International Journal of Research, Science, Technology & Management
ISSN Online: 2455-2240, Volume 4 Issue I, July 2016

IJRSTM
ISSN ONLINE: 2455-2240

# FPGA Implementation of Viterbi Decoder using Trace back Architecture

Ankita Rawat, Parul Singh

***Abstract*—Error correction is an integral part of any communication system and for this purpose, the convolution codes are widely used as forward error correction codes. For decoding of convolution codes, at the receiver end Viterbi Decoder is being employed. The parameters of Viterbi algorithm can be changed to suit a specific application. The high speed and small area are two important design parameters in today's wireless technology. In this paper, a high speed feed forward viterbi decoder has been designed using track back architecture and embedded BRAM of target FPGA. The proposed viterbi decoder has been designed with Matlab, simulated with Xilinx DSP Tool, synthesized with Xilinx Synthesis Tool (XST), and implemented on Xilinx Spartan 3E based xc3s500e FPGA device. The results show that the proposed design can operate at an estimated frequency of 86.6 MHz by consuming considerably less resources on target device to provide cost effective solution for wireless applications.***

**Keywords—DSP, FPGA, Matlab, Viterbi Decoder, XST**

## I. Introduction

Viterbi algorithm is being widely used in many wireless and mobile communication systems for optimal decoding of convolutional codes. Convolutional encoding with Viterbi decoding is a Forward Error Correction technique that is particularly suited to a channel in which the transmitted signal is corrupted mainly by additive white gaussian noise (AWGN). The purpose of forward error correction (FEC) is to improve the capacity of a channel by adding some carefully designed redundant information to the data being transmitted through the channel [1]. The Viterbi algorithm essentially performs maximum likelihood decoding to correct the errors in received data which are caused by the channel noise; however it reduces the computational load by taking advantage of special structure in the code trellis [2]**.** The Viterbi algorithm (VA) is a recursive optimal solution to the problem of estimating the state sequence of a discrete time finite-state Markov process.

Viterbi decoding has the advantage that it has a fixed decoding time and it is well suited to hardware decoder implementation. The requirements for the Viterbi decoder, which is a processor that implements the Viterbi algorithm, depend on the application in which it is used. Viterbi Algorithm is effective in achieving noise tolerance, but the cost is an exponential growth in memory, computational resources and power consumption. This paper proposes low power architecture for developing a viterbi decoder for various digital receivers.

## II. Design Algorithm

The Viterbi algorithm proposed by A.J. Viterbi in 1967 is a computationally efficient technique for determining the most probable path taken through a Markov graph. The decoding procedure can be explained by a trellis diagram. The trellis requires $2K-1$ states at each stage, where $K$ is the constraint length in convolutional encoding [3]. The stage is given by the length in bits of the message to be decoded. This algorithm calculates a measured distance between the received symbol and all possible paths at a certain stage in the trellis diagram. The viterbi decoding flowchart is given in Fig.1

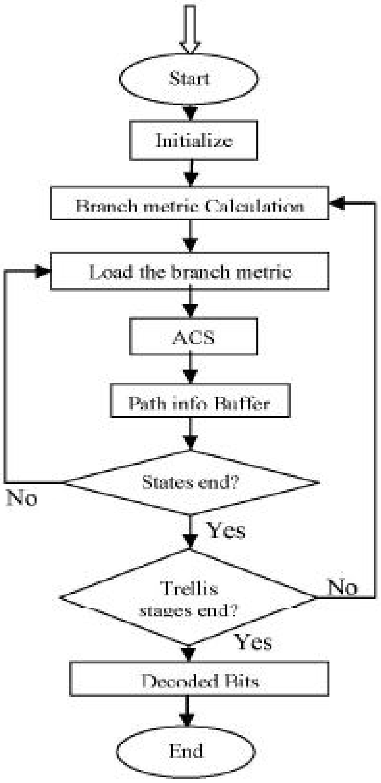

Fig.1 Viterbi Decoding Flowchart

The Viterbi algorithm applies the maximum-likelihood principle. The most common metric used is the Hamming distance metric. This is just the dot product between the received codeword and the allowable codeword. These metrics are cumulative so that the path with the largest total metric is the final winner. 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.

## III. VITERBI DECODER ARCHITECTURE

The general structure of Viterbi decoder is shown in Fig.2, which mainly consists of 3 parts: BMU (branch metric unit), ACSU (add compare select unit), and SMU (survivor management unit) [4].

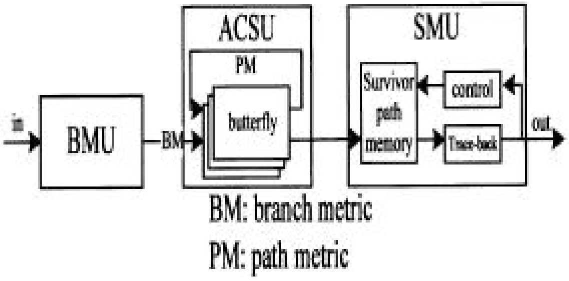

Fig. 2 Block Diagram of Viterbi Decoder

### A. Branch Metric Unit

BMU calculates all the Branch Metrics (BMs) from the received symbols. BMU is transition metric unit and its function is to generate corresponding merit of skip branch according to the input code sequence. The generated sequence merit is delivered to ACS unit, completing the process as bit calculation. Then BMs are fed into the add-compare-select unit

### B. Add Compare Select Unit

ACSU recursively computes path metrics and outputs decision bits for each state transition. The ACS module not only receives the code sequence from the branch merit module, but needs the path merit of last state and information related to state shift. It is necessary to calculate the sum of branch merit and path merit firstly, and then select the smallest path merit. For a given code with rate 1/n and total memory M, the number of ACS required to decode received sequence of length L is L×$2^M$.

### C. Survivor-path memory unit

It is responsible for keeping track of the information bits associated with the surviving paths. SMU uses these bits to find the final survivor path and decode the source bits. The arbitral bit which is generated from the ACS module will be saved in the survival path memory and will be used to search the surviving path. There are two basic approaches for searching survival path. One is named register-exchange (RE) and the other is called Trace-Back (TB).

Register Exchange (RE) is the most straightforward method to extract the information bits from the encoded bits stream. It has the lowest decoding latency and simple control circuit. RE is implemented by the connection of multiplexers and registers according to the trellis diagram, and its memory requirement is *NL* bits registers. As decoding, all *NL* bits are read and written, and it requires high memory access bandwidth. It consists of a two dimension register array with multiplexers between each two columns. RE is attractive in terms of the regularity of circuit structure and decoding latency [5]. However, in practical applications, when VD dealing with convolutional codes with large constraint length *K* (i.e., $K \geq 7$), RE becomes impractical due to its high power consumption and large routing overhead. Therefore it is not suited for low power applications such as wireless communications systems. One the other hand, Traceback (TB) traces back the maximum likelihood path starting from the best state. It traces back the survivor path after the entire code word has been received and generates the decoded output sequence [6]. It is a combinational block and is active during only one clock cycle In TB method, only *N* decision bits are written in each cycle, and RAM can be utilized here The power consumption of the TB architecture is much more efficient than RE when we have large constraint lengths [10]. Therefore traceback architecture is well suited for low power applications. Traceback memory stores the history of decision bits from ACS operation. This memory is a two dimensional circular buffer, with rows and columns. The number of rows is equal to the number of states N = 2 K-1 where K is the constraint length. Each column stores the results of N comparisons corresponding to incoming coded bits at each time interval. The number of columns is equal to the Trace back depth [7].

## IV. MATLAB BASED PROPOSED DESIGN SIMULATION

Clock gating is a power-saving and speed enhancement technique used in system-on-chip designs [8]. The basic concept is to de-activate the clocks for functions when they are not required and thereby reducing their power consumption and enhancing the speed since the signal will not be propagating through those parts of the circuit hence making it faster. In our proposed architecture, we will be exploiting this feature in the traceback module. In the traceback approach, each register storing the survivor path information updates its content only once during the entire period of a code word. The registers and the traceback module are active only for one clock period during the entire period of a code word in the traceback approach. The important factor is that the content of each register does not change as soon as it is updated. This forms the basis of low power and high speed design, as the registers do not have to be activated after each update which results in a reduction in the switching activity and hence increasing the speed of the circuit and a reduction in power dissipation can be reached [9],[4].

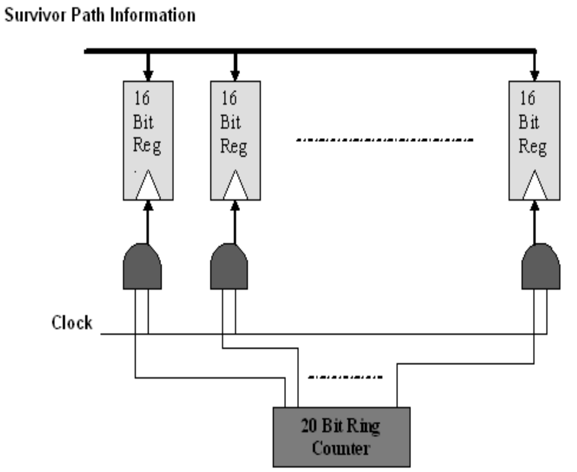

Fig.3 Survivor Memory Unit

Our proposed Survivor Memory Management module is given in Fig.5 with 20 registers each of 16 bits. The clock is enabled for only one clock cycle for a particular register, when it updates its survivor path information. The clock for other registers remains off during that time, thus saving power.

The proposed high Speed design is first developed using MATLAB code for the constraint length 5 and rate ½ encoder and the generating polynomials as (35,23). The proposed viterbi decoder has been developed using track back feed forward technique. Fig.4 shows how errors can be effectively removed using the viterbi decoder.

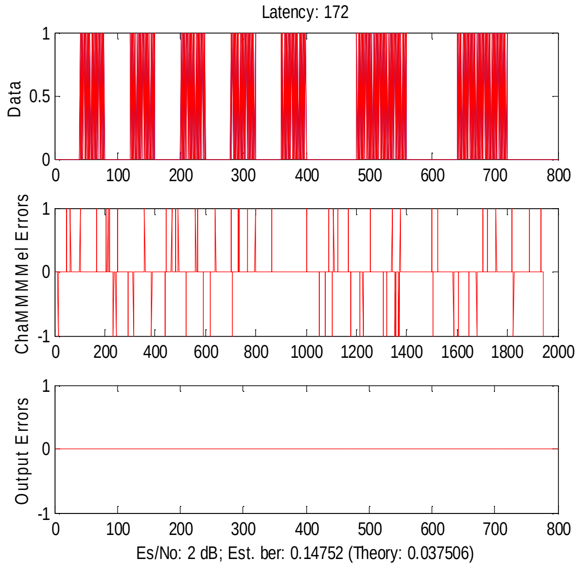

Fig.4 Simulation Results for viterbi decoder

Fig.5 shows how well the transmitted image can be received using viterbi decoding as compared to the case when viterbi decoding is not used. The figure tells how powerful a tool is viterbi decoding when long distance transmission of data is required.

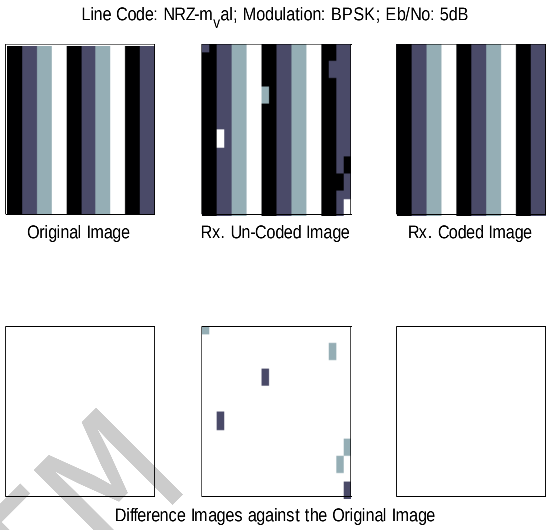

Fig.5 Coded vs. Un-coded Data Reception Results

## V. Hardware Implementation Results

To observe the speed and resource utilization, RTL is generated, verified and synthesized using Xilinx Synthesis Tool (XST) and implemented on Xilinx Spartan 3e based xc3s500e FPGA device. The benefits associated with FPGA such as flexibility, shorter time to market and re-configurability make them a very attractive choice for implementing the designs. The user programmability gives the user access to complex integrated designs without the high engineering costs associated with application specific integrated circuits. The benefits of clock gating are very much clear from the results since clock gating helps in switching off the parts of the circuit when not in operation, hence helping in power saving and increasing the speed of the circuit, because the clock will not be propagating through those parts of the circuit. Table.1 shows the device utilization summary.

Table 1: Device Utilization Summary

| Device Utilization Summary | | |
|---|---|---|
| **Logic Utilization** | **Used/Available** | **Utilization** |
| Number of Slices | 1550/4656 | 33% |
| Number of Slice | 1610/9312 | 17% |

| | | |
|---|---|---|
| Flip-flops | | |
| Number of 4 input LUTs | 2913/9312 | 31% |
| Number of BRAMs | 4/20 | 20% |

The proposed embedded BRAM and LUT based design can work at an estimated frequency of 86.6 MHz as compared to 47 MHz in case of [4] by using considerable less resources of target FPGA.

## VI. Conclusion

In this paper a high speed viterbi decoder has been proposed. The proposed viterbi decoder has been designed with MATLAB using track back feed forward technique. The embedded BRAM and LUTs of target FPGA have been efficiently utilized to enhance the speed of the developed decoder. The designed viterbi decoder has been simulated using Xilinx DSP Tool, synthesized with XST and implemented on Spartan 3E based xc3s500e target FPGA device. The results show that proposed design can work at an estimated frequency of 86.6 MHz by using considerable less resources of target FPGA to provide high performance cost effective solution for wireless communication applications.

.

## REFERENCES

[1] Jinjin H, Zhongfeng Wang, Zhiqiang Cui, Li Li, “Towards an Optimal Trade-ff of Viterbi Decoder Design”, Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium, 24-27 May 2009 pp. 3030 – 3033.

[2] Habib, I. Paker, O. Sawitzki, S., “Design Space Exploration of Hard-Decision Viterbi Decoding: Algorithm and VLSI Implementation”, IEEE Transaction on very Large Scale Integration (VLSI) Systems, May 2010, pp. 794-807.

[3] Adam O., Shengli Fu, Varanasi M, “Hardware Efficient Encryption Encoder and Decoder Unit”, Military Communication Conference, IEEE, 16-19 Nov. 2008, pp. 1-6.

[4] Sherif Welsen Shaker, Salwa Hussien Elramly and Khaled Ali Shehata, “Design and Implementation of Low-Power Viterbi Decoder for Software-Defined WiMAX Receiver”, TELFOR, IEEE, November 2009.

[5] Chun-Yuan Chu1, Yu-Chuan Huang2, An-YeuWu3, “Power Efficient Low Latency Survivor Memory Architecture for Viterbi Decoder”, IEEE International Symposium on VLSI Design, 2008, pp. 228 – 231.

[6] Mohamed Farid Noor Batcha, Ahmad Zuri Sha’ameri, “Configurable Adaptive Viterbi Decoder for GPRS, EDGE and Wimax”, International Conference on Telecommunications, 2007, pp. 237-241, 2007

[7] Shin-Pao Cheng, Shi-Yu Huang, “A Low Power SRAM for Viterbi Decoder in Wireless Communication” Consumer Electronics, IEEE Transactions, Volume 54, Issue 2,May 2008,pp.290-295.

[8] Sherif Welsen Shaker, Salwa Hussien Elramly and Khaled Ali Shehata, “FPGA Implementation of a Configurable Viterbi Decoder for Software Radio Receiver”, Autotestcon, IEEE, July 2009, pp. 140 – 144.

[9] Chien-Ching Lin, Yen-Hsu Shih, Hsie-Chia Chang and Chen-Yi Lee, “Design of a Power Reduction Viterbi Decoder for WLAN applications”, IEEE Transactions on Circuits and Systems, Vol.52, No.6, June 2005, pp. 1148-1156.

[10] Y. Gang, A. T. Erdogan, and T. Arslan, “An efficient pre-traceback architecture for the Viterbi decoder targeting wireless communication applications,” IEEE Trans. Circuits Syst. I, Reg. Papers, vo1. 52, no. 6, June 2005, pp.1148–1156.