Modified Register Exchange Method of Viterbi Decoder for 3GPP Mobile System

Chaiwat Keawsai[1], Keattisak Sripimanwat [2], and Attasit Lasakul[1]

[1]Department of Information Engineering, Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang, THAILAND klattasi@kmitl.ac.th
[2]Telecommunications and Computer Network Laboratory
National Electronics and Computer Technology Center (NECTEC), THAILAND keattisak.sripimanwat@nectec.or.th

ABSTRACT

This paper presents a design and implementation of small Viterbi decoder architecture. Its specifications are coding rates 1/2 and 1/3, with generator polynomial $(561, 753)_8$ and $(557, 663, 711)_8$ respectively. Both cases are designed with the constraint length of 9 (256 states). In addition, they comply with 3GPP (3rd Generation Partnership Project) standard. In this work, the serial architecture with four add-compare-select (ACS) combined with the modified register exchange method are proposed. The prototype has been successfully implemented on Xilinx Vertex II FPGA device. As a result, it obtains small resource of FPGA (slices and block RAM). Furthermore, its data rate exceeds 2 Mbps which is suitable for 3G (W-CDMA) mobile system. It also can be applied for other related communication systems.

Keywords: Viterbi decoder, 3GPP, FPGA

1. INTRODUCTION

In order to improve efficiency of signal transmission and to prevent the effect of channel's impairment in digital communications, forward error correction (FEC) is adopted as a required unit in the baseband part. Convolution code with viterbi decoding is one of FEC techniques which is widely used in digital communication systems such as in satellite communication systems, wireless LAN and mobile communication systems. In 3GPP mobile system, this code is used to improve the efficiency of voice transmission as defined in TS 25.212 standard [1][2].

For VLSI (Very Large Scale Integration Circuit), architecture of viterbi decoder can be categorized into two cases of parallel and serial architecture. Basically, its parallel architecture operates with higher speed than that of the serial style but it, of course, obtains a larger implemented area [3]. Actually, the area and complexity of parallel architecture exponentially increases followed the constraint length number (K) of encoder and the number of states. Refer to 3GPP standard, main specifications of viterbi decoder are set as coding rate 1/2 and 1/3 with constraint length of 9. The high constraint length (K is 9) of parallel structure consumes large area and with expensive cost. Due to the limitation of integrated circuits on their speed, implementation area and power consumption, thus the serial architecture is selected as the main goal for this work in order to have a small implementation area and less cost.

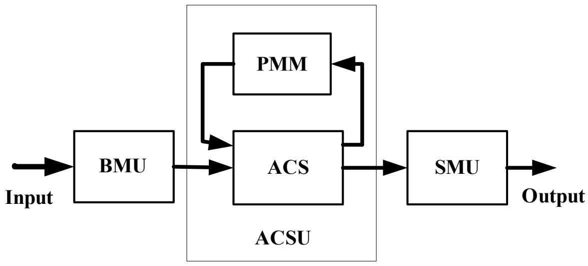

***Fig. 1**: Diagram of Viterbi Decoder*

In this paper, the 4 ACS scheme is an area-efficient VLSI architecture which achieves the maximum throughput described in [4], and register exchange (RE) method [5] which is modified to connect with this serial architecture style, are considered.

This paper is organized as follows. The principle architecture of viterbi decoder is described in section 2. Next, the proposed method is presented in section 3. Section 4, the implementation and results are presented. Finally, the conclusion is shown in section 5.

2. VITERBI DECODER ARCHITECTURE

The viterbi decoding algorithm performs maximum likelihood sequence detection on data which have been convolutionally coded. The decoding uses the trellis diagram to determine the output data sequence path which is the most likely to the input sequences from encoder. Viterbi decoder has three major processing units, branch metric unit (BMU), add-compare-select unit (ACSU), and survival memory unit (SMU) as shown in Fig. 1.

In the decoding process, the BMU will first calculate with branch metrics (BM) of Hamming distance (for hard decision) or Euclidean distance (for soft decision) from received input codeword and also from the branch word [6]. The branch word value depends on the constraint length, the generator polynomial, and the coding rate of encoder. Then, the BMU produces 2^n branch metric numbers which coding rate of $1/n$.

Next, the ACSU is used to find the path metric (PM) as well as decision bits (DB) from two branch metrics associated with two path metrics. The updated path metric will be the smallest previous metrics which has been accumulated by branch metric from BMU.

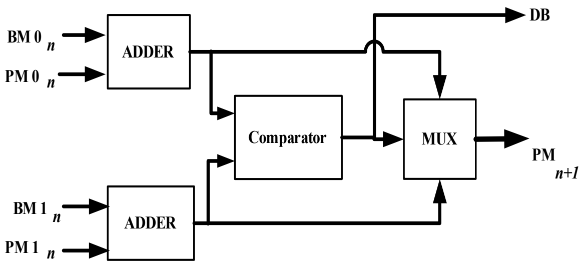

Fig. 2*: ACS Diagram*

Table 1*: The proposed Viterbi decoder specifications*

Coding rate	1/2 , 1/3
Constraint length	9
Generator polynomials	561_8, 753_8 , R =1/2 557_8, 663_8, 711_8, R =1/3
Survivor path length	64
Decision level	Soft decision 8 level (3 bits) Offset binary representation
Path Metric	8 bits Modulo Normalization
ACS unit	4
Data transmission	Continuous

Subsequently, it will be stored in the path metric memory (PMM) for calculating of the next path metrics and also decision bits. This ACS model is shown in Fig. 2 and described by

$$PM_{n+1}^{0} = min(PM_n^0 + BM_n^0, PM_n^1 + BM_n^1) \quad (1)$$

where $PM_{n+1}^{i}(i=0,1)$ and $BM_{n+1}^{i}(i=0,1)$ are path metrics and branch metrics at time n respectively [7].

In ACSU, parallel viterbi decoder architecture uses 2^{k-1} ACS (K is constraint length) while serial architecture uses 2^i ACS (i is 0, 1, 2, …). Obviously, that parallel structure takes large implemented area than that of the serial one.

Finally, the SMU stores decision bits from ACSU, then, to find survivor path, and to decode the received data. There are two methods to find survivor path and decoded data. These are register exchange (RE) and track back (TB). The first method, it is the conceptual simplest and fast decoding, but it requires large chip area. This register exchange consists of array of multiplexers and registers which are connected to resemble the trellis diagram of convolution encoder. The depth of array of multiplexers and registers are required five times larger than the value of constraint length for without resulting in significant performance degradation [4].

The second method is the track back that the decision bits from ACSU will be stored in memory and used to trace backward as done similarly as in the method of RE. This scheme works with a higher complexity and with slower in speed than that of the RE method. But it requires smaller area for implementation as the advantage.

3. THE PROPOSED CONCEPT

In this paper, the viterbi decoder is designed and implemented aims for 3GPP standard. Its specifications are summarized in Table 1. The diagram of proposed architecture is shown in Fig. 3. It is composed of branch metric unit (BMU), add compare select unit (ACSU), survival memory unit (SMU), and a control unit.

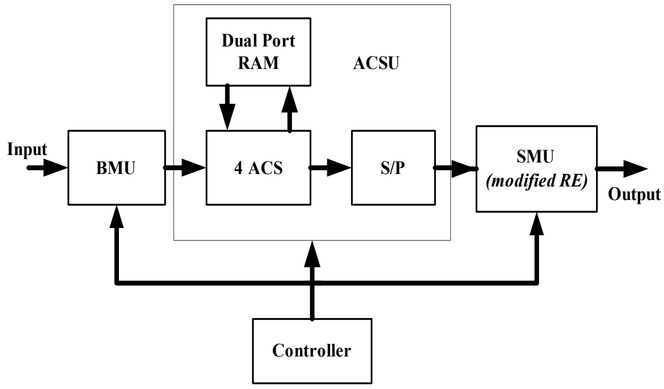

Fig. 3*: The proposed Viterbi decoder*

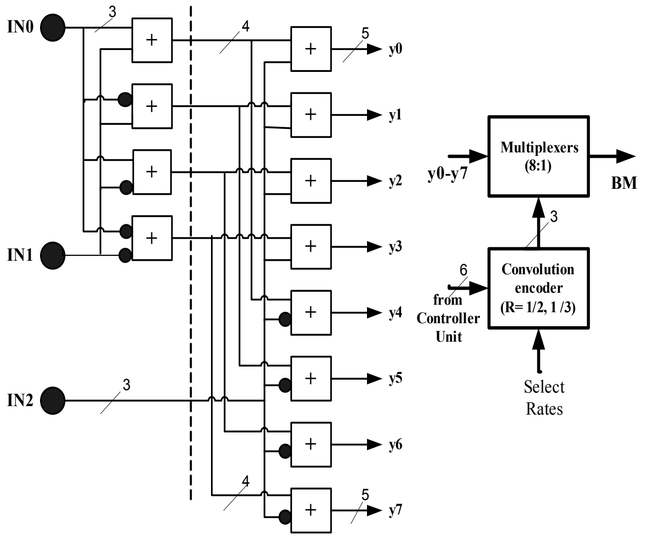

Fig. 4*: The 3 bit soft decision branch metric unit*

3.1 Branch Metric Unit (BMU)

This BMU works for calculating branch metric from received input sequences. Its structure of the 8-level (3 bits) soft decision and adaptive coding rate of both 1/2 and 1/3 are designed in this work. This is to suit with the 4 ACS architecture as the purpose. This BMU consists of 12 adders (8 bits), 8 multiplexers (8:1) and 8 dual rate of convolution encoder (1/2 and 1/3). The 8-bits adders calculate Euclidean distances from received input codeword following by

$$BM(b0,b1,b2) = -IN0s(b0) - IN1s(b1) - IN2s(b2) \quad (2)$$

where $IN0, IN1$ and $IN2$ are soft decision inputs, and $BM(b0,b1,b2)$ is branch metric. *b0, b1*, and *b2* are binary values varied from 0 to 7, and *S(bi)* represents the symmetric symbols of *a* or *–a* which corresponding to the logic value either "one" or "zero" [8]. For example *b0, b1* ,and *b2* represents to (0, 1, 0), then *S(bi)* is (-, +, -). Next, the branch metrics from adders are selected by dual rate of convolution encoder and multiplexers. In addition, this BMU is divided into two pipeline stages in order to increase its speed. The diagram of this 3 bit soft decision BMU is shown in Fig. 4.

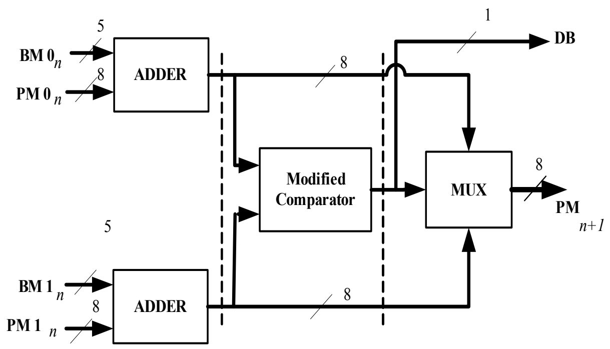

***Fig. 5**: The modulo normalization ACS with modified comparison rule [9]*

3.2 Add Compare Select Unit (ACSU)

The ACSU work in serial architecture style with 4 ACS for having small implementation area. This BMU consists of 4 ACS, a 64 x 64 dual port RAM, and a serial to parallel (S/P) unit as shown in Fig. 3. The ACS takes two inputs respectively of branch metric from BMU and path metrics from path metric memory (PMM) which resides in dual port RAM. Then, the updated path metric and the decision bit are given as the output. These outputs are followed to calculate for the decision bit and for the updated path metric by using Equation 1. This ACS takes modulo normalization technique with modified comparison rule of [9] to avoid the path metric overflow. In addition, The ACS is divided into three pipeline stages for increasing speed. The diagram of to 3-pipelines ACS is shown in Fig. 5. ACSU computes by using 64 clock cycles for calculating one of information bit on 256 states of decoding. While the dual port RAM stores that updated path metrics to use for the next bit of computing process. At the last step, S/P unit converts 4 to 256 decision bits on each of one-bit decoding and then transfers them to SMU.

3.3 Survival Memory Unit (SMU)

In SMU, register exchange (RE) method is used in this work. This RE is modified to suit with 4 ACS style for reducing implemented area. This modified RE method is shown in Fig. 6. Since, the original register exchange structure is large, it is modified in order to reduce the size of multiplexers (2:1) by replacing with $6K$ x 2^{K-1} shift registers and a column of 2^{K-1} multiplexers (2:1). The shift registers store decision bits and then the 2^{K-1} multiplexers (2:1) exchange decision bits for decoding operation. It is done by using $6K$ clock cycles to calculate one of information.

This proposed SMU structure consists of 64 x 256 shift registers and 256 multiplexers (2:1). The 256 decision bits from ACSU are saved to 256 shift registers on each of one-bit decoding (64 clock cycles). Then, decision bits are exchanged by 256 multiplexes on each of clock-cycle of the decoding. The decoded bit is initiated after an equivalent time of 54 (K is 9) has elapsed without resulting in significant performance degradation. In this design, 64 of survivor paths length of decoding to suite for ACSU operation, is selected. The proposed SMU can decrease 63-colums out of 256 of the multiplexers and also reduce complexity of multiplexers interconnection. This SMU with modified register exchange is shown in Fig. 7.

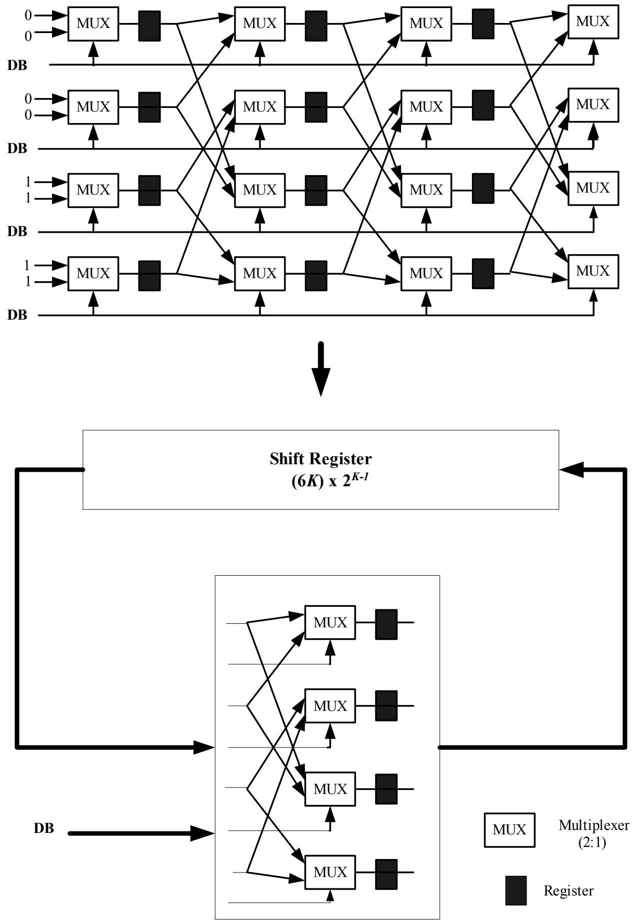

***Fig. 6:** The method of modified RE*

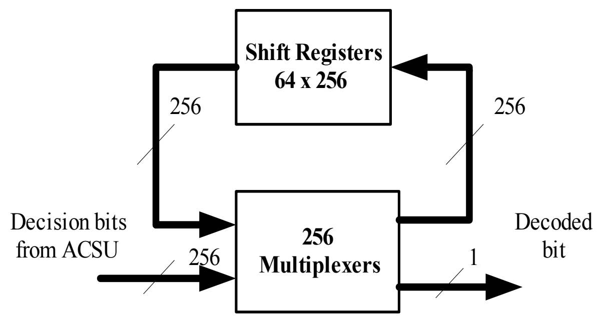

***Fig. 7:** SMU structure with modified RE*

4. IMPLEMENTATION AND RESULTS

The proposed viterbi decoder has been designed by using Verilog HDL and it is implemented on Xilinx Vertex II xc2v1000-6 FPGA platform. As a result, it uses 1,416 slices, 2 block RAMs and maximum frequency is 222 MHz. Thus, it achieves up to 3.46 Mbps. Its results are compared with a similar commercial product [10] as shown in Table 1. It is obvious that the proposed design obtains a small-implemented (slices) area and achieves maximum data rate up to 3.46 Mbps. Although, this speed is not as fast as that of [10], it is much faster than that of 3GPP's specification (2 Mbps). Moreover, there are more advantages of this proposed design on having smaller block RAM and operating with adaptive coding rates (1/2 and 1/3) as well. Finally, BER (bit error rate) of this design in an AWGN channel by Verilog HDL simulation for rates 1/2 and 1/3 are performed in Fig. 8.

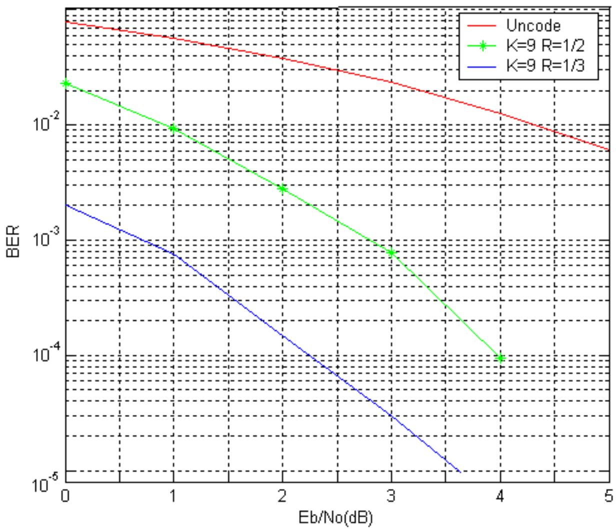

Fig. 8: *BER vs Eb/No (dB) in an AWGN channel for rates 1/2 and 1/3*

Table 2: *Performance comparison of the proposed work and Xilinx / FPGA xc2v1000-6 [10]*

	Proposed Work	Xilinx
Constraint length	9	9
Coding rate	1/2, 1/3	1/2
Survivor path length	64	64
Soft decision	3	3
Area (slices)	1416	1775
Block RAM	2	4
Max. clock frequency (MHz)	222	172
Max. data rate (Mbps)	3.46	15

5. CONCLUSIONS

In this paper, a small architecture of viterbi decoder for 3GPP mobile communication system is presented. Four ACS combined with the modified register exchange has been proposed in order to achieve small implemented area. The results show that this design got higher speed than that of 3GPP specification and having less number of slices and block RAM as the out come.

6. REFERRENCES

[1] 3rd Generation Partnership Project (3GPP) *Technical Specification Group (TSG) Multiplexing and channel coding (FDD)* TS 25.212 v2.3.0, 1999.

[2] Bickerstaff, M.A., Garrett, D.; Prokop, T., Thomas, C., Widdup, B., Gongyu Zhou, Davis, L.M., Woodward, G., Nicol, C., Ran-Hong Yan “A unified turbo/viterbi channel decoder for 3GPP mobile wireless in 0.18-/spl mu/m CMOS” *IEEE Journal of Solid-State Circuits*, Vol. 37, pp. 1555-1564, Nov 2002.

[3] Bupesh Pandita, Subir K Roy. “Design and implementation of a Viterbi decoder using FPGAs” *VLSI Design, Twelfth International Conference*, pp. 611-614, 1999.

[4] Sayed Aon Mujtaba “An area-efficient VLSI architecture of the Viterbi decoder for reverse link IS-95 (CDMA) air interface” *Signal Processing Proceedings. ICSP '98*., pp. 525-528 , 1998.

[5] Gennady Feygin, P.G. Gulak “Architectural Tradeoff for Survivor Sequence Memory in Viterbi Decoder” *IEEE Transactions on Communications*, Vol. 41, pp. 425-429, March 1998.

[6] HL Lou. “Implementing the Viterbi Algorithm” *IEEE Signal Processing Magazine*, Vol.15, pp 42-52. 1995.

[7] Hiroshi Suzuki, Yun-Nan Chang, Keshab K. Parhi “Low-power Bit-Serial Viterbi decoder for 3rd Generation W-CDMA Systems” *IEEE 1999 Custon Intergrated Circuits Conference*, pp. 589-592, 1999.

[8] Y. Chang, H. Suzuki, and KK Parhi, “A 2-Mb/s 256-state 10-mW rate-1/3 Viterbi decoder” *IEEE Journal of Solid-State Circuits*, Vol. 35, pp. 826-834, June 2000.

[9] Shung, C.B., Siegel, P.H., Ungerboeck, G.,Thapar, H.K. “ VLSI architectures for metric normalization in the Viterbi algorithm” *Supercomm Technical Sessions. SUPERCOMM/ICC '90. IEEE International Conference*, pp. 1723-1728,1990.

[10] Xilinx Inc. *LogiCore Viterbi decoder V2.0* data sheet [Online] Available: http://www.xilinx.com. 2002.