International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 10
September 2017

Implementation Of High Throughput And Area Efficient Hard Decision Viterbi Decoder Using Verilog Hdl

Parimi Hanumantha Rao & Smt.T.Vineela, M.Tech[2]

[1]M.Tech Student , VLSI&ES, Department of Electronics and Communication Engineering, Vasireddy Venkatadri institute of Technology, Namburu, Guntur [Dist], A.P, India.

[2]Associate Professor, Department of Electronics and Communication Engineering, Vasireddy Venkatadri institute of Technology, Namburu, Guntur [Dist], A.P, India.

Abstract- *We propose a pre-computation architecture incorporated with T-algorithm for VD, which can effectively reduce the power consumption without degrading the decoding speed much. A general solution to derive the optimal pre-computation steps is also given in the paper. The Add Compare Select (ACS) unit in path metric unit is designed to reduce the latency of ACS loop delay by using Modified Carry Look Ahead Adder and Digital Comparator. We also consider the design of Survivor Memory Unit (SMU) which combines the advantages of both Register Exchange method and Trace Back method, to reduce the decoding latency and total area of the Viterbi decoder. The proposed Viterbi decoder design is described using Verilog HDL Implementation result of a VD for a rate-3/4 convolutional code used in a TCM system shows that compared with the full trellis VD, the precomputation architecture reduces the power consumption without performance loss, while the degradation in clock speed is negligible.*

***Index Terms*—ACS Unit, Viterbi decoder, VLSI**

I. INTRODUCTION

In this project we propose architecture for viterbi decoder with T-algorithm which can effectively reduce the power consumption with a negligible decrease in speed. Implementation result is for code rate ½ with constraint length 9 used for trellis coded modulation. This architecture reduces the power up to 81% without any performance loss when compared with the ideal viterbi decoder, while the degradation in the clock speed is negligible. A. Implementation In this project we propose architecture for viterbi decoder with T-algorithm which can effectively reduce the power consumption with a negligible decrease in speed. Implementation result is for code rate ½ with constraint length 3 used for trellis coded modulation. This architecture reduces the power up to 81% without any performance loss when compared with the ideal Viterbi decoder, while the degradation in the clock speed is negligible. The Fig.6 shows the proposed system.

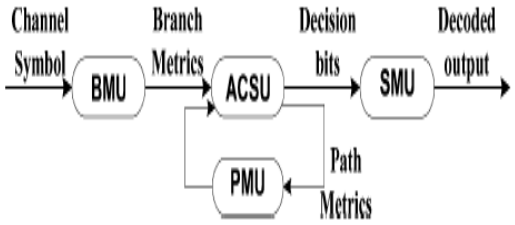

Fig. 1. Functional Diagram of Viterbi Decoder

It proposed the look-ahead technique to speed up the Add Compare Select (ACS) unit. It proposes the MSB First ACS unit with pipeline architecture to reduce the ACS loop latency. Bit level high speed Viterbi decoder is proposed. And also it proposes different design techniques to reduce the latency of ACS unit and SM unit for hard decision Viterbi decoder. Block based approach is described in to increases the throughput of ACS unit in Viterbi decoder. An efficient pre-trace back architecture to reduce the memory access and area of the survivor memory unit is presented in this paper. In this paper we proposes implementation of trace-back unit for reconfigurable Viterbi decoder to reduce the power consumption and area of survivor memory unit.

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 10
September 2017

The remainder of this paper is organized as follows. Section II gives the background information of VDs. Section III presents the precomputation architecture with T -algorithm and discusses the choice of precomputation steps. Details of a design example including the modification of survivor-path memory unit (SMU) are discussed in Section III. Simulation results are reported in Section IV and conclusions are given in Section V.

II. VITERBI DECODER

A typical functional block diagram of a Viterbi decoder is shown in Fig. 1. First, branch metrics (BMs) are calculated in the BM unit (BMU) from the received symbols. In a TCM decoder, this module is replaced by transition metrics unit (TMU), which is more complex than the BMU. Then, BMs are fed into the ACSU that recursively computes the PMs and outputs decision bits for each possible state transition. After that, the decision bits are stored in and retrieved from the SMU in order to decode the source bits along the final survivor path. The
PMs of the current iteration are stored in the PM unit (PMU).

T -algorithm requires extra computation in the ACSU loop for calculating the optimal PM and puncturing states. Therefore, a straightforward implementation of T-algorithm will dramatically reduce the decoding speed. The key point of improving the clock speed of T–algorithm is to quickly find the optimal PM.

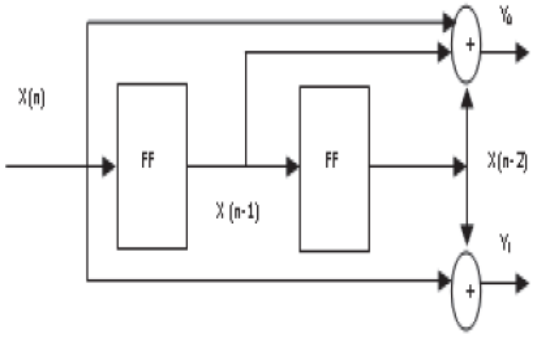

Fig.2. Rate ½ Encoder

Y1Yo- Encoder output bits,
X (n-1) X (n-2) – previous state of Encoder,
X (n)-input bit to Encoder.

Convolution Encoder

Convoluion Encoder One of greatest challenges in communication system design is efficient transmission and reception of information in the presence of errors introduced by the communication channel. The presence of errors is especially pronounced in radio communication, due to the variety of noise sources in the channel. Designers have adopted block coding methods that add redundancy in the encoding of information before transmission. Although the addition of redundant data reduces the overall throughput of the channel, forward error correction improves performance by using the redundant data to correct errors during decoding at the receiver. Convolution coding, that is, coding based on timeinvariant finite state machines, is widely used in wireless communications. This application note looks briefly at popular techniques for convolution encoding and decoding, especially Viterbi decoding. It illustrates the power of a configurable processor in handling the performance-intensive signal processing demands of coding and decoding. Specifically, user-defined instructions in the Ten Silica Instruction Extension Language (TIE) will be described which accelerate distance metric calculations, the most performance-critical task in Viterbi decoding, by more than 32 times over most popular digital signal processors and 155 times over most popular 32 bit RISC cores. Many periods as needed and each period repeats the possible transitions. The trellis diagram conventionally begins at state 00. Starting from here, the trellis expands and in L bits becomes fully "populated" such that all transitions are possible. The possible transitions then repeat from this point on.

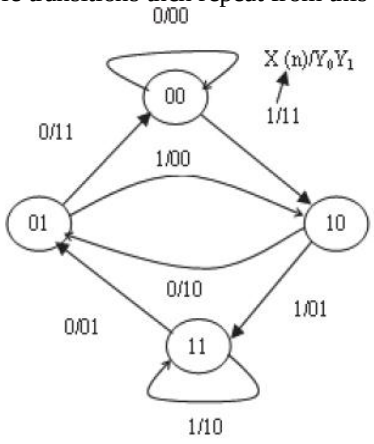

Fig.3. State Diagram

State diagram: This offers a complete description of the system. However, it shows only the instantaneous transitions. It does not illustrate how the states

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 10
September 2017

change in time. *Trellis diagram:* To include time in state transitions

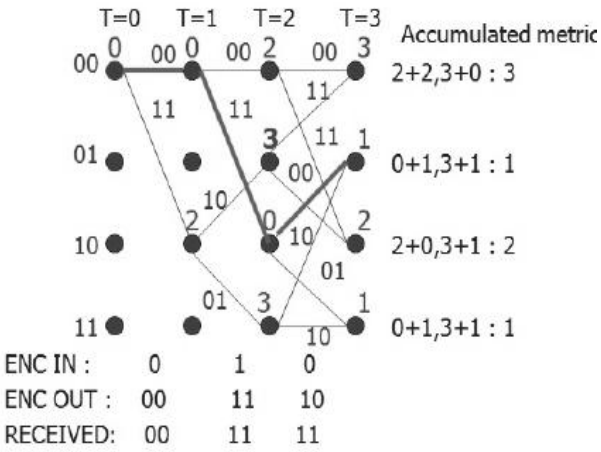

Fig.4. Trellis Diagram

III. PROPOSED VITERBI DECODER

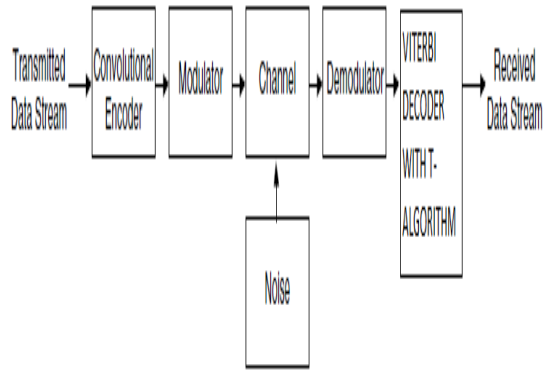

Fig.5. Block diagram of the system with T-algorithm.

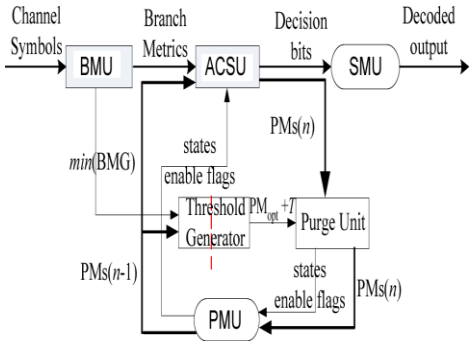

Fig.6. Architecture of Viterbi Decoder With T-Algorithm

A. Branch Metric Unit

The branch metric is a measure of the distance between what was transmitted and what was received and is defined each arc in the trellis. In hard decision decoding, where we have given a sequence of parity bits, the branch metric is the hamming distance between the expected parity bits and the received bits. As example is shown in Fig. 3.3.1.1.1 where received bits are 00. For each state transition, the number on the arc shows branch metric for its transition. Two of the branch metrics are 0, corresponding to only states and transitions where the corresponding hamming distance is 0.

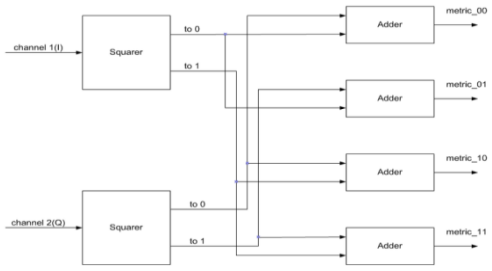

Fig.7. Block Diagram BMU

An attractive soft decision metric is the square of the difference between the received and expected. If the convolutional code produces the p parity bits, and the corresponding analog samples are $v = v_1, v_2, v_3, \ldots, v_p$, then we can construct the branch metric as (1)

$$BM[U,V] = \sum_{i=1}^{P} (Ui - Vi)2 \qquad (1)$$

Where $u = u_1, u_2, u_3, \ldots, u_p$ are expected parity bits.

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 10
September 2017

B. Path Metric Unit

Path metric unit (PMU) is comprised of Add Compare Select unit and a Memory to store the partial state metrics. The proposed Add Compare Select (ACS) unit is designed for code rate 1/2, constraint length K=3 and generator polynomials (101,111)8. The ACS unit is critical in terms of throughput and area due to the presence of feedback loop referred as the ACS loop. The ACS loop is non-linear in nature. This ACS loop recursively selects the minimum state metric and gives an output of a decision bit based upon minimum state metric. The selected minimum state metric is updated for the particular node or state in next clock cycle for the next operation. A survivor path is selected and its state metric updated for each state at each recursion.

Suppose the receiver has computed the path metric PM[*s*, *i*] for each state *s* (of which there are 2k1 , where k is the constraint length) at time step i. The value of PM[*s*, *i*] is the total number of bit errors detected when comparing the received parity bits to the most likely transmitted message, considering all messages that could have been sent by the transmitter until time step I (starting from state '00' , which we will take by convention to be the starting state always).

Add Compare Select Unit

A new value of the state metrics has to be computed at each time instant. In other words, the state metrics have to be updated every clock cycle. Because of this recursion, pipelining, a common approach to increase the throughput of the system, is not applicable. The Add-Compare-Select (ACS) unit hence is the module that consumes the most power and area In order to obtain the required precision, a resolution of 7 bits for the state metrics is essential, while 5 bits are needed for the branch metrics. Since the state metrics are always positive numbers and since only positive branch metrics are added to them, the accumulated metrics would grow indefinitely without normalization.

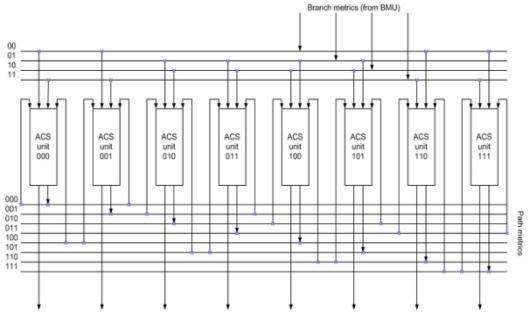

Fig.8. Implementation of PMU (Path Metric Unit)

In this project we have chosen to implement modulo normalization, which requires keeping an additional bit (8 instead of 7). The operation of the ACS unit is shown in Figure 6. The new branch metrics are added to previous state metrics to form the candidates for the new state metrics. The comparison can be done by using the subtraction of the two candidate state metrics, and the MSB of the difference points to a larger one of two.

Hamming distance between the received code word and the allowed code word is calculated by checking the corresponding bit positions of the two code words. For example hamming distance between the code words 00 and 11 is 0 or the hamming distance between the code words 00 and 11 is 2. The hamming distance metric is cumulative so that the path with the largest total metric is final winner. Thus the hard decision Viterbi decoding makes use of maximum hamming distance in order to determine the output of the decoder.

Purge Unit

When a state is purged in ACSU, the corresponding registers in SMU are not updated; thus, the power consumption is reduced. purged computation changes accordingly in the case of the T-algorithm on PMs, whereas in the case of the T–algorithm on BMs, the number of purged computation remains almost the same, regardless of the channel condition.

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 10
September 2017

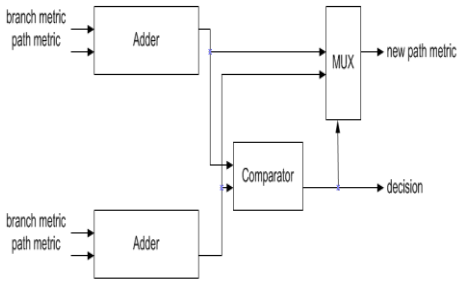

Fig.9. ACSU (Add-Compare-Select unit) Architecture

C. Survivor Memory Unit

In this section, we address an important issue regarding SMU design when T-algorithm is employed. There are two different types of SMU in the literature: register exchange (RE) and trace back (TB) schemes. In the regular VD without any low-power schemes, SMU always outputs the decoded data from a fixed state (arbitrarily selected in advance) if RE scheme is used, or traces back the survivor path from the fixed state if TB scheme is used, for low-complexity purpose.

IV. SIMULATION RESULTS

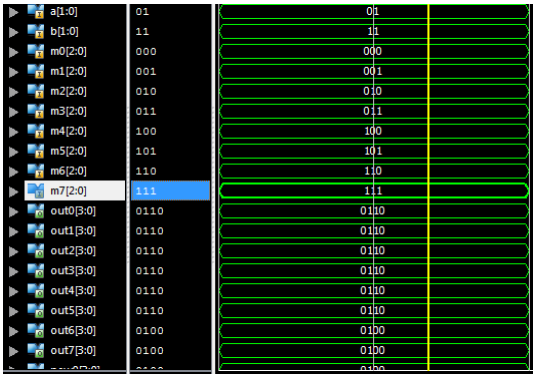

Fig.10. Simulation results of proposed Viterbi

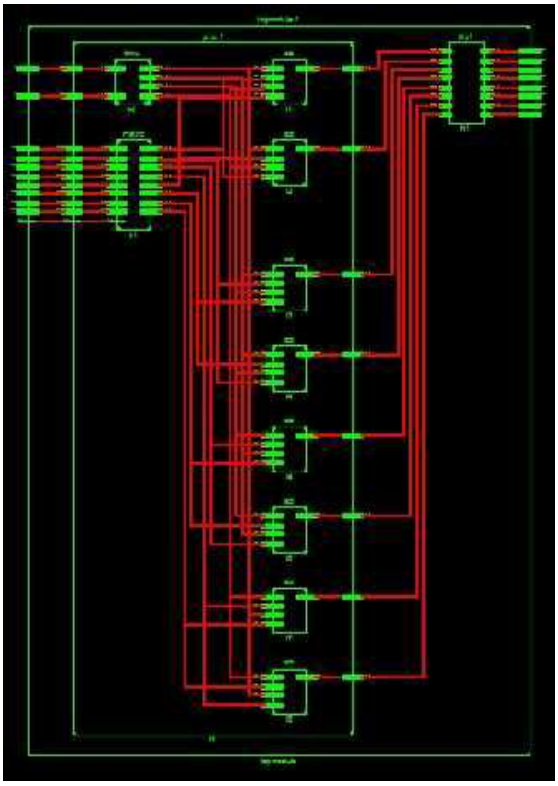

Fig.11. Enlarged Tech schematic of proposed Viterbi

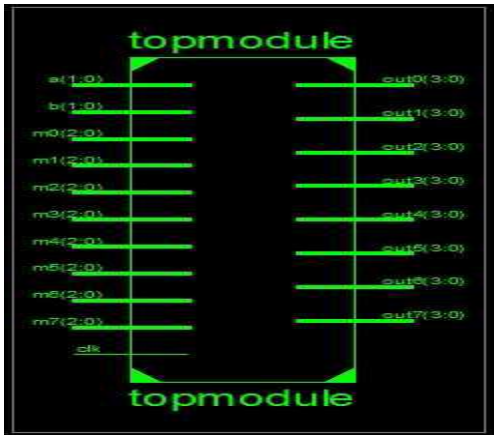

Fig.12. RTL Schematic diagram for Proposed Viterbi

V. CONCLUSION

We have proposed a high-speed low-power VD design. The pre-computation architecture that incorporates T –algorithm efficiently reduces the power consumption of VDs without reducing the

decoding speeds appreciably. This algorithm is suitable for TCM systems which always employ high-rate convolution codes. Finally, we presented a design case. Both the ACSU and SMU are modified to correctly decode the signal.

VI. FUTURE SCOPE

We can implement viterbi decoder with SPEC T-algorithm. Here, we take advantage of the SPEC-T algorithm proposed in to reduce the length of critical path. The key idea of the SPEC T-algorithm is to use an estimated optimal PM value derived from the optimal BM value, instead of searching for the optimal PM in each cycle.

REFERENCES

[1] Vasily P. Pribylov, Alexander I. Plyasunov (2005). "A Convolutional Code Decoder Design Using Viterbi Algorithm with Register Exchange History Unit". SIBCON. IEEE.
[2] John G. Proakis (2001). "Digital Communication". McGraw Hill, Singapore. S. K. Hasnain, Azam Beg and S. M. Ghazanfar Monir (2004).
[3] "PerformanceAnalysis of Viterbi Decoder Using a DSP Technique". INMIC IEEE.
[4] Irwin M. Jacobs (1974). "Practical Applications of Coding". IEEE. pp 305-310.
[5] YOU Yu-xin, WANG Jin-xiang, LAI Fengchang and YE Yi-zheng (2002). "VLSI Design and Implementation of High Speed Viterbi Decoder". IEEE .pp 64-66.
[6] Rodger E. Ziemer, Roger L. Peterson: Introduction to digital communication; Prentice- Hall International (UK), cop. 2001
[7] Samirkumar Ranpara: On a Viterbi Decoder design for low power dissipation; Virginia Polytechnic Institute and State University; April, 1999; IJCSIET--International Journal of Computer Science information and Engg., Technologies ISSN 2277-4408 || 01102015-001 IJCSIET-ISSUE5-VOLUME3-SERIES-2 9
[8] A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimumdecoding algorithm", *IEEE Transactions on Information Theory*, Vol. 13, Issue 2,pp 260-269, April 1967.
[9] J. A. Heller and I. M. Jacobs. "Viterbi Decoding for Satellite and Space Communication**",***IEEE Transactions on Communcation Technology*, Vol. pp. 835-848,October 1971.
[10] S. Lin and D. J. Costello, "Error Control Coding", *Second Edition. Prentice-Hall,Inc.*, 2004.
[11] G. Forney, "The Viterbi Algorithm", *Proceedings of the IEEE*, pp. 268-278, March1973.
[12] J.K. Omura, "On the Viterbi decoding algorithm", *IEEE Transactions on Information Theory*, pp.177-179, 1969.
[14] C. B. Shung, H-D. Ling, R. Cypher, P. H. Siegel, and H. K. Thapar, "Area-efficient architectures for the Viterbi algorithm Part I: Theory," *IEEE Transactions on Communications*, Vol. 41, pp. 636-644, April 1993.
[15] C. B. Shung, H-D. Ling, R. Cypher, P. H. Siegel, and H. K. Thapar, "Area efficient architectures for the viterbi algorithm Part II: Applications," *IEEETransactions on Communications*, Vol. 41, no 5, pp. 802 807, May. 1993.
[16] T. Jarvinen, P. Salmela, T. Sipila, J. Takala,; "Systematic approach for path metricaccess in Viterbi decoders" *IEEE Transactions on Communications*, Vol. 53, Issue 5, pp. 755 – 759, May 2005
[17] H.L. Lou, "Implementing the Viterbi Algorithm", *IEEE Signal Processing Magazine*, pp. 42-52, 1995.
[18] P. Hekstra, "An Alternative to Metric Rescaling in Viterbi Decoders", *IEEETransactions on Communication*, Vol. 37, no. 11, pp. 1220-1222, Nov. 1989.
[19] J. J. Kong and K.K. Parhi, "Low-latency architectures for high-throughput rate Viterbi decoders" , *IEEE Transactions on Very Large Scale Integration (VLSI)Systems*; Vol. 12, Issue 6, pp. 642 – 651, June 2004.
[20] R.Cypher and C.B.Shung, "Generalized Trace-back Techniques for Survivor Memory Management in the Viterbi Algorithm", *Journal of VLSI Signal Processing*, Vol.5, pp.85-94, 1993.