ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 12, April 2018

STUDY ON THE DESIGN OF THE HIGH VITERBI DECODERS

[1]Ms.Athira T G, [2]Ms.Suvitha P S
[1]PG Scholar (VLSI Design), [2]Assistant Professor, Department of ECE
[1,2]IES College of Engineering, Thrissur, Kerala, India
[1]athirathayyil121@gmail.com
[2]suvithabhilash@gmail.com

***Abstract*—The use of error-correcting codes has proven to be an effective way to overcome data corruption in digital communication channels. Although widely-used, the most popular communications decoding algorithm, the Viterbi algorithm, requires an exponential increase in hardware complexity to achieve greater decode accuracy.Viterbi algorithm is employed in wireless communication to decode the convolutional codes; those codes are used in every robust digital communication systems. Such decoders are complex & dissipiate large amount of power. In this paper, describe the analysis and implementation of a reduced-complexity decode approach, the Viterbi algorithm (VA).The decoder design is implemented on Xilinx,modelsim using VHDL code.**

Keywords— Viterbi algorithm,convolutional code, VHDL,Xilinx

I. INTRODUCTION

Most digital communication systems nowadays convolutionally encoded the transmitted data to compensate for Additive White Gaussian Noise ie AWGN, fading of the channel, quantization distortions and other data degradation effects. For its efficiency the Viterbi algorithm has proven to be a very practical algorithm for forward error correction of convolutionally encoded messages. The requirements for the Viterbi decoder or Viterbi detector depend on the applications used. Most of the researches work to reduce cost, the power consumption, or work with high frequency for using the decoder in the modern applications such as 3GPP, DVB, and Wireless communications. Some of them comparing between using FPGA, ASIC, and DSP to find which one is suitable for the applications, other studies the differences method for back trace unit to find the correct path, and the other trying to work with high frequency by using parallel operations of
decoder units . The complexity of these decoders increased with the increasing of the constraint length. convolutional codes become more powerful,the complexity of corresponding decoders generally increases. The Viterbi algorithm [9], which is the most extensively employed decoding algorithm for convolutional codes, works well for less-complex codes, indicated by constraint length K. However, the algorithm's memory requirement and the computation count pose a performance obstacle when decoding more powerful codes with large constraint lengths.This algorithm reduced the average number of computations required per bit of decoded information. hardware requirements for our decoder grow predictably with constraint length at a rate substantially less than the exponential growth exhibited by standard Viterbi algorithms.

The Viterbi algorithm process is similar to finding the most likely sequence of states, resulting in sequence of
observed events and, thus, boasts of high efficiency as it consists of finite number of possible states. Viterbi decoders are composed of three major components: branch metric unit (BMU), add-compare-select (ACS) unit, and survivor path memory unit (SMU). BMU generates the metrics corresponding to the binary trellis depending on the received signal, which is given as input to ACS which, then, updates the path metrics.SMU is responsible for managing the survival paths and giving out the decoded data as output. BMU and SMU units happen to be purely forward logic. ACS recursion consists of feedback loops. Therefore, the speed is limited by the iteration bound . Thus, the ACS unit becomes the speed bottleneck for the system.

II. EXISTING METHODOLOGY

Viterbi algorithm was devised by Andrew J. Viterbi in 1967. The optimality and the relatively modest complexity ,for small constraint lengths have served to make the Viterbi algorithm the most popular in decoding of convolutional codes with constraint length less than the 10. Viterbi algorithm is called an optimum algorithm because it minimizes the probability of the error. The Viterbi algorithm is one of the standard sections in number of high speed modems of the process for the information infrastructure applicable in modern world. The unit of branch metric will calculate all the branch metrics and then processed to add compare for selecting the surviving branches as per the branch metrics finally the decoded data bits are generated by the trace back unit.

The Viterbi algorithm can divided into following steps

Step 1:

- Finding the metric value for each node.
- Initially metric value becomes zero

Step2:

- After stage 2 each node receiving to path,each path having single metric value
- After second state node have 2 metric value

Step3:

- find the survivor path with help of minimum metric value

Step4:

- Step 2 &3 repeat until the received bit sequence stop

Step5:

- finally find the active path.
- This active path code word is the corrected codeword & find the original message sequence

III. ARCHITECTURE OF VITERBI DECODER

The input to our proposed design is an identified code symbols and frames, i.e. The design decodes successive bit stream and the proposed decoder has no need to segment the received bit stream into n-bit blocks that are corresponding to a stage in the trellis in order to compute the branch metrics at any given point in time . The block diagram of the viterbi decoder is illustrated in figure 1.

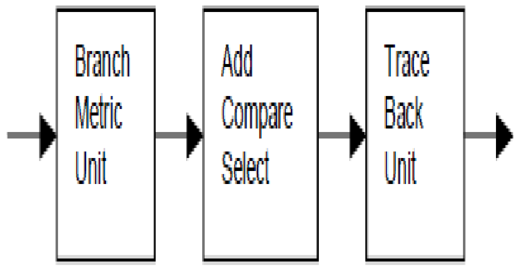

Fig 1.Block diagram of the Viterbi decoder

The basic performance of the Viterbi decoder isanalyzed with the block diagram shown below. It consists of three main blocks-

1. Branch Metric Unit
2. Path Metric Calculation(ACS,PMU)
3. Trace Back Unit

A. The Branch Metric Unit(BMU)

This is typically based on a look-up table containing the various bit metrics. The computer looks up the n-bit metrics associated with eachbranch and sums them to obtain the branch metric.The result is passed along to the path metric Calculation. The responsibility of this unit is to compute the Hamming code between the expected code and the receiving code as a frame. At each processing, the BMU finds the Hamming code for these symbols.

B. Path Metric Calculation

There are Path Metric Unit (PMU) and Add Compare Select Unit(ACSU) blocks in it.

1. Path Metric Unit (PMU)

It computes the partial path metrics at each node in the trellis.

2. Add Compare Select Unit (ACSU)

This ACSU is the main unit of the survivor path decoder . The function of this unit is to find theaddition of the Hamming code received from

BMU's and to compare the total Hamming code. This takes the branch metrics computed by the BMC and computes the partial path metrics at each node in the trellis. The surviving path at each node is identified, and the information-sequence updating and storage unit notified accordingly. Since the entire trellis is multiple images of the same simple element, a single circuit called Add- Compare-Select may be assigned to each trellis state.

C Trace Back Unit

This is responsible for keeping track of the information bits associated with the surviving paths designated by the path metric Calculation. There are two basic design approaches: Register Exchange and Trace Back. In both techniques, a shift register is associated with every trellis node throughout the decoding operation . Since one of the major interests is the low power design, the proposed decoder has been implemented using the trace back approach which dissipates less power. The major disadvantage of the RE approach is that its routing cost is very high especially in the case of long-constraint lengths and it requires much more resources.

IV.IMPLEMENTATION OF VITERBI DECODER

1.Branch Metric Unit:

BMU calculates the branch metric. For hard decision MLdecoding, the branch metric for a

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 5, Special Issue 12, April 2018

branch is the hammingdistance between the received code and the expected code atthat instant. BMU comprises of XOR gate and adder. TheBMU block is shown in the Fig 2

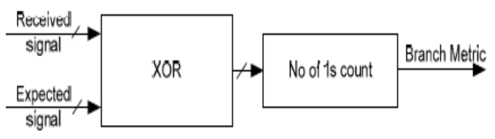

Fig 2 BMU Unit

2. Add Compare Select Unit

This structure contains a pair of source and destination states, and four interconnecting branches as shown in Fig. 3. In this Fig. 3 the upper or lower branch from each state A(B) is taken, when the corresponding origin input bit is ‘1’ or ‘0’. If the source input bit is ‘1’ or ‘0’, the next state for both A or B is C or D. On the basis of the butterfly notation the architectural block of ACSU can be decomposed as shown in Fig. 3. On the basis of the butterfly notation the architectural block of ACSU [2] can be decomposed as shown in Fig.7. The implementation of two wings of the butterfly module is same except the expected signal. The architecture has shown only one wing of the butterfly module. Branch metric is added with the path metric for state S0 (S2), which is equal to hamming distance between the received signal and the expected signal. The expected input is exclusive-or with received signal. ‘No. of 1’s Count’ Blocks counts the number of 1’s in its input.

Adder blocks add the branch metric and state metrics to create a new state metric. Comparator is used to compare the upper path metric and lower path metric for determining the survivor path. Multiplexer select line, connected with the output of the comparator block, selects the survivor path.

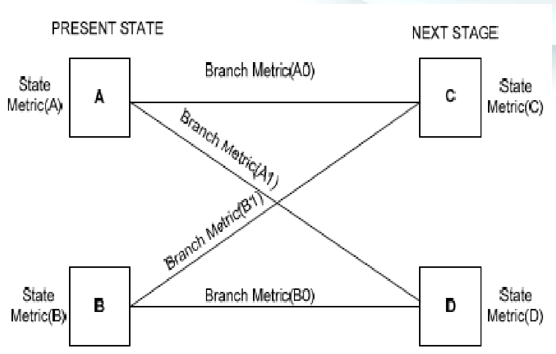

Fig 3 ACS Block updates of the node

3.Trace Back Unit

This block is needed only for trace back method. Two incoming branches are there for every state except head and tail part of trellis diagram. Between these two branches, it determines which branch will survive (lower or upper). For flagging the survivor path only one bit is enough. ACSU gives one decision bit for every survivor path. All the decision bits are stored in a memory. The size of the memory depends on both the number of ACSU used in proposed architecture and the trellis length.

V.RESULTS & DISCUSSIONS

The following section gives the software implementation results of the paper.Results are simulated using ModelSim software in VHDL language section by section.Xilinx software is used to measure the delay,area and power parameters.The

output of the viterbi encoder and decoder is shown in fig.4 and 5 below.

Synthesis report of the existing system analyzed by Xilinx software. Delay, area and power parameters are calculated using this software. Performance parameters like delay, power and area of the high speed Viterbi decoders found.Fig 6 shows the output of the Viterbi decoder.In this input is 8 bit number ie 11100110.Then decode it as the 2bit numbers and finally got the Viterbi decoder output in the simulation window as 01100111 in it.Reverse output is because of the traceback unit ie last bit came first in the decoder section.

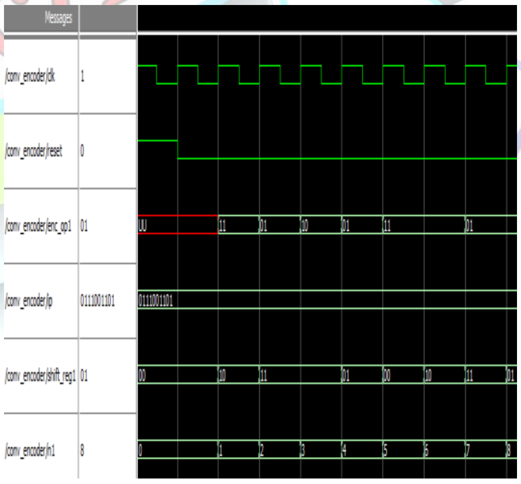

Fig.4.output waveform of convolutional encoder

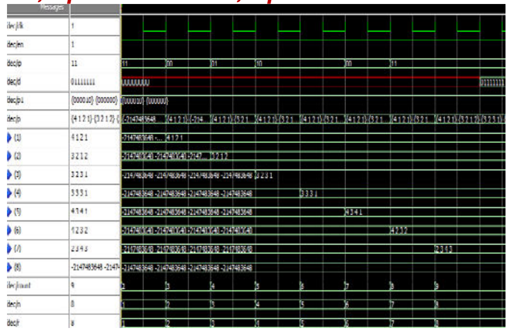

Fig.5. output waveform of decoder

Performance Analysis

	Area	Delay	Power
Viterbi decoder	12,232	16.60ns	564mW

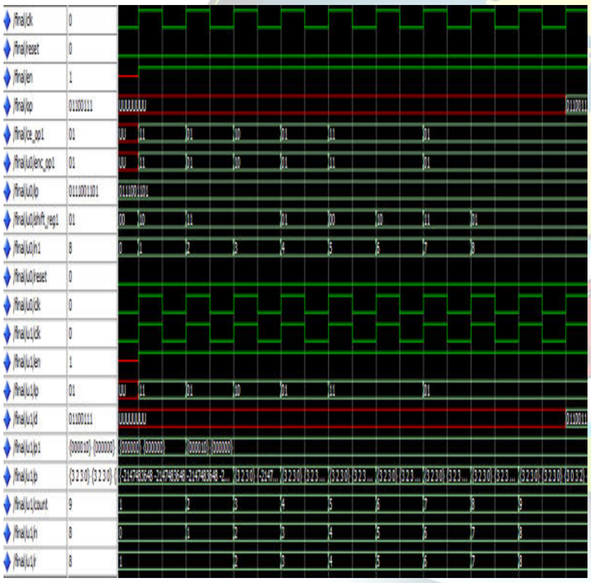

Fig.6. combined output waveform

VI.CONCLUSION

The simulation results of the existing Viterbi decoder verified using the Modelsim software in VHDL language.And also calculated the performance parameters like delay,area and power of the existing system.The survivor path algorithm used, the address of the memory unit to select the correct path which specify output code.mobile and wireless communication becomes increasingly ubiquitous, the need for dynamic reconfigure ability of hardware shall pose fundamental challenges for communication algorithm designers as well as hardware architectures and attempts to solve this problem for the particular case of the Viterbi decoder used in wireless communication systems.

ACKNOWLEDGMENT

I express my sincere thanks to my guide Ms.Suvitha P S for her valuable guidance and useful suggestions which help me throughout this work

REFERENCES

[1] R. Liu and K. Parhi, (2009)“Low-latency low-complexity architectures for Viterbi decoders,” *IEEETrans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 10, pp. 2315–2324

[2] G. Fettweis and H. Meyr,(1989) “Parallel Viterbi algorithm implementation: Breaking the ACS-bottleneck,” *IEEE Trans. Commun.*, vol. 37, no. 8, pp. 785–790

[3] V. Gierenz, O. Weiss, T. Noll, I. Carew, J. Ashley, and R. Karabed,(2000) “A 550 mb/s radix-4 bit-level pipelined 16-state 0.25-μm CMOS Viterbi decoder,” in *Proc. IEEE Int. Conf. Appl.-Specific Syst. Archit. Process.*, pp. 195–201.

[4] P. J. Black and T. H. Meng,(1992) “A 140-Mb/s, 32-state, radix-4 Viterbi decoder,” *IEEE J. Solid-State Circuits*, vol. 27, no. 12, pp. 1877–1885.

[5] T. Gemmeke, M. Gansen, and T. Noll, (2002)“Implementation of scalable power and area efficient high-throughput Viterbi decoders,” *IEEE J. Solid-State Circuits*, vol. 37, no. 7, pp. 941–948

[6] A. Yeung and J. Rabaey, (1995)“A 210 Mb/s radix-4 bit-level pipelined Viterbi decoder,” in *Proc.IEEE Conf. Int. Solid-State Circuits*, pp. 88–89.

[7] J. J. Kong and K. K. Parhi, (2004)“Low-latency architectures for high-throughput viterbi decoders,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 6, pp. 642–651

[8] D. Vasudevan, P. Lala, and J. Parkerson,(2007) “Self-checking carry-select adder design based on two-rail encoding,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 12, pp. 2696–2705,

[9] M. Akbar and J.-A. Lee,(2014) “Comments on ‘self-checking carry-select adder design based on two-rail encoding’,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 7, pp. 2212–2214

[10] M. Nicolaidis, (2003)"Carry checking/parity prediction adders and ALUs," *IEEE Trans. VeryLarge Scale Integr. (VLSI) Syst.*, vol. 11, no. 1, pp. 121–128

[11] C.-H. Yen and B.-F. Wu, (2006)"Simple error detection methods for implementation of advanced encryption standard," *IEEE Trans. Comput.*, vol. 55, no. 6, pp. 720–731

[12] M. Mozaffari-Kermani, R. Azarderakhsh, and A. Aghaie, (2015)"Reliable and error detection architectures of Pomaranch for false-alarm-sensitive cryptographic applications," *IEEE Trans.Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 12, pp. 2804–2812

[13] M. Mozaffari Kermani and R. Azarderakhsh,(2015) "Reliable hash trees for post-quantum stateless cryptographic hash-based signatures," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSISyst. (DFT)*,pp. 103–108.

[14] M. Mozaffari-Kermani and A. Reyhani-Masoleh,(2013) "Efficient fault diagnosis schemes for reliable lightweight cryptographic ISO/IEC standard CLEFIA benchmarked on ASIC and FPGA," *IEEE Trans. Ind. Electron.*, vol. 60, no. 12, pp. 5925–5932