Adaptive Viterbi Decoder for Space Communication Application

R.V.Babar[1], Dr. M.S. Gaikwad[2], Pratik L. Parsewar[3]

[1]Assistant Professor, Sinhgad Institute of Technology, Department of Electronics and Telecommunication, Lonavala, Pune

[2]Principal, Sinhgad Institute of Technology, Lonavala, Pune

[3]Research Scholar (M.E), Sinhgad Institute of Technology, Department of Electronics and Telecommunication, Lonavala, Pune

E-mail- pratik.parsewar@gmail.com

Abstract— Day by day need of increase in data transmission rate in wireless communication systems increases rapidly. Viterbi Algorithm is known as optimum-decoding algorithm for convolutional codes and has often been served as a standard technique in digital communication systems for maximum likelihood sequence estimation. In this paper, by making existing well-known Viterbi algorithm, an adaptive Viterbi algorithm that is based on strongly connected trellis decoding is proposed. Using this algorithm, the design and a field-programmable gate array implementation of a low-power adaptive Viterbi decoder with a constraint length 9 and a code rate of 1/2 is presented. It is shown that the proposed algorithm can reduce by up to 70% the average number of ACS computations over that by using the non-adaptive Viterbi algorithm, without degradation in the error performance. The proposed Adaptive Viterbi decoder can be used in high speed decoding applications such as space communication. This results in lowering the switching activities of the logic cells, with a consequent reduction in the dynamic power. Also in this paper, with the help of Matlab, comparison of results of BER of adaptive Viterbi decoder and Viterbi decoder has been concluded.

Keywords - Convolution codes, Adaptive Viterbi decoder, ACS unit, field-programmable gate array (FPGA) implementation.

I. INTRODUCTION

CONVOLUTIONAL codes and the Viterbi algorithm are known to provide a strong forward error correction (FEC) scheme, which has been widely, utilized in digital communication applications. As the error-correcting capability of convolutional codes is improved by employing codes with larger constraint lengths K, the complexity of decoders is increased. The Viterbi algorithm (VA), which is the most extensively employed decoding algorithm for convolutional codes, is effective in achieving noise tolerance, but the cost is an exponential growth in memory, computational resources, and power consumption. To overcome this problem, the reduced-complexity adaptive Viterbi algorithm (AVA), has been developed. The average number of computations per decoded bit for this algorithm is substantially reduced versus the VA, while comparable bit-error rates (BER) are preserved.

It has been shown that the larger the constraint length used in a convolutional encoding process, the more powerful the code produced. However, the complexity of the Viterbi decoding process becomes increases for a constraint length is more than 9 sizes. As an effect, it would not possible to achieve a hardware implementation of a Viterbi decoder for, in order to meet the requirements of the power, speed and area. In recent years, Viterbi decoders have been mostly used in mobile systems that require portable battery operations, thus making the power consumption a critical concern to the designers.

II. Background

The idea behind the Viterbi Decoder (VD) is quite simple, in spite of its inherent implementation difficulty. Moreover, there is a wide gap in complexity with the transmission side, where convolutional encoding can easily be implemented. Since convolutional codes are represented by a state trellis, the decoder is a finite state machine that explores the transitions between states, stores them in a large memory, and comes to a final decision on a sequence of transitions after some latency due to the constraint length of the input code. Decisions are usually taken by considering the transition metrics among states, which are updated in terms of either Euclidean or Hamming distance with the error-corrupted received sequence. The performance of convolutional codes strongly depends on their minimum distance, which in turn depends on the constraint length and coding rate. As a consequence, in order to increase the gain with respect to the uncoded case, there is a continuous trend towards increasing such parameters. Thus, complexity may grow up to a limit where classic implementation techniques are no longer viable. Recently, Adaptive Viterbi Decoding (AVD) for the algorithmic part and systolic architectures for the implementation aspects are increasing their popularity in the technical literature. In the AVD approach, only a subset of the states is stored and processed, significantly reducing computation and storage resources at the expense of a small performance loss.

III. Viterbi Algorithm

The Viterbi algorithm proposed by A.J. Viterbi is known as a maximum likelihood decoding algorithm for Convolutional codes. So, it finds a branch in the code Trellis most likely corresponds to the transmitted one. The Algorithm is based on calculating the Hamming distance for every branch and the path that is most likely through the trellis will maximize that metric. The algorithm reduces the complexity by eliminating the least likely path at each transmission stage. The path with the best metric is known as the survivor, while the other entering paths are non-survivors. If the best metric is shared by two or more paths, the survivor is selected from among the best paths at random.

The selection of survivors lies at the heart of the Viterbi Algorithm and ensures that the algorithm terminates with the maximum likelihood path. The algorithm terminates when all of the nodes in the trellis have been labeled and their entering survivors are determined. We then go to the last node in the trellis and trace back through the trellis. At any given node, we can only continue backward on a path that survived upon entry into that node. Since each node has only one entering survivor, our trace-back operation always yields a unique path. This path is the maximum likelihood estimate that predicts the most likely transmitted sequence.

Various coding schemes are used in wireless packet data network of different standards like GPRS, EDGE and WiMAX to maximize the channel capacity.

IV. Architecture of Viterbi Decoder

The architecture of the Viterbi decoder is illustrated in Fig. 1.

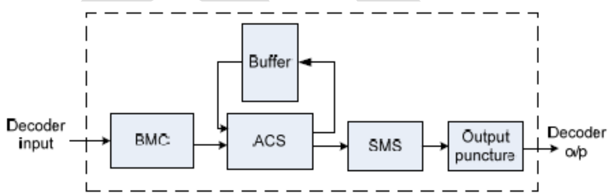

Fig. 1. Basic building blocks of the Viterbi decoder.

A. The Branch Metric Computer (BMC)

This is typically based on a look-up table containing the various bit metrics. The computer looks up the n-bit metrics associated with each branch and sums them to obtain the branch metric. The result is passed along to the path metric update and storage unit. The dashed rectangle in Fig. 2 shows the BMC.

B. The Path Metric Updating and Storage

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. Add-Compare-Select (ACS)

ACS is being used repeatedly in the decoder. A separate ACS circuit can be 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 L×2M. In our implementation we combined both the BMC and the ACS in one unit representing a single wing of each trellis butterfly as illustrated in Fig. 2.

D. Survivor Memory Management (SMM)

This is responsible for keeping track of the information path metric updating and storage unit. Bits associated with the surviving paths designated by the path metric updating and storage unit.

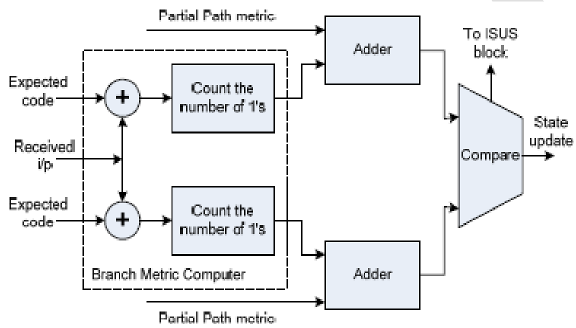

Fig. 2. ACS module.

V. Adaptive Viterbi Decoder

The well-known VA has been described in literature extensively. The data path of the Viterbi Decoder is composed of three major components: Branch Metric Calculation Unit (BMU), ACS and Survivor Memory Unit (SMU) as shown in Fig 1. The branch metrics are calculated from the received channel symbols in BMU and then fed into the ACS which performs Add-Compare-Select for all the states. The decision bits generated in ACS are stored and retrieved in the SMU in order to finally decode the source bits along the final survivor path. The state metrics of the current iteration are stored into the Path Metric Memory Unit (PMU) and read out for the use of the next iteration.

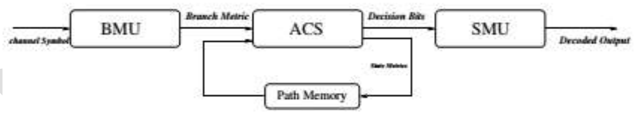

Fig. 3 Top Level Diagram of Viterbi Decoder

In ACS unit, the VA examines all possible paths in the Trellis graph and determines the most likely one. The AVA only keeps a number of the most likely states instead of the Whole of 2K−1 states, where K is the constraint length of the convolution encoder. The rest of the states are all discarded. The selection is based on the likelihood or metric value of the paths, which for a hard decision decoder is the Hamming distance and for a soft decision decoder is Euclidean distance. The rules of the selecting the survivor Path is:

1. Every surviving path at trellis level n is extended and Its successors at level n+1 are kept if their path metric are smaller or equal to P_{Mmin} n + T, where P_{Mmin} n is the minimum path metric of the surviving path at stage n+1, and T is the discarding threshold configured by the user.

2. The total number of survivor paths per trellis stage is up bounded to a fixed number: N_{max}, which is preset prior to the start of the communication. In order to illustrate how the AVA operates, an example using a code rate R = 1/2, constraint length K = 3 is given in Fig 2. The threshold T is set to 1 and Nmax is set to 3 respectively. Initially at t = 0, we set the P_{Mmin} n Equal to 0 and the decoder states equal to 00. The received Sequence is {01, 10, 11, 01, and 00}. The symbol X represents the discarded path and bold line represents the final decision path by the AVA algorithm. For the sake of the simplicity, the minimum path metric of the nth iteration P_{Mmin} n is denoted by dm. It can be seen that at each trellis stage, the number of the survivor states is smaller than the VA (2K−1) and gets the same decision paths as the VA. The optimal selection strategy for architecture parameter Nmax and T is discussed. In this paper, a range of T from 20 to 30 and a range of Nmax up to 2K−2 are considered. The top level block diagram of ACS unit of AVA Decoder is shown in Fig 4. Path Metric Adder and State Merge Unit correspond to the operation of Add and Compare-Select Operation in VA respectively. Compared to conventional VA, two additional processing units are inserted into the data path of the VA: Threshold Selection and Survivor Contender which correspond to the AVA rule 1 and rule 2 respectively in AVA architecture. The Threshold Selection unit discards the paths exceeding the sum of the preset value T and the minimum path metric of the last iteration P_{Mmin} and the survivor contender is responsible for sifting N_{max} states out of 2Nmax states. In addition, Min Path Calculation unit is responsible for calculating the minimum path of the current iteration.

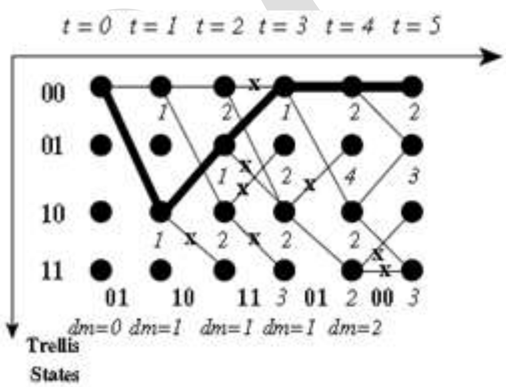

Fig. 4 Trellis Graph of adaptive Viterbi decoding

Fig. 5 block Diagram of Adaptive ACS Architecture

The conventional Threshold Selection architecture is Shown in Fig 5. At time step n, the path metric of state i denoted as PMin and the branch metric from BMU, denoted as BMij associated with a state transition from i to j are added in the path metric adder. The accumulated path metric of state j, denoted as PMj n+1 is compared to the sum of P_{Mmin}^{n} and pre-set constant T. Those exceeding will be discarded. In parallel to the operation of the Threshold Selection Unit and Survivor State Contender, the path metric of state j, PMj n+1 is fed into the Min Path Calculation for determining the minimum path metric of current iteration PMminn+1 ,which is stored for the use of next iteration.

VI. RESULTS

In order to evaluate the performance, the ACS unit as shown in Fig 3 is implemented with both conventional and reformulated scheme in Verilog models and mapped into standard cell based ASIC and LUT based FPGA technologies respectively. Here we do not take BMU and SMU into the consideration because the two components are the same in the different approaches. The specifications of the implementations are:

• 64 states, constraint length K = 9

• Code rates R = 1/2

• 3 bits, 8 level soft decision inputs

• Nmax = 16, T = 20

• ASIC Approach: UMC .18u stand cell library

• FPGA Approach: Xilinx Virtex600E

Improvement is achieved in speed respectively. Significant improvement can be observed both in standard cell approach and LUT approach.

In addition, the power efficiency is enhanced compared to the basic comparison unit results. Further reduction in power can be contributed to the reduction of complexity in Min Path Calculation unit and Path Metric Adder.

Below, BER (bit error rate) calculated with the help of MATLAB. Comparison graph shown below represent BER rate has been improved in Adaptive Viterbi decoder (fig 6):

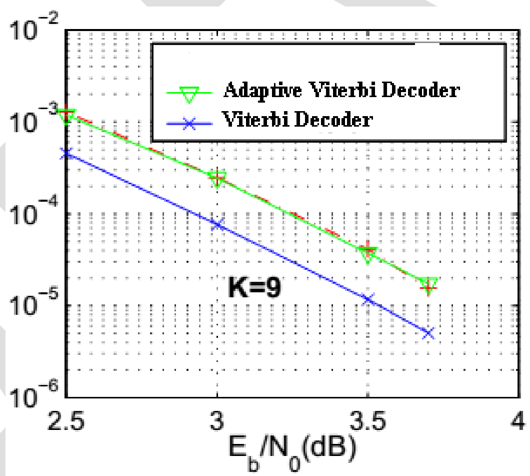

Fig. 6 BER rate of Adaptive Viterbi and Viterbi Decoder for K=9

VII. CONCLUSION AND FUTURE WORK

In this work, a high speed implementation of an adaptive Viterbi decoder which uses modified T-algorithm is presented .The use of error-correcting codes has proven to be an effective way to overcome data corruption in digital communication channels. Some of the conclusions drawn from the design are as under below. Efficient reformulation based architecture for Threshold Selection in Adaptive Viterbi Decoding is presented. The Reformulated architecture exploits the inherent parallism Between the Add Compare Select Operation and rescales operation in Adaptive Viterbi

Decoding. Through reformulation, the hardware complexity for the threshold selection in Adaptive Viterbi Decoding is significantly reduced both in ASIC and FPGA technologies, which leads to a corresponding significant reduction in area, power and delay. It should be noted that the proposed technique will also achieve a similar power, area and speed efficiency with different specifications e.g. K = 9.

Power and area has been reduced by dividing the Trellis Coding structure into two segments. Significant amount of power has been reduced in the design by modifying branch metric architecture.

In the future, plan to consider the decoding benefits of using a hybrid microprocessor and FPGA device. The tight integration of sequential control with parallel decoding may provide further run-time power benefits.

REFERENCES:

[1] M. Cummings and S. Huruyama, "FPGA in the Software Radio," *IEEE Comm Magazine*, volume: 37, no. 2, pp. 108-112, February 1999.

[2] Xilinx Inc., Virtex-6 SXT for DSP and memory-intensive applications with low-power serial connectivity, http://www.xilinx.com/products/v6s6.htm. Last visited: April. 2009.

[3] A.J. Viterbi, "Convolutional codes and their performance in Communication systems," *IEEE Trans. Commun*., vol. COM-19, pp. 751-772, Oct., 1971.

[4] B. Sklar, "Digital Communication Fundamentals and Applications," Prentice Hall, Englenwood Cliffs, New Jersey, 2000, Part 2 chapter 7.

[5] GSM 05:03: "Channel coding", *Version 8.9.0* Release 1999.

[6] GSM 03.64: "Overall description of the GPRS radio interface; Stage 2" *Version 8.12.0*. Release 1999.

[7] *IEEE Std 802.16e-2005*: "Part 16: Air interface for Fixed and Mobile Broadband Wireless Access Systems".

[8] Sherif Welsen Shaker, Salwa Hussien Elramly, Khaled Ali Shehata, "FPGA Implementation of a Configurable Viterbi Decoder for Software Radio Receiver", IEEE 44th annual Systems Readiness Technology Conference, AUTOTESTCON 2009, Anaheim, California, USA, 14-17 September 2009.

[9] Maunu, J.; Laiho, M.; Paasio, A., "Performance analysis of BMC and Decision Units in the differential analog Viterbi decoder", *IEEE 25th Norchip Conference*, Aalborg, Denmark, Volume, Issue, 19-20 Nov. 2007 Page(s):1 – 4.

[10] Kamuf, M. Owall, V. Anderson, J.B., "Survivor Path Processing in Viterbi Decoders Using Register Exchange and Trace forward", *IEEE Trans on Circuits and Systems II*: Express Briefs, vol. 54, Issue 6, pp. 537-541, June 2007.

[11]Pravin Bhagwat, Charschik Bisdikian, Ibrahim Korpeoglu, Arvind Krishna, and Mahmoud Naghshineh, "System design issues for low-power, low-cost short range wireless networking," in IEEE International Conference on Personal Wireless Communications (ICPWC'99), February 1999.

[12] Cheetham, B., 2010. Power saving convolutional decoder. University of Manchester. [Email] (Personal Communication, 3 March 2010)

[13] Samir Palnitkar, "Verilog HDL – A Guide to Design and Synthesis", 2nd edition, Prentice Hall, ISBN: 0-13-044911-5 2003.