



The Design of Viterbi Decoder with Higher Efficiency

¹ Aparna Busi, ² Latha. K, ³ Rahamtula. Shaik

¹ PG Scholar, Dept of ECE, QIS Institute of Technology, Ongole, AP, India.

^{2,3} Assistant Professor at QIS Institute of Technology, Ongole, AP, India.

ABSTRACT

A Viterbi Decoder (VD) is an elegant and efficient method to decode convolutional codes, this decoding method avoids explicitly enumerating the 2^N possible combinations of N-bit parity bit sequences. This method was invented by Andrew Viterbi '57 and bears his name. Convolutional codes are commonly used to encode digital data before transmission. However, there is a large variety of modern wireless communication standards; a flexible hardware platform that can be configured to support different standards is still needed. In this paper, a reconfigurable Viterbi decoder has been designed. The proposed Viterbi decoder has an architecture that supports constraint lengths 3, 5, and 7, and code rates $1/2$ and $1/3$ and in extension work we propose an architecture to detect the soft errors also due to that we can reduce the area occupation which makes it

compatible with many common standards, like Wi-Max, WLAN, 3GPP2, GSM and LTE.

Keywords: Viterbi Decoder; Flexible Architecture; BMU; CSA; Trace Back

I. INTRODUCTION

In telecommunication, a convolutional code is a type of error-correcting code in which each mbit information symbol (each m-bit string) to be encoded is transformed into an n-bit symbol, where m/n is the code rate ($n \geq m$) and

The transformation is a function of the last k information symbols, where k is the constraint length of the code.

In telecommunication and information theory, forward error correction (FEC) (also called channel coding) is a system of error control for data transmission, whereby the sender adds

systematically generated redundant data to its messages, also known as an error-correcting code (ECC).

The carefully designed redundancy allows the receiver to detect and correct a limited number of errors occurring anywhere in the message without the need to ask the sender for additional data. FEC gives the receiver an ability to correct errors without needing a reverse channel to request retransmission of data, but this advantage is at the cost of a fixed higher forward channel bandwidth. FEC is therefore applied in situations where retransmissions are relatively costly, or impossible such as when broadcasting to multiple receivers. The process of adding this redundant information is known as channel coding.

Convolutional coding and block coding are the two major forms of channel coding. Convolutional codes operate on serial data, one or a few bits at a time. Block codes operate on relatively large (typically, up to a couple of hundred bytes) message blocks. There are a variety of useful convolutional and block codes, and a variety of algorithms for decoding the received coded information sequences to recover the

original data. Convolutional codes are employed to implement FEC.

Convolution codes are used extensively in numerous applications in order to achieve reliable data transfer, including, digital, video, radio, mobile communication and satellite communication. convolutional encoder is used to obtain convolution codes .It take a single or multi-bit input and generate a matrix of encoded outputs.

In digital modulation communications systems (such as wireless communication systems, etc.) noise and other external factors can alter bit sequences. By adding additional bits we make bit error checking more successful and allow for more accurate transfers. By transmitting a greater number of bits than the original signal we introduce a certain redundancy that can be used to determine the original signal in the presence of an error. Several algorithms exist for decoding convolutional codes. For relatively small values of k , the Viterbi algorithm is universally used.

A Viterbi decoder uses the Viterbi algorithm for decoding a bitstream that has

been encoded using forward error correction based on a convolutional code. There are other algorithms for decoding a convolutionally encoded stream (for example, the Fano algorithm). The Viterbi algorithm is the most resource-consuming, but it does the maximum likelihood decoding. Here we designed the two stage convolutional encoder and viterbi decoder and will implement in FPGA. For our illustration we will assume a 4-bit and give as an input to the convolutional encoder rate-1/2 code (two output bits for every input bit). This will yield a 2x4 output matrix, with the extra bits allowing for the correction. This 8 bit output is given as the input to the viterbi decoder which decodes the convolutional codes into original data.

II. VITERBI DECODER REVIEW

Viterbi algorithm is a well-known maximum likelihood algorithm for decoding convolutional codes. Fig. 1 indicates three basic elements of the encoding decoding system that is a convolutional encoder, communication channel and Viterbi decoder.

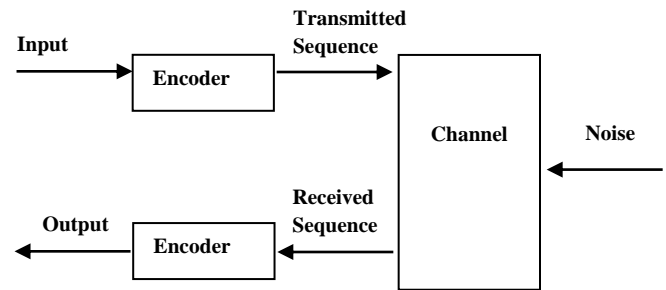


Fig. 1: Encoding decoding system

A. Convolutional Encoder

A convolutional encoder is a linear system [12] and operates on a continuous stream of data using shift registers. A convolution encoder takes an input stream of data and produces encoded output streams to be sent over wireless channel. In this method, the encoder generates more than one output bits, for one input bit and these redundant bits signs in the output bits creates the transmitted data more resistant to the noise in the channel [13], [14]. The redundant bits aid to select and correct the errors in the received data by Viterbi decoder.

The general block diagram of convolutional encoder is given in Fig. 2 where C is the input and X and Y are the encoded output [15]. If the encoder produces a group of n encoded bits per group of k information bits, the code rate R is widely

defined as $R=k/n$. Fig. 2 has a coding rate of $R=1/2$.

The circuit shown in Fig. 2 contains of two series D-Flip-Flops working at the same clock. Data enters at point C and each bit goes through points C, B and A in three clock periods. The output bits X and Y are shaped by exclusive-OR gates

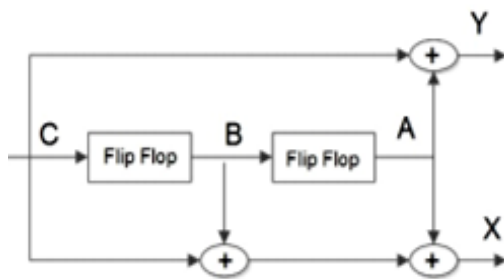


Fig. 2: Block diagram of convolutional encoder with $R=1/2$

The encoder can be denoted using a trellis diagram and a state diagram. Fig. 3 shows the state diagram of the same convolution encoder given in Fig. 2. The state diagram displays the state information of a convolutional encoder.

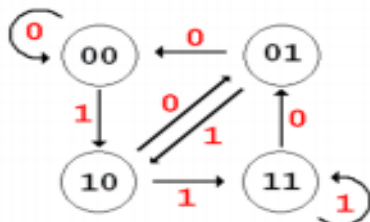


Fig. 3: State diagram of convolutional encoder with $R=1/2$

B. Viterbi Decoder

Viterbi decoder consists of the three major parts: a Branch Metric Unit (BMU), an Add-Compare-Select Unit (ACSU) and a Path Memory Unit (PMU) as shown in Fig. 4. The BMU calculates the hamming distance between the received code word and the expected word. The ACSU updates path metrics by adding branch metrics to the path metrics, and then it selects the smaller path metric as a survivor path. The PMU stores the survivor path metric for trace back.

There are two algorithms which can be used for decoding the convolution codes are Sequential algorithm and Viterbi Sequential algorithm. The Viterbi algorithm is commonly utilized for many wireless communication systems. Viterbi decoding has a fixed decoding time compared to sequential decoding [16].

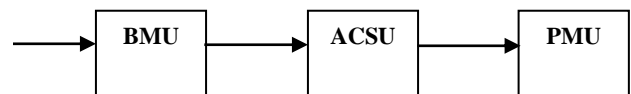


Fig. 4: Block diagram of Viterbi decoder.

III. THE PROPOSED FLEXIBLE ARCHITECTURE

The proposed flexible Viterbi decoder main blocks are shown in Fig. 5. It is designed to operate on a butterfly (two states at a time). It contains a branch metric unit, an add-compare select unit (ACS will be replaced by CSA, explained later), and a multiplexer which select the output of the butterfly.

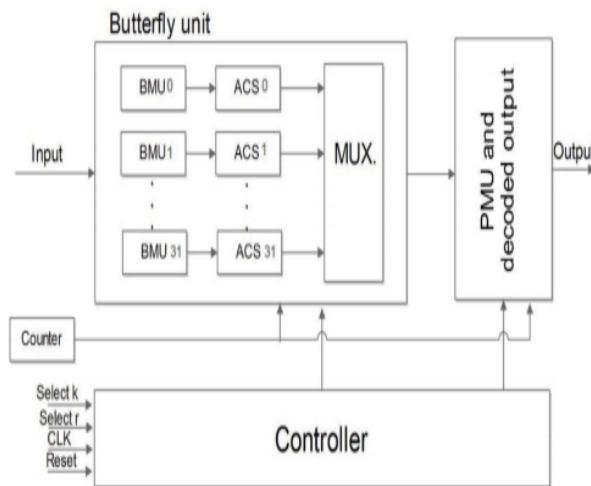


Fig. 5: Block diagram of the proposed flexible Viterbi decoder

The inputs to the controller are the select k which change the constraint length of the decoder and select r which controls the code rate of the decoder. Select k and select r are the selection inputs to the decoder which determine the mode of

operation of the decoder according to the wireless communication standards. The operation of the decoder and the behavior of the different components of the proposed flexible Viterbi decoder will be discussed in further details in the following sub-sections.

A. The Butterfly Block

The Butterfly block is a compilation of the branch metric block and the ACS block. There are thirty two butterflies in the proposed Viterbi decoder, where each butterfly resembles two states as illustrated in Fig. 6.

The main loop of the algorithm for the butterfly has two main steps: first, computing the branch metric for the next set, and second, calculating the path metric for the next column. The path metric calculation can be considered as an add-compare-select procedure as follows:

- The path metric for the old state is added to the corresponding branch metric.
- The sums for paths reaching the new state are compared (each state has only two paths to compare).
- The path with the smallest value is selected.

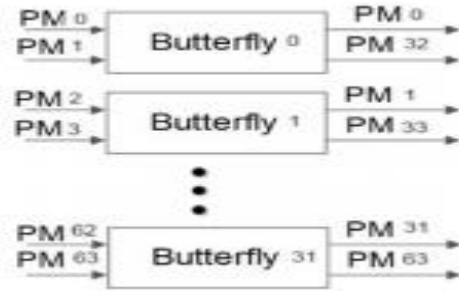


Fig. 6: The butterfly Blocks

A.1 Branch Metric Unit

The branch metric unit calculates the branch metrics from the input data as shown in Fig. 7 [17]. Hamming distance is a traditional and easy method to calculate metric in hard decision. The Hamming distance between the received word and the expected word is computed by comparing the received word symbol with the expected word symbol and counting the number of the different bits as shown in Table 2. This unit is capable of calculating the branch metrics of the eight states from (000) to (111) for every input data at rates 1/2 and 1/3.

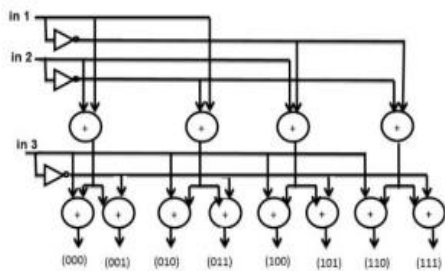


Fig. 7: Architecture of BMU

Table 2: The Hamming Distance

Received bits	BM	BM	BM	BM	BM	BM	BM	BM
	000	001	010	011	100	101	110	111
000	0	1	1	2	1	2	2	3
001	1	0	2	1	2	1	3	2
010	1	2	0	1	2	3	1	2
011	2	1	1	0	3	2	2	1
100	1	2	2	3	0	1	1	2
101	2	1	3	2	1	0	2	1
110	2	3	1	2	1	2	0	1
111	3	2	2	1	2	1	1	0

A.2 Add-Compare-Select unit

Figure 8 shows the operation of the add compare select (ACS) unit. Each state uses two path metrics from the previous state to generate a new path metric and a decision bit for the current state. Each of the two incoming path metrics is added to the corresponding branch metric, resulting in two new values for the path metrics. These new values are compared to determine the smaller value, which is stored for the current state. The decision value is represented by a single bit since it only needs to indicate which of the two input paths was chosen. The path metric is calculated by the following formula.

$$PM_{out} = \min(PM1 + BM1, PM2 + BM2) \quad (1)$$

To make the ACS unit flexible, a 4-input multiplexer is added to feed the output of the butterfly. The output of the multiplexer is controlled by the select K input of the decoder controller; for example:

$$PM2 = \text{mux}(\text{select } K, PM2, PM8, PM32) \quad (2)$$

If select K is “00”, (meaning constraint length 3),

$$PM2 = PM2$$

Else, select K is “10”, (meaning constraint length 5),

$$PM2 = PM8$$

Else, select K is “11”, (meaning constraint length 7),

$$PM2 = PM32.$$

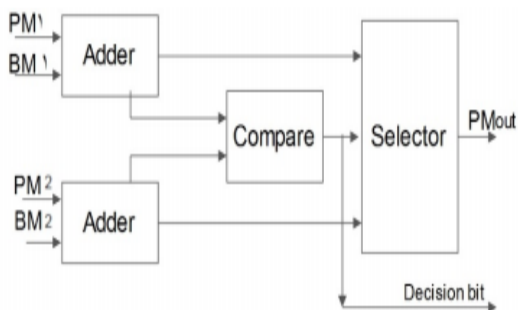


Fig. 8: Architecture of ACS

From Fig. 8, it can be concluded that each ACS operation needed two adder, one

comparator and two select operations. The increase in the number of operations of an ACS unit leads to higher power consumption and larger areas.

This makes the design unsuitable for high speed, low power applications [18]. Therefore, it is more reasonable to replace the ACS unit with compare select add (CSA) unit [18]. In this way, the two path metrics are compared to determine the lower value. Next, this value is added with two different branch metrics to get new path metrics [18]. These new path metrics would be used for the next iteration. Since the path metrics values are much higher than those of the branch metrics, only the path metrics values are used in the compare operation [18]. A block diagram of CSA architecture is illustrated in Fig. 9.

Replacing the ACS operation by CSA will lower the power consumption of the decoder making it more suitable for high speed low power applications as will be shown in the results.

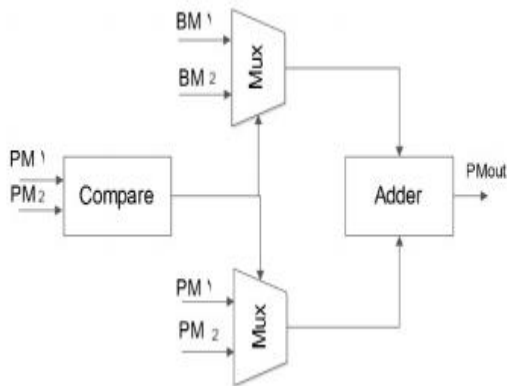


Fig. 9: CSA module used in the proposed design.

B. PATH MEMORY UNIT (PMU)

Using path memory unit (PMU) stores the survivor path of each state as a matrix form for trace back. In this matrix, trace back operates from the final state which has a minimum path metric to the initial state to estimate the input, next, the decoded output is created. The main steps for this PMU block are:

- Obtain the survivor paths for input bits.
- Trace back operates from the end of survivor paths to the beginning.
- Send decoded bits to the output.
- Obtain the survivor paths for other input bits.

There are two algorithms for Viterbi decoder: register exchange and trace back. In this proposed design, we decide to use

trace back because trace back is proper for reconfiguration purposes.

In every one of the executions of the SOVA detailed in the writing, a two-stage process is embraced. In the initial step, hard Viterbi calculation is connected to discover the ML way in the trellis. Once the ML way is resolved, trace back is completed along the ML way by refreshing the unwavering quality estimations of the bits to be recognized by (1). In this paper, we propose a solitary stride sliding piece SOVA indicator, the square schematic of which is appeared in Fig 10. Here, both the hard yield and the delicate unwavering quality esteems are produced in an enlist trade way, dispensing with the trace back procedure totally.

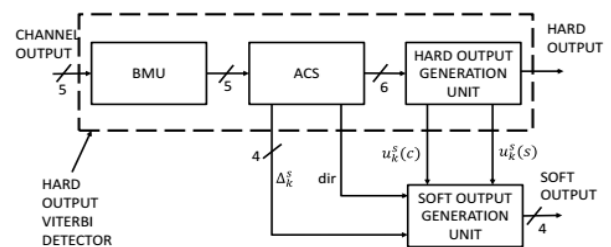


Fig 10: Square Schematic

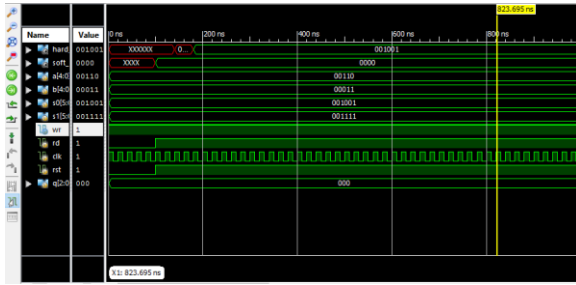


Fig 11: Simulation Results

Table 3: Utilization Summary

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	50	35200	0%
Number of Slice LUTs	74	17600	0%
Number of Fully used LUT-FF pairs	38	86	44%
Number of Bonded IOBs	39	100	39%
Number of BUFG/BUFGCTRLs	1	32	3%

IV. CONCLUSION AND FUTURE SCOPE

In this paper we presented a design of a reconfigurable Viterbi decoder. The decoder has an architecture that supports varying constraint lengths and code rates which make it compatible with many common standards like WiMAX, WLAN, 3GPP2, 3GPP, GSM and LTE. Thus the proposed reconfigurable Viterbi decoder is suitable for multi-standard receiver. The architecture of the decoder replaces the ACS

unit with a CSA unit. This modified decoder consumes less power and area. The proposed design using CSA reduces the power consumption by 26%. Also, it reduces the area by 21% compared to traditional design.

REFERENCES

- [1] N. Bhatt, M. Shah, and B. Asodariya, "FPGA Implementation of Power Efficient Low Latency Viterbi Decoder," in International Journal of Engineering Research & Technology (IJERT), vol. 2, no. 5, May 2013.
- [2] P. GOHEL, and K.C. DAVE, "Implementation of Viterbi Decoder on FPGA to Improve Design," in International Journal for Scientific Research & Development (IJSRD), 2013.
- [3] A. Kulkarni, D. Mantri, N.R. Prasad, and R. Prasad, "Convolutional encoder and Viterbi decoder using SOPC for variable constraint length," in Advance Computing Conference (IACC), 2013 IEEE 3rd International, Ghaziabad, 2013, pp. 1651-1655.
- [4] H. Pujara, and P. Prajapati, "RTL Implementation of Viterbi Decoder using



VHDL," in IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), vol. 2, no. 1, pp. 65-71, Mar. – Apr. 2013.

[5] R. V. W. Putra, and T. Adiono, "in A configurable and low complexity hard-decision viterbi decoder in VLSI architecture," in Information and Communication Technology (ICoICT), 2014 2nd International Conference, Bandung, 2014, pp. 182-186.

[6] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," in IEEE Trans. Inform. Theory, Vols. IT-13, pp. 260-269, Apr. 1967.

[7] R .D. Kadam and S. L. Haridas, "FPGA Implementation of Soft Output Viterbi Algorithm Using Memoryless Hybrid Register Exchange Method," in International Journal of VLSI design & Communication Systems (VLSICS), vol. 2, pp. 51- 59, September 2011, pp. 52-59.

[8] A. Viterbi, "Convolutional codes and their performance in communication systems," in IEEE trans. On communication technology, Vols. com-19, 1971.

[9] P. J. Black and T. H. Meng, "A 140-Mb/s, 32-state, radix-4 Viterbi decoder," in IEEE Journal of Solid-State Circuits, vol. 27, pp. 1877-1885, 1992.

[10] X. Corporation, "Viterbi decoder: product specification (v6.2)," October 2007.

[11] L.ZHOU, M. TANG, D. LIU, and H. LIU, "A Flexible Viterbi Decoder for Software Defined Radio," in Journal of Theoretical and Applied Information Technology, vol. 47, pp. 702- 706, 2013.

[12] S. B. SUKHAVASI, S. B. SUKHAVASI, H. Khan, and C. Pilla, "Performance Evaluation of Turbo Codes Using Hard Decision Viterbi Algorithm in VHDL," in International Journal of Engineering Research and Applications (IJERA), vol. 2, May-Jun 2012, pp. 2846-2891.

[13] M. Jabeen, and S. Khan, "Design of Convolution Encoder and Reconfigurable Viterbi Decoder," in International Journal of Engineering and Science, vol. 1, Sept 2012, pp. 15- 21.

[14] V. Soni, and S. Nemade, "FPGA Design for Efficient Architecture for the Convolution Encoder," in International

Journal of Engineering Trends and Technology (IJETT), vol. 9, no. 2, pp. 74-77, 2014.

[15] C. R. Kumar, and P. Latha, "Area and Power Efficient Viterbi Decoder for Storage Devices," in IJREAT International Journal of Research in Engineering & Advanced Technology, vol. 1, no. 1, 2013, pp. 1- 5.

[16] H. Pujara, and P. Prajapati, "Development of IP Core of Viterbi Decoder using VHDL," in International Global Research Analysis, vol. 2, no. 4, pp. 130- 132, April 2013.

[17] Chu Yu and Yu-Shan Su, Po-Hsun Cheng, Bor-Shing Lin, Sao-Jie Chen, "A Memoryless Viterbi Decoder for LTE Systems," in The 1 st IEEE global conference on consumer electronics, 2012.

[18] S. Bhowal, "Speed, Noise Immunity, Power Consumption and Area Comparison between Different Approaches of Low-Power Viterbi Decoder for Digital Wireless Communication Applications," in Network Protocols and Algorithms, vol. 6, pp. 19- 36, 2014.



Author details:-

Aparna Busi received her B.Tech Degree in Electronics And Communication Engineering, from BVSR Engineering College, Chimakurthi (Affiliated to JNTU, Kakinada). She is presently Pursuing M.Tech (VLSI & ES) in QIS Institute of Technology, Ongole, AP, India. Her current research interests are VLSI, Embedded Systems, digital signal processing and Communication.



Latha .K received her B.Tech Degree in Electronics and communication Engineering, from Rao & Naidu Engineering college, Ongole (Affiliated to JNTU, Kakinada). She received her M.Tech degree in Digital Electronics and Communication systems from QIS College of Engineering & Technology, Ongole.AP, India. She is presently working as Assistant Professor in QIS Institute of Technology, ongole A.P. Her Current research interests are Digital signal processing and communication.

Mr. Shaik. Rahamtula is currently working as Associate Professor in ECE Department, QIS Institute of technology, Ongole, A.P, India. He received his B. Tech degree in the department of Electronics and Communication Engineering, from KMCET



(Affiliated to JNTU Hyderabad). He received his M. Tech from QIS college of Engineering and Technology (Affiliated to JNTU Kakinada). His research interests in the area of Face recognition, Content based image retrieval, Image compression, VLSI, ES.