IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Power Efficient Viterbi Decoder Using Trace Back Architecture

Swati Gupta
ECE Department, PIET, Samalkha, Panipat, India
swatigupta13@gmail.com

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 and for their decoding at the receiver end viterbi decoders are employed. Viterbi Decoders are employed in digital wireless communication systems to decode the convolution codes which are the forward correction codes. The decoders are quite complex and dissipate large amount of power. The high speed and small area are two important design parameters in today's wireless technology. In this paper, a high speed viterbi decoder has been designed using track back architecture. The proposed viterbi decoder has been designed with Matlab, simulated with Xilinx Xilinx DSP Tool, synthesized with Xilinx Synthesis Tool (XST). The results show that the proposed design can operate by consuming considerably less resources on target device

Keywords: FPGA, Matlab, Viterbi Decoder, Clock Gating, XST

Introduction

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 [4]. 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.

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.

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. It is one of the best techniques for communications, especially wireless where energy efficiency is the most important factor. It works on the principle of selecting a code word closest to the received word. 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

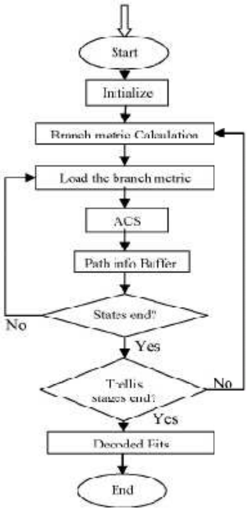

Fig.1 Viterbi Decoding Flowchart

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.

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].

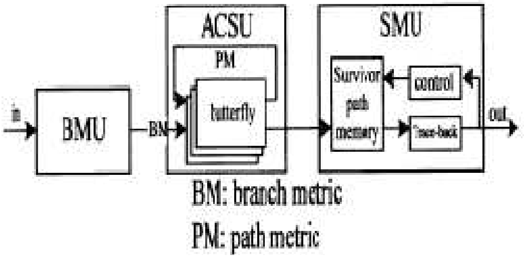

Fig. 2 Block Diagram of Viterbi Decoder

Branch Metric Unit

The branch metric unit (BMU) takes the fuzzy bit and calculates the cost for each branch of the trellis. A simple branch metric unit may use hamming or Euclidean distance as the metric for calculating the cost of the branch [7]. It is 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.

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 metric module but needs the path merit of last state and information related to state shift. For a given code with rate 1/n and total memory M, the number of ACS required to decode receive sequence of length L is $L\times2^M$.

Survivor Memory Unit

The survivor memory unit (SMU) is responsible for keeping track of the information bits associated with the surviving paths designated by the path metric updating and storage unit. There are two basic design approaches for SMU: Register Exchange and Trace Back. In both techniques, a shift register is associated with every trellis node throughout the decoding operation. This register has a length equal to the frame length. RE is implemented by the connection of multiplexers and registers according to the trellis diagram, and its memory requirement is *NL* bits registers. The register exchange method works well for small constraint lengths. For 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. The trace back method works well for longer constraint length codes. It traces back the survivor path after the entire code word has been received and generates the decoded output sequence [6]. The trace back method stores the decisions from the ACS into a RAM and also the path information in the form of an array of recursive pointers [9]. The best path is determined by reading backwards through the RAM. The general approach to trace back is to accumulate path metrics for up to five times the constraint length ($5 * (K - 1)$), find the node with the largest accumulated cost, and begin trace back from this node [15]. The trace-back unit can then output the sequence of branches used to get to that state. In practice, the survivor paths merge after some number of iterations. The trellis depth at which all the survivor paths merge with high probability is referred to as the survivor path length.

Matlab Based Proposed Simulation

Clock gating is a power-saving and speed enhancement technique used in system-on-chip designs [2]. 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].

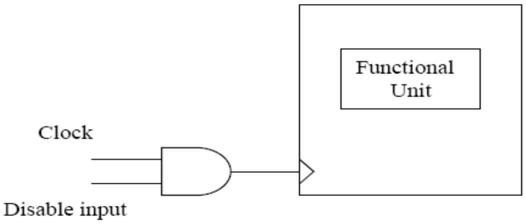

Fig. 3 Clock Gating

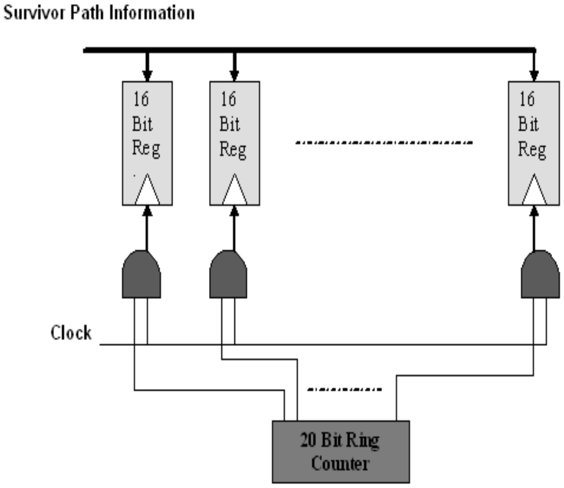

Fig.4 Survivor Memory Unit

Our proposed Survivor Memory Management module is given in Fig.4 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.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.

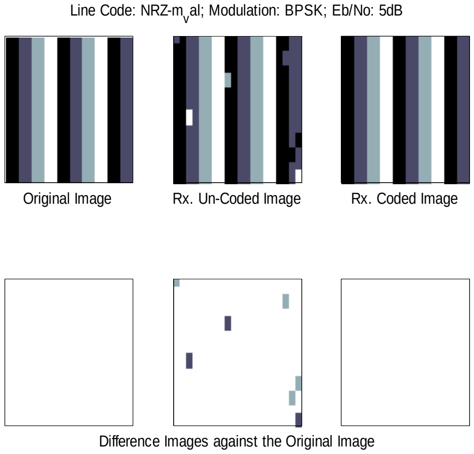

Fig.5 Coded vs. Un-coded Data Reception Results

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.

Device Utilization Summary		
Logic Utilization	**Used/Available**	**Utilization**
Number of Slices	1550/4656	33%
Number of Slice Flip-flops	1610/9312	17%
Number of 4 input LUTs	2913/9312	31%
Number of BRAMs	4/20	20%

Table 1: Device Utilization Summary

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 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 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] 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.

[3] Jinjin He, Huaping Liu, Zhongfeng Wang, “A fast ACSU architecture for Viterbi decoder using T-algorithm”, Forty-Third Asilomar Conference on signals, Systems and Computers, 2009, pp. 231-235.

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

[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] 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.

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

[8] Wei Shao, Brackenbury L., “Pre-processing of convolutional codes for reducing decoding power consumption”, IEEE International Conference on Acoustics, Speech and Signal Processing, 2008 , pp. 2957 - 2960

[9] 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.

[10] 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.

[11] 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.