International Journal of Engineering Research and Development
ISSN: 2278-067X, Volume 1, Issue 1 (May 2012), PP. 37-41
www.ijerd.com

Constraint Length Parametrizable Viterbi Decoder for Convolutional Codes

P.Deepak Vamsi[1], Mr. M.Murali Krishna[2]
[1]*PG Scholar,* [2]*Assistant professor, Department of ECE, GIT, GITAM University, Vishakhapatnam,*

***Abstract*—** Convolutional codes are the widely used as Forward Error Correction (FEC) codes that are used in robust digital communication system. The parameterized implementation of a Viterbi decoder is presented in this paper where we can fix the constraint length for a code rate of $1/2$. This improves the decoding performance in area, accuracy and computational time. Viterbi algorithm is the widely employed in wireless communications to decode the convolutional codes. The proposed architecture is can be reconfigured to decode with constraint length ranging from 3 to 7.The main theme of this paper lie in the architecture of the computational blocks .In this paper, a generic, reconfigurable[2], low power[8] and area efficient Viterbi decoder for software defined radio is described using a VERILOG code for FPGA implementation. The proposed design of the Viterbi decoder is considered to be generic so that it facilitates the prototyping of the decoder with different specifications. The proposed design is implemented on Xilinx Virtex5, Xc5vlx30 FPGA using the FPGA Advantage Pro package provided by ISE 10.1 by Xilinx.

Keywords*—FPGA; Constraint length ; Viterbi decoder.*

I. INTRODUCTION

This paper deals with the implementation of convolutiuonal codes, a type of FEC (Forward Error Correction) schemes that are widely used in the wireless communications. The various channel noise that are introduced in the channels during transmission of the data, corrupts the data which is received at the receiver end. These FEC schemes are used to reconstruct the original data from the corrupted data. The re-transmission of the data will lead to loss of power, speed and mainly time. Hence we can get the original data from the corrupted data without any re-transmission of the data again and again by using these FEC schemes.FEC schemes are classified into convolutional codes and block codes. In block codes direct mapping is done between the data and the code symbols. In convolutional type the code symbols are determined based on the current and the previous data by using the memory elements.

In this paper we deal with convolutional codes where decoding is done using the viterbi algorithm. Viterbi algorithm is the widely used decoding algorithms for the convolutional codes. Viterbi algorithm is based on the Maximum likelihood algorithm. This method works on the principle of Hidden Markov's model where the most probable sequence of the hidden states is based on the given sequence of observed outputs. Constraint length of the code defines the no of shifts in the convolutional encoder that can influence the output. The architecture implemented in this paper would support this parameterization of the constraint length and is implemented on FPGA.

Generations, like in Vertix-5 FPGAs by Xilinx, that are optimized for high-performance logic and DSP with low power serial connectivity are brought up. Many sophisticated signal processing tasks are performed and can be implemented on FPGA, including advanced compression algorithms, channel estimation, power control, forward error control, synchronization, equalization, and protocol Management… etc .

Channel coding is considered as one of the challenging signal-processing tasks in wireless communications. Most of the digital communication systems use convolutional coding to compensate the Additive White Gaussian noise (AWGN) and the effects of other data degradation like channel fading and quantization noise. Viterbi algorithm for convolutional codes decoding has been found to be efficient and robust for the advantage that it has a fixed decoding time and it well suited for hardware implementation.

The complexity of the Viterbi decoder grows exponentially with the increase of the number of the states of the convolutional encoder that relates to the constraint length. Hence better designs of the decoder have to be found to realize convolutional decoder. The aim of any decoder realization is to reduce the area, reduce the power consumption or to enhance the performance. While reducing the area can be gained by the advances reached in electronics, low power designs [3] have to be developed especially because they are important issues for mobile and portable applications. This paper proposes constraint length, re-configurable[2], low power architecture[3] for developing and modeling a generic and configurable Viterbi decoder .The design is described using a VHDL code for the implementation on FPGA hence it can be reconfigurable. It can also be fabricated as a low power Viterbi decoder on ASIC for either a base station or portable unit. In this paper, we considered the design of our decoder to be generic so that it facilitates the prototyping of the decoder with different specifications. The remainder of this paper is organized as follows: the following section introduces a quick overview on the Viterbi algorithm followed by the architecture of the proposed Viterbi decoder and the results.

II. VITERBI ALGORITHM

The Viterbi algorithm proposed by A.J. Viterbi is based on the maximum likelihood decoding algorithm. Viterbi algorithm is implemented as per the trellis. The depth of the trellis is depending upon the constraint length and the code rate of the encoder. It is well suited for decoding of convolutional codes. It finds a branch in the code trellis most likely corresponds to the transmitted one. The depth of the trellis is equal to the constraint length K. The algorithm is based on calculating the Hamming distance for every input and the expected output of each branch along the path that is most likely through the trellis will maximize that metric. The complexity is reduced by finding the least path metric between the branches entering the same node(choosing the least likely path) at each stage. The path metric that is least likely (chosen) is known as the survivor, while the other paths entering the same node are non-survivor paths. If two or more paths is having the best(least likely)path metric then the surviving path is chosen at random. The selection of survivor paths is the main theory in the Viterbi algorithm and ensures that the algorithm terminates with the maximum likelihood path [4]. The algorithm stops where all the nodes in the trellis have been processed and their survivor paths are determined. At the final stage of the trellis after the survivor path of the nodes have been determined then trace-back through the trellis is done. The trace back mechanism starts with selecting the node at the final stage of the trellis that has the least path metric weight. At any given node, we can only continue backward on a path that survived upon entry into that node. Our trace back mechanism yields a unique path since each node has only one entering survivor .
The outptut of the trace back unit is the decode output i.e the message signal. This path is the maximum likelihood estimate that predicts the most likely transmitted sequence. The maximum likelihood equation [3] is given by:

$$P\left(\frac{A}{K(m')}\right) = \max P\left(\frac{A}{K(m)}\right) over\ all\ K(m)\ (1)$$

A: is the received sequence
K (m): one of the possible transmitted sequences, and chooses the maximum.

Coding schemes like GPRS, EDGE and WiMAX uses various coding schemes for wireless packet data network to maximize the channel capaxity. GPRS uses a constraint length 7 and rate 1/2 Viterbi decoder, while EDGE uses a constraint length 7 and rate 1/3 with both tail biting Viterbi on the header portion and zero tail on data portion . WiMAX 802.16e, has the Viterbi with constraint length 7 and rate ½ with tail biting as mandatory and zero tail as optional. This paper focuses on the parametrization of constraint length ,high speed and low power Viterbi decoder with constraint length K = 7 and code rate r = ½.The trace-back depth depends mainly on the memory management of the algorithm. The longer the trace-back depth the larger the trellis will grow, and the larger the memory requirements. There is a trade off that if the trace-back depth is also made too short; the performance of the codes will be affected.

III. ARCHITECTURE OF THE VITERBI DECODER

The decoder architecture is implemented according to the the trellis diagram. This paper is implemented based on the hard decision decoding. Viterbi decoder architecture [1] is mainly depending on the organizing the memory with computational units like the ACS, BMG, and mainly on the Trace Back mechanism that is being adopted in the design. The decoder architecture is implemented according to the trellis diagram. The inputs to the receiver are a continuous stream of analog, modulated signals. The primary tasks of the receiver are the recovery of the carrier and bit timing so that the individual received data bits can be removed from the carrier and can be separated from one another in an efficient manner, which is performed by the use of phase locked loops. The analog baseband signals are applied to the A/D converter with quantizer to get a received bit stream. Then obtained bit stream is applied as the input to the Viterbi decoder. In order to compute the branch metrics at any given point in time, the Viterbi decoder must be able to segment the received bit stream into n-bit blocks, each block corresponding to a stage in the trellis. In this paper, we assume that the input to our proposed design is an identified code symbols, (i.e. binary data) the design decodes successive binary bit stream. The Viterbi decoder consists of some basic building blocks as illustrated in Fig.1.

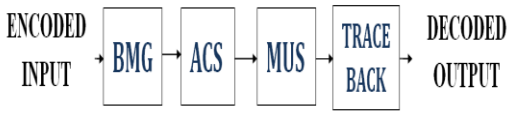

Fig.1 Viterbi decoder block diagram

A. The Branch Metric generator (BMG)

Thy Branch Metric generator computes the hamming distance between the received input sequence and the expected input sequence. In this paper Branch metric generator is based on a truth table containing the various bit metrics. The truth table implementation [2] is chosen in order to reduce the memory elements in the design there by have less area...Etc. The computer looks up the n-bit metrics associated with each branch and sums them to obtain the branch metric. These results are passed to the path metric update and storage unit. Fig.2 shows the block diagram of the BMC.

Fig.2 Branch metric Block

B. The Path Metric

The path metric is the sum of the branch metric of the present input and the path metric of the previous iteration. This takes the branch metrics computed by the BMG and computes the partial path metrics at each node in the trellis. These Metrics are stored in the memory elements for making use of them in the next iterations and also to find the min path metric at the last stage(iteration) of the trellis.

The surviving path at each node is identified, and the information-sequence updating and storage unit. Since the entire trellis is multiple images of the same simple element, an Add-Compare- Select may be assigned to each trellis state[4].

C. Add-Compare-Select (ACS)

ACS circuit is used repeatedly in the decoder for the purpose of the calculation of the new branch metrics and also for updating the memory elements with least branch metrics. The no of times the ACS circuit used is depending on the constraint length. As the constraint length increases the trellis stages increases hence the no of times the ACS used also increases. Constraint length K= no of stages of the trellis = no of times the ACS is computed.

The entire decoder can be based on one such ACS, resulting in a very slow, but low-cost, implementation. Hence to enhance the speed of the decoder implemented in parallel [8]. A separate ACS circuit is dedicated to every element in the trellis, resulting in a fast, massively parallel implementation. For a given code with rate 1/n and total memory$M$, the number of ACS required to decode a received sequence of length L is $Lx2M$[8]. In our implementation we combined both the BMC and the ACS in one unit representing a single wing of each trellis butterfly and we refer to it in this paper as ACS module as illustrated in Fig.3.

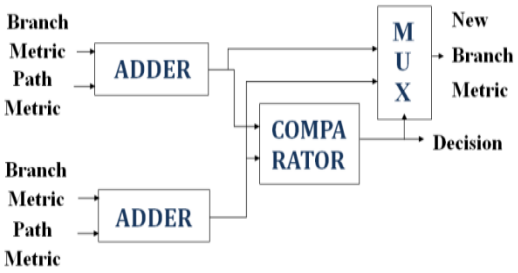

Fig.3 Add/Compare select unit

D. Path Metric, branch metric update and Storage (MUS)

This is responsible for keeping track of the information bits associated with the surviving paths designated by the path metric updating and storage unit. There are two basic design approaches: Register Exchange and Trace Back. In both techniques, a shift register is associated with every trellis node through out the decoding operation. Each state is assigned a register which has a length equal to the frame length. The register exchange may offer high-speed operation but it is not power efficient while the trace-back is complex but has the advantage of low power dissipation because the switching activity of the registers is much lower than that of the register exchange. Since one of the major interests is the low power design, the proposed decoder in this paper has been implemented using the trace-back approach. This feature is very helpful in our design, since the trace-back module does not work until the end of each frame and hence we don't have to activate the trace-back module until the end of the frame and only for one clock cycle, by this feature, more saving in dissipated power.

F. Parallel-to-Serial Interface (PtoS)

P to S converter gets the information (decode sequence) from the trace back unit and gives the output serially. The bottleneck of this approach is the P to S converter that limits the speed of this architecture.

IV Architecture Implemented

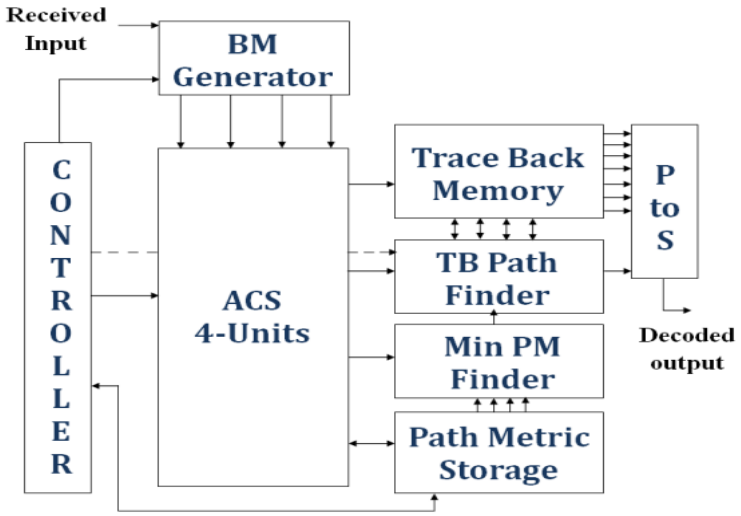

Fig.4 Architecture implemented

Fig.4 shows the architecture that has been implemented in the paper. The implementation of this mainly concentrates on parameterization of the constraint length K through which the trace back is processed. The architectural changes [1] that are done in the memory organization and the low power efficient design [8] are been implemented to get better results. The memory organization is made efficient by using the dual edge trigger PM registers for saving the path metrics as each stage in the trellis is processed. The low power is achieved by just enabling the trace back only after the final stages has been processed.

V. RESULTS

Table.1 Summary of architecture implemented in this paper

Device Utilization Summary		
Logic Utilization	**Used/Available**	**Utilization**
Number of Slices	89/ 19200	1%
Number of Slice Flip Flops	153/ 19200	1%
Number of 4 input LUTs	153/ 19200	1%
Number of bonded IOBs	54/ 220	4%

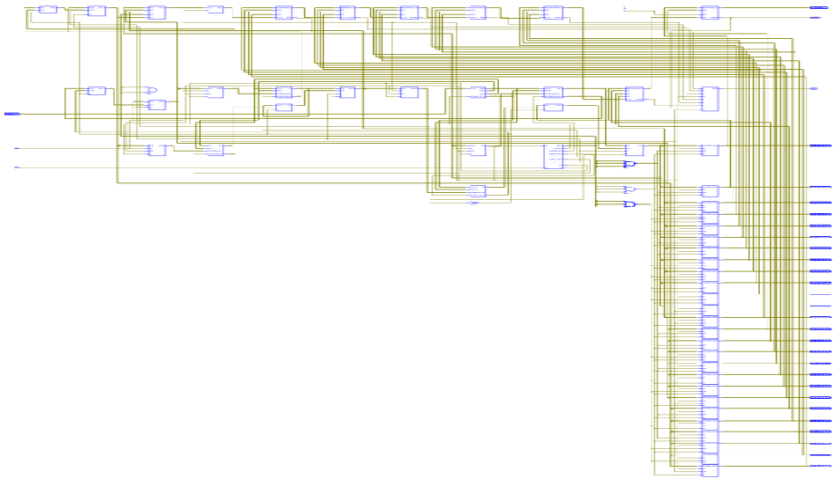

Fig.5 RTL Schematic of implemented design

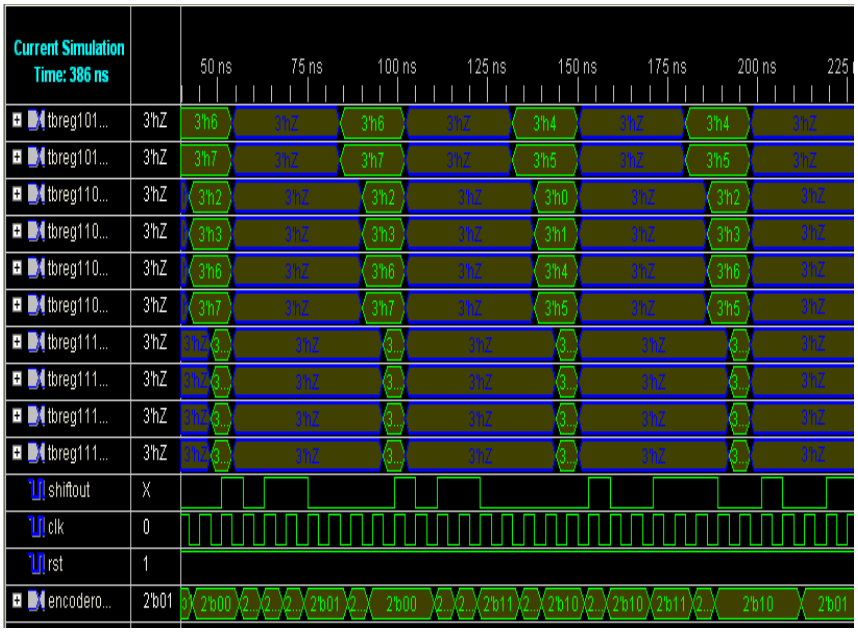

Fig.6 Simulation output of viterbi decoder with error in the input

VI. CONCLUSION

In this paper, a design for generic, low-power, area efficient and re-configurable Viterbi decoder has been proposed and its performance is evaluated on FPGA's. Many sophisticated signal processing tasks are performed and can be implemented on optimized FPGA's. The main theme of the paper lies in the parameterized constraint length that can be varied that can supports maximum constraint length of 7 for a code rate of ½ and the architectural imp0lementation i.e. the use of dual edge triggered registers. The low power technique has been achieved by the implementation of the trace back unit that does not work until the end of each frame and hence we don't have to activate the trace-back module until the end of the frame and only for one clock cycle, by this feature, more saving in dissipated power. The design has been described by structural architecture VHDL, targeted on Xilinx Virtex5, Xc5vlx30 FPGA using ISE design suit 10.1 by Xilinx. The maximum operating frequency is 125 MHz and the power consumed in 0.139 Watt that is found adequate to our application. The power report is obtained from Xpower analyzer. The performance of various parameters like area, power, speed…etc, on FPGA's are recorded and are listed in the below tables.My future work includes the implementation of the viterbi decoder for parameterized code rate.

REFERENCES

[1]. Man Guo, M. Omair Ahmad, M. N. S. Swamy, Chunyan Wang "FPGA Design and Implementation of a Low-Power Systolic Array-Based Adaptive Viterbi Decoder", IEEE Transactions on Circuits and Systems —I: Regular papers, vol. 52, no. 2, February 2005.

[2]. Behzad Momahed heravi, "Multi-rate Parameterized Viterbi Decoding for Partial Reconfiguration",ISBN: 1-9025-6013-9 ©2006 PGNet.

[3]. Sherif Welsen Shaker, Salwa HussienElramly, Khaled Ali Shehata, "FPGA Implementation of a Configurable Viterbi Decoder for Software radio Receiver", IEEE Transactions on Circuits and systems, IEEE,Sept 2009,pp.132-136.

[4]. Jinjin He, Huaping Liu, Zhongfeng Wang, "A fast ACSU architecture for Viterbi decoder uses T-algorithm", Forty-Third Asilomar Conference on signals, Systems and Computers, 2009, pp. 231-235.

[5]. Dong-Sun Kim D.S., Seung-Yerl Lee, S.Y., Kyu-Yeul Wang, K.Y., Jong- Hee Hwang, J.H., Duck-Jin Chung, D.J., "Power Efficient Viterbi Decoder based on Pre-computation Technique for Portable Digital Multimedia Broadcasting Receiver", IEEE Transactions on Consumer Electronics, Vol. 53, Issue. 2, 2007, pp. 350-356.

[6]. Ying Li, Weijun Lu, Dunshan Yu, Xing Zhang, "Design Technique of Viterbi Decoder in Satellite Communication", IET Conference on Wireless, Mobile and Sensor Networks, 2007, pp. 162-165.

[7]. Wei Shao, Brackenbury L., "Pre-processing of convolutional codes for reducing decoding power consumption", IEEE International Conference on Acoustics, Speech and Signal Processing, 2008 , pp. 2957 – 2960.

[8]. Swathi gupta, Rajesh mehra,"Reconfigurable Efficient Design of Viterbi Decoder for Wireless Communication Systems", IJCSNS International Journal of Advanced Computer Science and Applications, Vol. 2, Issue. 7, 2011.