Low latency and power efficient VD using Register Exchanged state-mapping algorithm

Sang-Ho Seo, and Sin-Chong Park
Information and Communications University
ttiseo@icu.ac.kr , scpark@icu.ac.kr

Abstract

In this paper, a new implementation of the Viterbi decoder is proposed. RE state-mapping algorithm combines TB algorithm with the RE algorithm. By updating the starting point of the state for each memory bank, the trace back operation can be eliminated. This result reduces the latency of the TB algorithm and the resource usage of RE algorithm. When the memory unit is 3, the resource usage is 6306 bits and the latency is 72 clock. The resource usage is 74% smaller than the RE algorithm. and the latency is 34% smaller than the k-pointer even TB algorithm.The power consumption of each algorithms are also analyzed and compared. Because the resource usage is directly related to the power, the power consumption of this scheme is also reduced[8]. Actually, the estimated power consumption is about 12% smaller than the RE algorithm.

1. Introduction

The viterbi algorithm, which uses maximum likelihood decoding [1] (MLD), is widely used one for modern digital communication to achieve low-error-rate decoding. There are two main algorithms of the implementation-based viterbi decoder. The first one is the Trace Back algorithm(TB) [2], and the second one is the Register Exchange algorithm(RE) [3]. Because of its simplicity, many applications are using the TB algorithm. However, the drawback of this algorithm is a rather long latency (2~3 times of Depth T). Register exchange algorithm uses trace forward scheme to achieve optimum latency(T). Because of the large resource usages and the large power consumption, TB algorithm is preferred in the design of large constraint length.

Recently, the combination of TB and RE was introduced in Han's MRE algorithm[4]. By eliminating the trace back operation from the TB algorithm, its resource usage and latency is enhanced.

In Feygin's paper [5], the latency and the resource usage of various decoding methods are generalized as an equation. In this paper, the Feygin's equations are revised and applied to the k-pointer even TB algorithm, RE algorithm, MRE algorithm and the proposed state-mapping algorithm. The generalized performance equations are to be compared with the proposed algorithm. All of the above four algorithms are analyzed and compared in that resource usages, latency and power consumption.

This paper consists of following contents. In Section 2, the basic operation of the TB and RE method is mentioned with an example (constraint length K=3). In Section 3, the resource usage, latency and power consumption of each algorithm is analyzed with the performance equation. In Section 4, The resource usage and the latency of the proposed algorithm is compared with upper three algorithms. In Section 5, all of the pre-described viterbi algorithms are compared as a power consumption point of view. Finally we conclude the paper in Section 6

Table 1. The abbreviations used in this paper

Abbreviation	Description
T	Decoding Depth (4~5 times of constraint length)
U	Number of columns in one decoding unit
k	Number of writing operation to decode one decoding unit
B	Number of accessed decoding units to complete the decoding of one decoding unit.
B`	Number of total decoding units for the architecture.
L	Total latency
Mpu	Memory usage for the update of Path Metric.
Msu	Memory usage for the update of state
Mp	Memory usage for the Path Metric
Ms	Memory usage for the state

Proceedings of the Fifth International Workshop on System-on-Chip for Real-Time Applications (IWSOC'05)
0-7695-2403-6/05 $20.00 © 2005 IEEE

Md	Memory usage for ACS decision information
Tr	Total Resource usage
Tr	Total Resource usage
w	The power-dissipation cost function for writing one bit into the memory block.
r	The power-dissipation cost function for reading one bit from the memory block.

2. The Basic operation of ML decoding viterbi algorithm (with K = 3)

The viterbi algorithm was developed in 1967 to decode the convolutional encoder. Figure 1 shows the convergence point of the viterbi decoder(K=3), which means that states always converge at one state after tracing back for depth stages(T). This characteristic enables the VA to work in the pipelined architecture

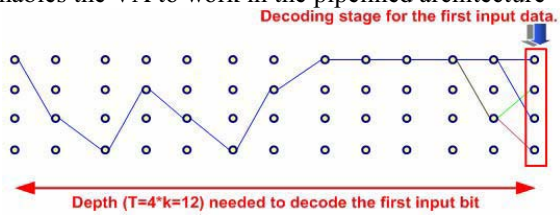

Figure 1. Convergence characteristic of VA

2.1. Trace back algorithm

The trace back sequence is depicted in Figure 2, where k is 3, and source sequence is 11011. After tracing back from the last to the first stage, we can get a reverse ordered decoded sequence, which means the TB algorithm requires an LIFO operation.

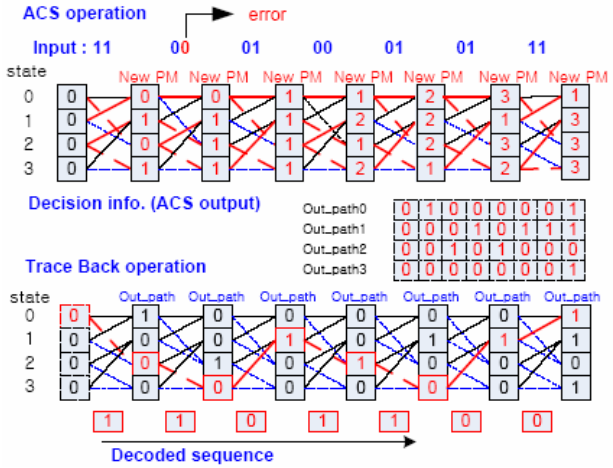

Figure 2. The operation of the TB algorithm.

2.2. Register exchange algorithm

Figure 3 shows the Register exchange sequence. The memory for the decision information is updated and exchanged at every stage. The exchange operation involves high power consumption, but the contents of the memory itself is the decoded output sequence, which means the RE algorithm does not require a LIFO operation.

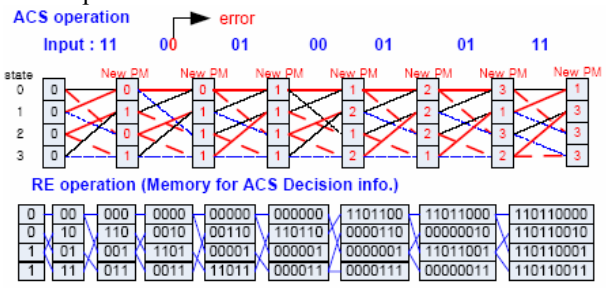

Figure 3. The operation of the RE algorithm.

3. Resource usage and latency of each algorithm

3.1. k-pointer even algorithms

The k-pointer even algorithm is found in Bustamante's paper [6]. This algorithm is a generalized decoding scheme with latency of 2T~3T.Although this latency is a little bit longer than the others, it is widely used because of its simplicity.

Decision information is processed through 3 kinds of modes. WR is the writing decision information to the memory bank. and TB is the trace back operation, and the last one is the decoding operation(DC) which includes trace back and decoding.

For the k-pointer even algorithm, the number of columns per one bank(U) is T/(k-1) [5]. The number of accessed banks to decode one bank(B) is 2k, and also the total number of bank (B`) is 2k. Therefore, the latency(L) is the number of columns per bank multiplied by the number of banks(L=UB = 2kT/(k-1)). From a resource point of view, PM update memories and decision information memories are used (Mpu=$2^{K-1}$*10*2, Md=$2^{K-1}$*UB`). We assumed that the bit width of PM is 10 bits. The memories for LIFO(2U-2 bits) are also used. So, the total resource usage is $20*2^{K-1}+(2kT*2^{K-1}+2T)/(k-1)-2$.

Figure 4 shows the example of the 3-pointer even algorithm. In this architecture, the value of K is assumed to 7 and that of k is equal to 3. So the parameter U is 18, B = 6 and B` = 6. therefore, the latency is 108 clock and the total resource usage is 8226 bits.

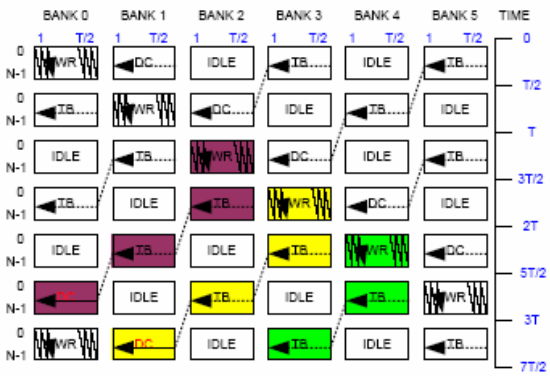

Figure 4 The operation of the 3-pointer even algorithm

3.2. Register exchange (RE) algorithm

The operation of this algorithm is described in Fig. 3. When the value of K is 7, each size of ACS decision information memory is 36(T). This algorithm has fixed latency of T clock. But it can be analyzed and represented as the same methods of the k-pointer TB algorithm. The parameter U = T/(k-1), B=k-1, B`=k-1, Mpu=$2^{K-1}$*10*2, Mp=$2^{K-1}$*10*UB`, and Md=2^{K-1}UB`. Therefore, the latency L=UxB=T and Tr= 2^{K-1}(20+11T)

When the value of K is equal to 7 and that of k is equal to 37, the parameter U is 1, B = 36 and B` = 36. Therefore the latency is 36 clock and the resource usage is 24356 bits.

3.3. Modified RE algorithm

The modified RE [4] is a combined version of TB and RE algorithm. While the RE algorithm exchange the decision information itself, the MRE algorithm exchanges the pointer of state. Because there are only 64 states for the viterbi decoder with K=7, the length of the pointer is only 6 bit which is exchanged at every clock. This operation eliminates the trace back sequence. That is , the starting point of decoding operation is resolved at the time the writing operation is finished (T clocks later).

U=T/(k-1), B=k+1, B`=k. So the latency L is UB, which is equal to T(k+1)/(k-1). From a resource point of view, Mpu=$2^{K-1}$*10*2, Msu=$2^{K-1}$*6*2(k-1), Md=$2^{K-1}$*UB`, and the memory block for LIFO is 2U-2. Therefore the total resource is equal to 2^{K-1}(12k+8+kT/(k-1))+ 2T/(k-1)-2. Figure 5 shows the example of MRE algorithm with k=3 and K=7. In this case, the latency is 72 clock and the resource usage is 6306 bits. Table 2. shows the summary of the generalized equation for the pre-described viterbi algorithms.

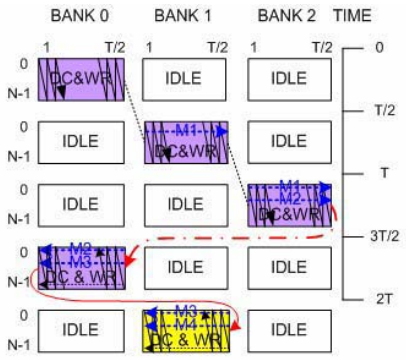

Figure 5. The operation of the MRE algorithm.

Table 2.The summary of resource usage and latency for each VA.

	k-pointer even	RE	MRE
U	T/(k-1)	T/(k-1)	T/(k-1)
B	2k	k-1	k+1
B`	2k	k-1	k
Mpu	2^{K-1}*10*2	2^{K-1}*10*2	2^{K-1}*10*2
Msu			2^{K-1}*6*2(k-1)
Mp		2^{K-1}*10*UB`	
Md	2^{K-1}*UB`	2^{K-1}UB`	2^{K-1}*UB`
LIFO	2U-2		2U-2
Tr	20*2^{K-1}+2T(k*2^{K-1}+1)/(k-1)-2	2^{K-1}(20+11T)	2^{K-1}(12k+8+kT/(k-1))+2T/(k-1)-2
L	2kT/(k-1)	T	T(k+1)/(k-1)

4. Register exchanged state-mapping algorithm

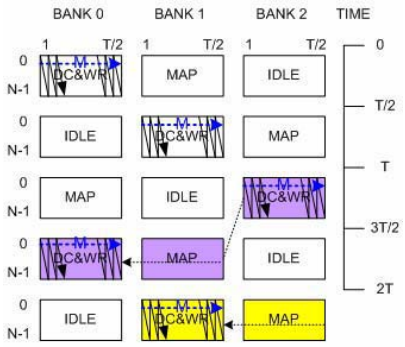

Figure 6 The operation of the RE state-mapping algorithm

The block diagram of a viterbi decoder using RE state-mapping algorithm is shown in Figure 6. Although this algorithm is based on the MRE algorithm, instead of using (k-1) MRE schemes, only one MRE scheme is used at each bank. The result of each bank's MRE scheme is stored at the other

memory space, and then referenced at the mapping stage(MAP mode). Table 3 shows the resource usage of this algorithm.

Table 3. The resource usage and latency of the RE state-mapping algorithm.

U	T/(k-1)	Msu	$2^{K-1}*6*2$
B	k+1	Ms	$2^{K-1}*6(k-1)$
B`	k	Md	$2^{K-1}*UB`$
Mpu	$2^{K-1}*10*2$	LIFO	2U-2
Resource usages(Tr)		2K-1(6k+26+kT/(k-1))+ 2T/(k-1)-2	
Latency (L)		UB = T(k+1)/(k-1)	

By using additional Ms resource, Msu of the RE state-mapping scheme is 1/(k-1) of the MRE algorithm.

When the value of K is equal to 7 and that of k is 3, the parameter U is 18, B = 4 and B` = 3. Therefore, the latency is 72 clocks and the total resource usage is 6306 bits. Figure 7 and Figure 8 shows the graph of resource usages and latency with the parameter k. As shown, the smallest resource usages are achieved by the k-pointer even algorithm, the shortest latency is achieved by the RE algorithm. And also the latency of MRE converges very rapidly to RE latency. Although the proposed scheme's latency is the same as the MRE algorithm, but the resource usages are almost half of the MRE algorithm.

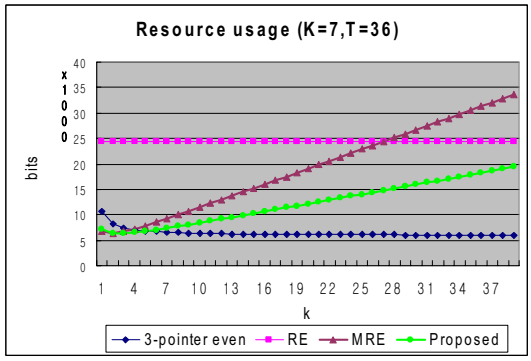

Figure 7 The resource usage of each VA

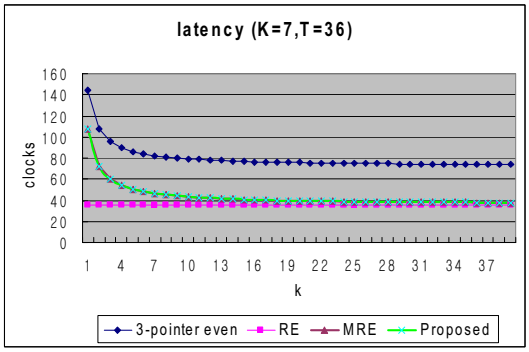

Figure 8 the latency of each VA

5. Power reduction estimation

In table 4, w, r represent the power-dissipation cost function for writing one bit into the memory block, reading one bit from the memory block, respectively.

Because the reading can be performed by adopting a reduced swing on the bit lines, a memory cell uses less power for reading than for writing. It is assumed that w:r~2:1 [4].

The table below shows the number of bits written and read. The total value is the summation of all the other columns in each algorithm.

Table 4. The power consumption of each viterbi algorithm.

	k-pointer even	RE	MRE	proposed
Writing PM update bits into memory	$[2^{K-1}*10*2UB]w$	$[2^{K-1}*10*2UB]w$	$[2^{K-1}*10*2UB]w$	$[2^{K-1}*10*2UB]w$
Reading PM update bits from memory	$[2^{K-1}*10*2UB]r$	$[2^{K-1}*10*2UB]r$	$[2^{K-1}*10*2UB]r$	$[2^{K-1}*10*2UB]r$
Writing state update bits into memory			$[w2^{K-1}*6*2(k-1)UB]w$	$[2^{K-1}*6*2UB]w$
Reading state update bits from memory			$[2^{K-1}*6*2(k-1)UB]r$	$[2^{K-1}*6*2UB]r$
Writing state bits into memory				$[2^{K-1}*6*(k-1)B]w$
Reading state bits from memory				$[2^{K-1}*6*(k-1)B]r$
Writing PM bits into memory		$[2^{K-1}*10*UB`*UB]w$		
Reading PM bits from memory		$[2^{K-1}*10*UB`*UB]r$		
Writing decision bits into memory	$[2^{K-1}*UB`*UB]w$	$[2^{K-1}*UB`*UB]w$	$[2^{K-1}*UB`*UB]w$	$[2^{K-1}*UB`*UB]w$
Reading decision bits from memory	[UB]r	[UB]r	[UB]r	[UB]r
Total	[285768k/(k-1)]r	[165924+55296(k-1)]r	[82944(k+1)+142884(k+1)/(k-1)]r	[225828(k+1)/(k-1)+41472(k+1)]r

Figure 9 is the graph of estimated power consumption. This result shows that the power consumption of proposed algorithm is about 60% of MRE algorithm.

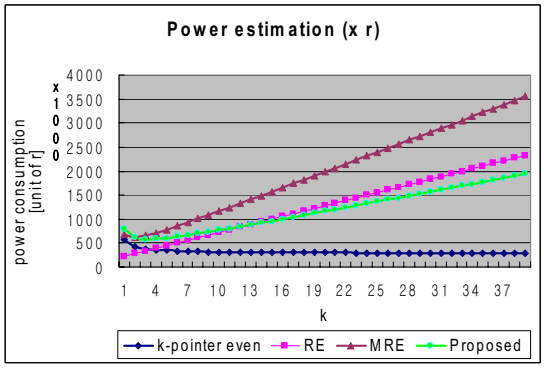

Figure 9 The power estimation of each VA

6. Conclusion

This paper has proposed an implementation scheme of the viterbi decoder. RE state-mapping algorithm uses the pointer of state which indicates the starting point of each trace unit. By mapping this pointer, the trace back stages can be eliminated. The reduction of updating information and stages means that of resource usages and latency, respectively. When the memory unit is 3, the resource usage is 6306 bits and the latency is 72 clock. The resource usage is 24% smaller than the k-pointer even TB algorithm and 74% smaller than the RE algorithm. Because the resource usage is directly related to the power, the power consumption of this scheme is also reduced [8]. Actually, the estimation result of power consumption is almost half of the MRE algorithm.

7. Reference

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

[2] S.B. Wicker, *Error Control Systems for Digital Communication and Storage*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[3] Steinert, M., Marsili, S., " Power consumption optimization for low latency Viterbi Decoder", *Circuits and Systems,* 2004. ISCAS '04. Proceedings of the 2004 International Symposium on , vol. 2 , 23-26 May 2004, pp. 377-380

[4] Jae-Sun Han, Tae-Jin Kim, Chanho Lee, " HIGH PERFORMANCE VITERBI DECODER USING MODIFIED REGISTER EXCHANGE METHODS ", *Circuits and Systems*. , ISCAS '04. , vol. 3 , 23-26 May 2004

[5] Feygin, G., Gulak, P. " Architectural tradeoffs for survivor sequence memory management in Viterbi decoders " , *Communications, IEEE Transactions* on , vol. 41 , Mar. 1993, pp.425– 429

[6] H.A. Bustamante, et al., "Stanford telecom VLSI design of a convolutional decoder." In IEEE Conf. *Military Commun*., vol. 1, Boston, MA, Oct. 1989, pp. 171-178.

[7] El-Dib, D.A., Elmasry, M.I., " Modified register-exchange Viterbi decoder for low-power wireless communications", *Circuits and Systems I: Regular Papers,* IEEE Transactions on [see also Circuits and Systems I: Fundamental *Theory and Applications*, IEEE Transactions on] , vol. 51 , Feb. 2004, pp. 371 – 378

[8] D. Liu and C. Svensson, "Power consumption estmation in CMOS VLSI chips, " IEEE J. *Solid-State Circuits*, vol.29, June 1994, pp. 663-670