1999, pp. 68-69.

[3] Heinrich Meyr, Marc Moenenclaey, and Stefan A. Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing*, Wiley-Interscience, 1997.

[4] Jens Sparso, Henrik N. Jorgensen, Erik Paaske, Steen Pendersen, and Thomas Rubner-Petersen, "An Area-Efficient Topology for VLSI Implementation of Viterbi Decoders and Other Shuffle-Exchange Type Structures," *IEEE J. Solid-State Circuit*, vol. 26, no. 2, Feb. 1991, pp. 90-97.

[5] Shu Lin and Daniel J. Costello Jr, *Error Control Coding: Fundamentals and Applications*, Prentice Hall, 1983.

[6] J.L. Heller and I.W. Jacobs, "Viterbi Decoding for Satellite and Space Communication," *IEEE Trans. Commun.* COM19, no. 5, Oct. 1971, pp. 835-848.

[7] Peter J. Black and Teresa H.-Y. Meng, "A 1-Gb/s, Four-State, Sliding Block Viterbi Decoder," *IEEE J. Solid-State Circuits*, vol. 32, no. 6, June 1997, pp. 797-805.

[8] R. Cypher and C.B. Shung, "Generalized Traceback Techniques for Survivor Memory Management in the Viterbi Algorithm," *GLOBECOM*, Dec.1990, pp. 1318-1322.

[9] O. Collins and F. Pollara, "Memory Management in Traceback Viterbi Decoders," *TDA Prog. Rep.*, Nov. 1989, Jet Prop. Lab., Pasadena, CA, pp. 42-99.

[10] G. Feygin and P.G. Gulak, "Survivor Sequence Memory Management in Viterbi Decoders," *CSRI Tech. Rep.* 262, Univ. of Toronto, Jan. 1991.

[11] G. Feygin and P.G. Gulak, "Architectural Tradeoffs for Survivor Sequence Memory Management in Viterbi Decoders," *IEEE Trans. Commun.*, vol. 41, no. 3, Mar. 1993, pp. 425-429.

[12] Jung-Gi Baek, Sang-Hun Yoon, and Jong-Wha Chong, "Memory Efficient Pipelined Viterbi Decoder with Look-Ahead Traceback," *The 8th IEEE Int'l Conf. Circuit and Systems*, vol. 2, 2001, pp.769-772.

[13] Xun Liu and Marios C. Papaefthymiou, "Design of High-Throughput Low-Power IS95 Viterbi Decoder," *2002 Design Automation Conf.*, vol. 39, June 2002, pp. 263-268.

Chanho Lee received the BS and the MS degrees in electronic engineering from Seoul National University, Seoul, Korea, in 1987 and in 1989, and the PhD degree from the University of California, Los Angeles, in 1994. In 1994, he joined the Semiconductor R&D Center of Samsung Electronics, Kiheung, Korea. Since 1995, he has been a faculty member of the School of Electronic Engineering, Soongsil University, Seoul, Korea, and he is currently an Associate Professor. His research interests are the design of channel codec and security processor, low power design, and on-chip network. He is a senior member of IEEE.