

Implementation of reduced memory Viterbi Decoder using Verilog HDL

Kumar Pari¹, B. Raghavaiah²

¹ PG Student (M. Tech), Dept. of ECE, Chirala Engineering College, Chirala, A.P, India.

² Associate Professor, Dept. of ECE, Chirala Engineering College, Chirala, A.P, India.

Abstract: The best way of decoding against random errors is to compute the received sequence with every possible code sequence. This is called maximum likelihood (ML) decoding. The criterion for deciding between two paths is to select the one having the smaller metric. The rule maximizes the probability of a correct decision. The Andrew Viterbi proposed an efficient algorithm to find the minimum distance to received sequence in a trellis. It's named after him as Viterbi algorithm (VA). This was recognized by Forney to be Maximum Likelihood decoder. The Viterbi algorithm occupies large memory and computational resources. To address this problem Proposed Viterbi Algorithm is introduced. The Proposed Viterbi decoder functionally is same as the previous Viterbi decoder but it reduces memory and the hardware resources. The proposed block diagram checks every node for path metric value and eliminates the path that is found if it is not having minimum distance.

I. Introduction

Convolutional encoder with Viterbi decoder acts as powerful method for Forward Error Correction (FEC). Viterbi Algorithm is an optimum decoding Algorithm for convolution code if is applied to transmission in additive white Gaussian noise channel. The Viterbi algorithm (VA) occupies large memory, computational resources. So to address this issue reduced memory Viterbi algorithm (RMVA) is introduced.

Error correction

The rise of probability of error in the transmission due to presence of noise leads to requirement error and detection.

Error control Strategies

The two major type of error control strategies. They are Forward Error Control (FEC) and Automatic Repeat (ARQ)

Forward error correction (FEC)

Forward Error Control is one of the most widely used in error control. Forward Error correction block in the receiver can correct a transmission error without asking the sender for more information or for a retransmission. It is done by Error correction code (ECC).

Automatic repeat request Method (ARQ)

In Automatic repeat request method, after the error detection the receiver sends the request to the transmitter to retransmit the data.

Types of EEC

An error-correcting code is an algorithm for expressing a sequence of numbers such that any errors which are introduced can be detected and corrected (within certain limitations) based on the remaining numbers. There are basically two mechanisms for adding redundancy to error control coding techniques. They two basic mechanisms are block coding and convolution coding. This classification is based on the presence or absence of memory in the encoders for these two codes. An encoder for a block code is memory less and for a convolution encoder is with memory. Tree diagram for types of EEC is shown in Figure 1. Error-correcting codes are also used in DVD players, high speed modems and Multimedia phones etc.

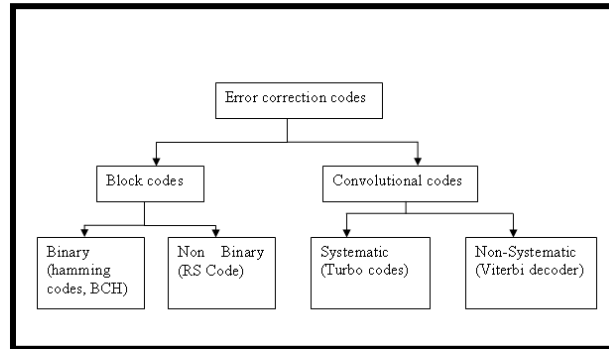


Figure 1 Tree diagram for type of ECC

In this paper, convolution coding with viterbi decoder as FEC decoder is considered for analysis.

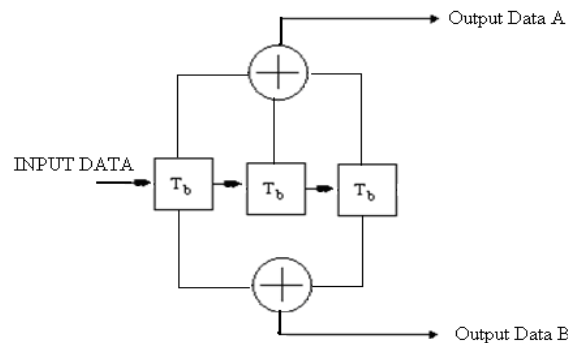


Figure 2 Block diagram Convolutional Encoder (k = 3)

Block diagram of Convolution encoder for $k=3$ is shown in Figure 1.4. A convolutional encoder with constraint length k consists of a k -stage shift register. Information symbols are shifted in at the left and the two modulo-2 adders yield two coded symbols which form a single code word. Generator polynomials define which are flip to be considered for which output. The connections between the shift register and the modulo-2 adder can be represented by the coefficient of polynomials. Polynomials for 802.11a standard are $G_0=1718$, $G_1=1338$ for $k=3$ at code rate of $1/2$.

The function of the FEC decoder is to attempt to reconstruct the input sequence transmitted by the encoder by evaluating the received channel output. Values received at the decoder may differ from values sent by the encoder due to channel noise. The interaction between states represented by the trellis diagram is used by a decoder to determine the likely transmitted data sequence.

Performance of convolution codes are affected by various factors, the most important one that has a critical impact on performance are decoding algorithm and distance properties of the code.

II. History of Viterbi decoder

The best way of decoding against random errors is to compute the received sequence with every possible code sequence. This is called maximum likelihood (ML) decoding. The criterion for deciding between two paths is to select the one having the smaller metric. The rule maximizes the probability of a correct decision. In 1967 Andrew Viterbi proposed an efficient algorithm to find the minimum distance to received sequence in a trellis. It's named after him as Viterbi algorithm (VA). This was recognized by Forney to be Maximum Likelihood decoder in 1973. It was the first proposed for decoding convolution codes. The convolutional encode with Viterbi decoding is a powerful method for forward error correction.

The Viterbi algorithm operates on a state machine assumption. That is, at any time the system being modelled is in some state. There are a finite number of states, however large, that can be listed. Each state is represented as a node. Multiple sequences of states (paths) can lead to a given state, but one is the most likely path to that state, called the "survivor path". This is a fundamental assumption of the algorithm because the algorithm will examine all possible paths leading to a state and only keep the one most likely. This way the algorithm does not have to keep track of all possible paths, only one per state. A second key assumption is that a transition from a previous state to a new state is marked by an incremental metric, usually a number. This transition is computed from the event. The third key assumption is that the events are cumulative over a path in some sense, usually additive. So the crux of the algorithm is to keep a number for each state. When an event occurs, the algorithm examines moving forward to a new set of states by combining the metric of a possible

previous state with the incremental metric of the transition due to the event and chooses the best. The incremental metric associated with an event depends on the transition possibility from the old state to the new state.

It has widely deployed in many wireless communication systems to improve the limited capacity of communication channel. The complexity of Viterbi algorithm (VA) is proportional to number of states in the decoding trellis, where the number of state $2k$ and k is total number of encoder memory bit used in the encoder for the convolution code. If k larger, will leads to gain high correction capability and large circuit, power and low speed decoding.

Viterbi decoder

The receiver can deliver either hard or soft symbols to the Viterbi decoder. A hard symbol is equivalent to a binary ± 1 . A soft symbol, on the other hand, is multi-leveled to represent the confidence in the bit being positive or negative. For instance, if the channel is non-fading and Gaussian, the output of the matched filter quantified to a given number of bits is a suitable soft input. In both cases, 0 is used to represent a punctured bit. In case of hard decision demodulation, data is demodulated into either 1s or 0s, or quantized into two levels only.

The process described above makes a hard binary decision about each incoming bit and then uses only the Hamming distances. This simplifiers the hardware, but does not result in optimal performance.

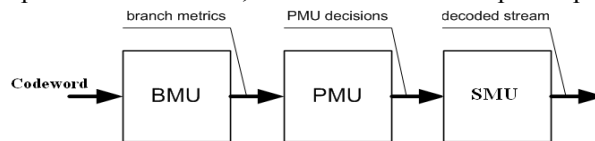


Figure 3 Block diagram of Viterbi decoder

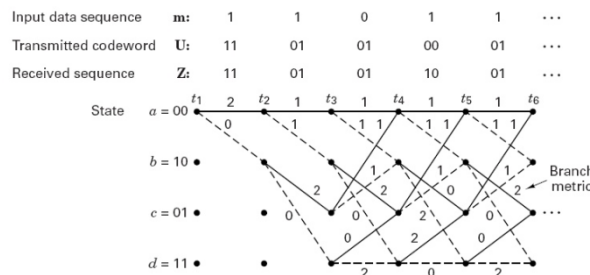


Figure 4 Trellis diagram of Viterbi decoder

The Working of Viterbi decoder in term of block diagram and trellis diagram is show in Figure 3 and 4. For hard decision decoding, the Viterbi algorithm uses the hamming distance to find the branch metric and path metric. Codeword is given to branch metric unit. Branch metric unit's function is to calculate branch metrics, which are Hamming distances between every possible symbol in the codeword and the received symbol. Path metric unit summarizes branch metrics to get metrics for $2K - 1$ path, one of which can eventually be chosen as optimal. Survivor memory unit can be trace-back process or register exchange method, where the survivor path and the output data are identified. The error probabilities achieved by Viterbi algorithm depends on the code, the rate of the code, its free distance, channel SNR and demodulation Quantized output

The quality of Viterbi decoder design is mainly measured by three criteria

- Coding gain
- Throughput
- Power dissipation

Viterbi Algorithm

The VA examines all possible paths in trellis graph and determines the most likely one. The AVA only keeps a number of the most likely state instead of the whole of $2k-1$ state, where is constraint length of convolution encoder. The rest of the other states are all discarded. The selection is based on the likelihood or metric value of the path, which for hard decision is the hamming distance and a soft decision decoder is Euclidean distance

Every Surviving path at trellis level $L - 1$ is extended and its successors at level l are kept if their path metric is smaller or equal to $d_m + T$, where d_m is the minimum path metric of the surviving path at stage $L-1$, T is discarding threshold configure by the designer.

The total number of survivor path per trellis stage is up bounded to fixed number, which is pre-set prior to the start of the communication. Generally this will be same as number of states in decoder.

Viterbi Decoder (VD)

Figure 5 shows the data flow diagram of an Viterbi algorithm, which adds two functional blocks, including the best winner search and non survivor purge, into the original Viterbi algorithm.

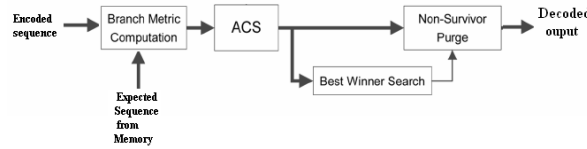


Figure 5 Block diagram of Viterbi decoder

Codeword is applied to branch metric computation unit. It calculates branch metric by comparing with expected symbol. ACS updates path metric by cumulative accumulation of branch metric. Best winner search determines final winner and give it non survivor purge unit. It deletes all paths expect winner.

Branch metric computation unit

The first unit is called branch metric unit BMU is the simplest block in the Viterbi decoder design. Here the received data symbols are compared to the ideal outputs of the encoder from the transmitter and branch metric is calculated. Hamming distance or the Euclidean distance is used for branch metric computation.

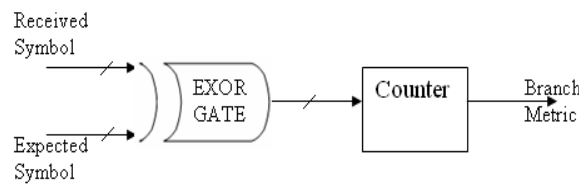


Figure 6 Block diagram of Branch Metric Unit

Block diagram of BMU (Branch metric unit) is shown in Figure 6. The BMU calculates the branch metrics from the input data. For hard decision BMU calculate every thing in term of hamming distance. Hamming distance between the received Codeword and the expected is calculated by compares the received code symbol with the expected code symbol and counting the number of different bits.

BMC (branch metric computation) unit to calculate the branch metrics which are then moved to the ACS (add compare select) unit.

Add Compare Select (ACS) Unit

The major task of the ACS is to calculate the metrics and selected paths. The add-compare-select (ACS) unit recursively accumulates the branch metrics to path metrics for all the incoming paths of each state and selects the path with minimum path metric as the survivor path. An ACS module is shown in Figure 7. The two adders compute the partial path metric of each branch, the comparator compares the two partial metrics, and the selector selects an appropriate branch.

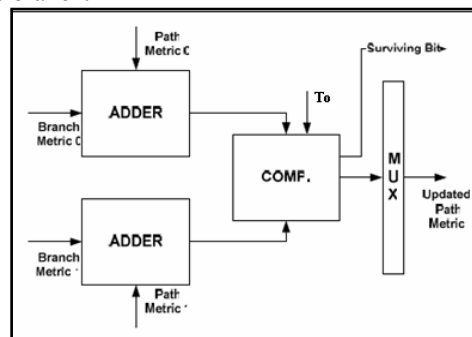


Figure 7 Add-Compare-Select Unit

ACS units determine their own local winners, the best winner search block finds the one having the best (minimum) path metric among all the winners, and the non survivor purge block deletes the local winners

Survivor Unit

Survivor Unit of Viterbi decoder based on either an RE (register exchange) or TB (trace-back) design style

The TB method takes up less area but requires much more time as compared to RE method because it needs to search or trace the survivor path back sequentially. The trace back approach is generally a lower power alternative to the register exchange method

The Register Exchange (RE) method is the simplest conceptually and a commonly used technique. In this technique, the Trellis Window is constructed of a bank of registers connected in the same manner as the trellis diagram. The data path in Register Exchange is shown in Figure 8. For instance, at time slot T1 the survivor branch for state 1 is from state 0 at T0; therefore, the initial content of the state 0 register, which is a '0', is shifted into state 1 register at T1 and the corresponding decoded data for the survivor branch, which is a '1', is appended to it. In this method a register assigned to each state contains information bits for the survivor path from the initial state to the current state. Bold arrow indicate Global winner path.

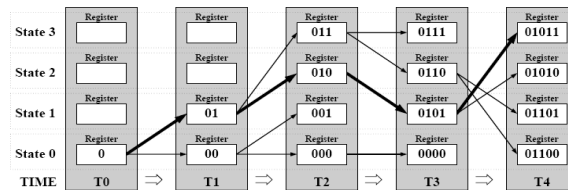


Figure 8 Block diagram of Register Exchange method

This method eliminates the need to trace back since the register of the final state contains the decoded output sequence. However, this method results in complex hardware due to the need to copy the contents of all the registers in a stage to the next stage.

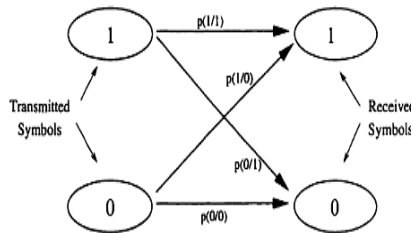


Figure 9 Binary memory less channel

III. Design and Implementation of VD

According to Viterbi there are 3 assumptions. First any system can be modelled in some state. There are a finite number of states, however large, that can be listed. Each state is represented as a node. Second a transition from a previous state to a new state is marked by an incremental metric. Third moving forward to a new set of states by combining the metric of a possible previous state with the incremental metric of the transition due to the event and chooses the best. The incremental metric associated with an event depends on the transition possibility from the old state to the new state.

First Transition table is prepared from convolution encoder for the specification and stored in memory. This transition table describes about next possible outcome and there encoded value for that particular transition for given input state. Since our encoding rate is 1/2 every state while have two possible outcomes and similar every state while have two incoming branch. It's works like a controller for whole trills. Transition table k=3 at rate 1/2 displayed for example in table 1.

Table 1 Transition table k=3 at rate 1/2

state	Next possible Outcome	Encoded value for Outcome	Next possible Outcome	Encoded value for Outcome
	1	1	2	2
0	0	0	10	10
1	0	11	10	1
10	1	11	11	1
11	1	0	11	10

Branch metrics unite used to compare encoded inputs applied to Viterbi decoder and encoded values for transition from one stage of trills to next stage based on this branch weight is calculated. It feed as input Add compare and select unit.

Add Compare and select (ACS) works on every state of stage on trills. Two incoming branch of state is decided based on transition table. ACS unit while update path metric for incoming branch and eliminate path with higher path metric. ACS and BMU are clubbed together to form a butterfly module. So for 64 stages there will be 64 butterfly modules.

Best winner search unit works on stage of trills. Work of ACS unite is to find local winner on every node (state) of trills. Work of Best winner search unit is find final winner at end of operation.

Finally trills is formed by integrate all above module together. In trills at first stage path metric set to non-zero value for all states expect stage 0(path metric is set to zero). Similar if ACS unit want to delete path then path

metric of that state is set to maximum value. By doing this path get automatic eliminated at the next stage of trills.

In order to keep sufficient error correction capability, the length of the path (trace back length) should be $5(K - 1)$. In this design value of k is 7 so trace back length should be thirty.

Decoded can be done either by Trace Back method (using memory) or by Register Exchange method. Each method have there own advantages and disadvantage. Trace Back unit start working once best winner give as final winner. The current best state is used to predict the previous state by referencing the corresponding value of the first column of the Trellis Window. This process is repeated for each computed state and the corresponding column till the end of the table. So it doubles the amount of time taken for decoded data.

The RE technique is a straightforward technique for managing the decision vectors. In this technique, the Trellis Window is constructed of a bank of registers connected in the same manner as the trellis diagram. The newest decision hits are inserted in the left column of the Trellis Window as the oldest bits out at the right of the window. Decoding is done at end of trills once best winner search chosen final winner. Its increases number of registers in design result increase in area and power.

In a RE architecture the trellis window is the main contributor for power consumption. This is because the data is being shifted through the complete table which results in a huge transition activity. Size of register in Register Exchange based decoder depends on two factors one stage of Trilles and second Length of trace back.

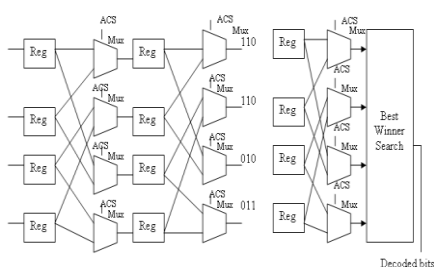


Figure 10 R.E based Viterbi architecture

To reduce power and size of decoder it designed as shown in Figure 10 first local winner is chosen and then survivor bit of local winner from ACS unit is given to register for storing.

IV. Results and Conclusion

The Convolutional encoder for the constraint length of $K=3$ and code rate of $r=1/2$ has been developed and the synthesis is carried out. It has been simulated and the simulation result is shown in fig. 11. The Viterbi decoder has been developed using and the synthesis is carried out.

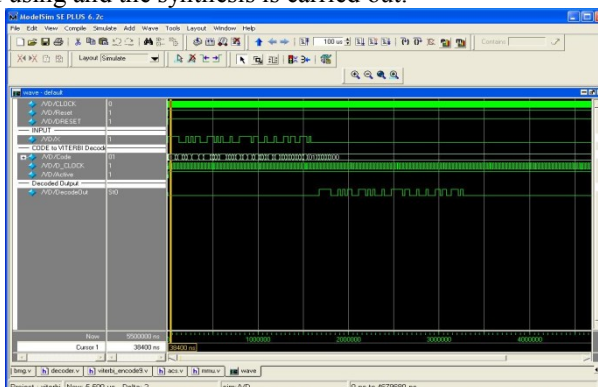


Figure 11 Simulation results

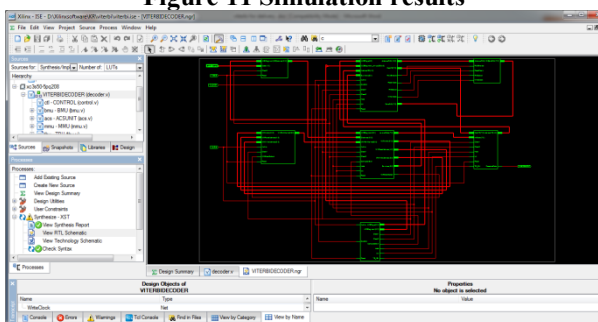


Figure 12 RTL Schematic Diagram

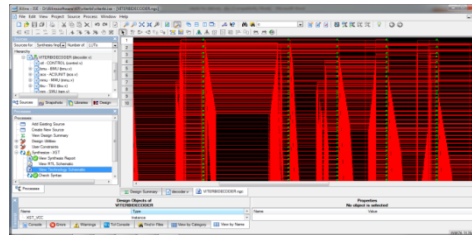


Figure 13 Technology Schematic

In this paper, a reduced memory implementation of a Viterbi decoder is presented. The use of error-correcting codes has proven to be an effective way to overcome data corruption in digital communication channels. The Register exchange method and the VA incorporated in design resulted in a high throughput in terms of memory and hardware resources. In this paper the Xilinx ISE EDA Tool is used for synthesis and Modelsim is used for simulation.

Table 2 Hardware resources and memory Summary

Logic Utilization	Used	Available
Number of Slices	3394	4656
Number of Slice Flip Flops	4225	9312
Number of 4 input LUTs	4819	9312
Number of bonded IOBs	6	232
Number of GCLKs	4	24
Total memory usage	259996 kilobytes	

Acknowledgements

The authors would like to thank the anonymous reviewers for their comments which were very helpful in improving the quality and presentation of this paper.

References:

- [1] Bernard Sklar, "Digital communication Fundamentals and Applications", 2nd edition, Prentice Hall, ISBN:81-7808-373-6, 2001.
- [2] http://www.vocal.com/data_sheets/802.11a5.html (last accessed on 12/11/08)
- [3] http://www.doe.carleton.ca/~jknight/97.48/97.478_01F/Conv1_2LabsB.html (last accessed on 21/12/08)
- [4] Michael Pursler, "Introduction to Error- correction codes", Artech House INC, ISBN: 0-89006-784-8, 1996.
- [5] Shu Lin and Daniel J. Costello, "Error Control Coding Fundamentals And Applications", 2nd edition, Prentice Hall, 1984.
- [6] Fei Sun and Tong Zhang, "Low-Power State-Parallel Relaxed Viterbi Decoder", IEEE Transactions on Circuits and systems, Vol. 54, Page(s)-1060-1069, No. 5, May 2007.
- [7] ftp://ftp.cs.man.ac.uk/pub/amulet/theses/Shao07_phd.html (last accessed on 2/2/09)
- [8] Rex Andrew Antony, "An Adaptive threshold strategy for soft decision Viterbi Decoder", Dalhousie university, December 2002
- [9] QIN Xiang-Ju', ZHU Mmg -Cheng', WEI Zhong-Yi2, CHAO Du', "An Viterbi Decoder Based on FPGA Dynamic Reconfiguration Technology", IEEE International Conference on Field-Programmable Technology 2004, Vol. 10, Page(s)-6-8 December, 2004.
- [10] Ming-Hwa Chan, Wen-Ta Lee, Mao-Chao Lin and Liang-Gee Chen, "IC Design of an Viterbi Decoder", IEEE Transactions on Consumer Electronics, Vol. 42, Page(s)-52-62, No. 1, February 1996
- [11] Man Guo, M. Omair Ahmad, M.N.S. Swamy, and Chunyan Wang, "A Low-Power Systolic Array-Based Viterbi Decoder and its FPGA Implementation", International Symposium on Field-Programmable Technology 2003, Vol 2, Page(s)- 276 - 279, 25-28 May 2003.
- [12] Abdulfattah M. Obeid, Alberto Garcia, Mihail Petrov, Manfred Glesner, "A Multi-path high speed Viterbi decoder", Proceedings of the 2003 10th IEEE International Conference on Electronics, Circuits and Systems, 2003. ICECS 2003, Vol 3, Issue, 14-17 Page(s): 1160 – 1163, December 2003
- [13] <http://asic-soc.blogspot.com/2007/10/physical-design-flow.html> (last accessed on 1/02/08)
- [14] Samir Palnitkar, "Verilog HDL – A Guide to Design and Synthesis", 2nd edition, Prentice Hall, ISBN: 0-13-044911-5 2003.

Authors Profile:



KUMAR PARI is Pursuing his M. Tech from Chirala Engineering College, Chirala in the department of Electronics & Communications Engineering (ECE) with specialization in VLSI & Embedded systems.



B.RAGHAVIAH is working as an Associate Professor in the department of Electronics & Communication Engineering in Chirala Engineering College, Chirala. He has completed masters from JNTUK. He has over 8 years of teaching experience.