

Design and Implementation of a Modified Register Exchange Based Adaptive VITERBI Decoder

Shiny Paul

PG Scholar,

Dept. of ECE (VLSI),

Baba Institute of Technology and Sciences,

JNTUK, AP, India.

B. Ratna Raju

Associate Professor,

Dept. of ECE,

Baba Institute of Technology and Sciences,

JNTUK, AP, India.

Abstract:

This project work designs and implements an optimized solution for a widely used forward error correcting algorithm known as Convolutional Encoder with Viterbi Decoder. Convolutional encoding with Viterbi decoding is a good forward error correction technique suitable for channels affected by noise degradation. Viterbi decoder uses Viterbi algorithm which is an optimum decoding algorithm for Convolutional code if it is applied to transmission in additive white Gaussian noise channel. The Viterbi algorithm occupies large memory, computational resources and consumes more power. To address this problem Adaptive Viterbi Algorithm was introduced. The Adaptive Viterbi Decoder (AVD) uses a survivor management unit (SMU) for finding the best winner bits. This is done by using either a trace back (TB) method or a register exchange (RE) method. The trace back method requires complex hardware, so register exchange method is preferred. But the register exchange method results in increased computational complexity, increased area, more computation time and more power consumption. Therefore in this project, to address this issue we are using a modified register exchange based Adaptive Viterbi Decoder. The modified RE based AVD reduces the number of registers in the design thereby reducing the area, increases the frequency thereby reducing the time and hence the result is increased speed. The modified RE based AVD uses a new technique called T-Algorithm which adopts truncation length for decoding. This technique reduces the number of times the path memory is accessed thereby reducing

the power consumption of the system. The system is realized using Verilog HDL. It is simulated and synthesized using Xilinx ISE (v14.2i)

Keywords: Convolutional Encoder, Forward Error Correction (FEC), Modified Register Exchange Method, T-Algorithm, Adaptive Viterbi Decoder.

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 it is applied to transmission in additive white Gaussian noise channel. The Viterbi algorithm (VA) occupies large memory, computational resources, and power consumption to address this issue adaptive Viterbi algorithm (AVA) is introduced. Encoding the information sequence prior to transmission implies adding extra redundancy to it, which is then used at the receiver end to reconstruct the original sequence, effectively reducing the probability of errors induced by a noisy channel. Different structures of codes have developed since, which are known as channel coding. The encoder adds redundant bits to the sender's bit stream to create a code word. The decoder uses the redundant bits to detect and/or correct as many bit errors as the particular error control code will allow. Like any error correcting code, a Convolutional code works by adding some structured redundant information to the user's data and then correcting errors using this information. There have been a few Convolutional decoding methods such as sequential and Viterbi decoding, of which the most commonly employed

technique is the Viterbi Algorithm (VA). Viterbi decoding was developed by Andrew. J. Viterbi, the founder of Qualcomm Corporation in April, 1967. Since then, other researchers have expanded on Viterbi's work by finding good Convolutional codes, exploring the performance limits of the technique, and varying decoder design parameters to optimize the implementation of the technique in hardware and software. Viterbi algorithm is being widely used in many wireless and mobile communication systems for optimal decoding of Convolutional codes. The Viterbi alignment is a dynamic programming algorithm for finding the most likely sequence of hidden states – called the Viterbi path – that results in a sequence of observed events, especially in the context of Markov information sources and hidden Markov models. Applications using Viterbi decoding include digital modems and digital cellular telephone, where low latency, component cost and power consumption are must.

1. Digital communication system

The basic block diagram of a digital communication system is shown in figure 1. The channel is always susceptible to errors because of noise so to improve the channel capacity we add some redundant bits into the binary information. This is done by the encoder and the process is called as channel coding. The demodulator will process the corrupted waveform and detector will decide whether the signal is a 0 or a 1. The decoder reconstructs the original bits by using the redundant bits and then exploits the redundant bits to correct for channel disturbances.

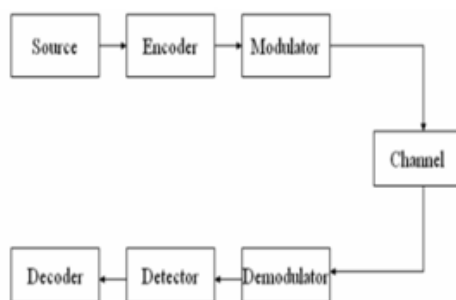


Figure 1: Block diagram of a Digital Communication System

2. FEC (Forward Error Correction)

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). There are basically two mechanisms for Error correction code:

1. Block coding
2. Convolution coding

In this project, Convolution Coding with Adaptive Viterbi Decoder using FEC is considered for analysis

II. CONVOLUTIONAL CODES

Convolutional encoder takes k bits as input and produces n output bits. Convolutional encoder takes k input bits but here these k bits are not taken as a block. The convolutional encoder processes these k bits in a serial manner instead of processing as a block and it produces n output bits. So this eliminates the necessity for using a buffer. A convolutional encoder consists of:

- a. Shift registers (made up of memory elements i.e. flip flops)
- b. Modulo-2 adders

The inputs and outputs of flip flops are in some way connected to the modulo-2 adders to produce the outputs. Here we are considering a rate i.e. $r = k/n = 1/2$ and constraint length $K=3$ for analysis. Where k = input, n =output, r = code rate and K =constraint length. If a convolutional encoder takes one input bit and produces two output bits then the convolutional encoder is called rate, i.e. $r = k/n$, or $r = 1/2$ convolutional encoder.

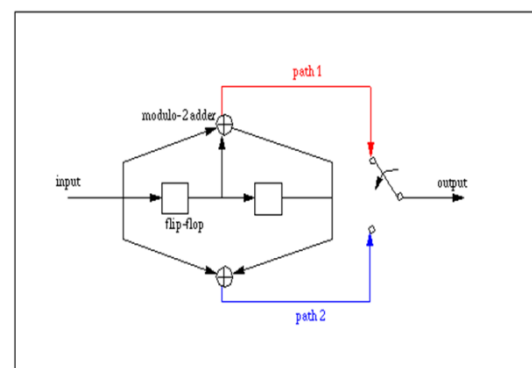


Figure 2: Structure of a convolutional encoder

In the figure 2. we are considering a rate $r = \frac{1}{2}$ convolutional encoder. It has two outputs so we use two adders. At the output there will be a multiplexer or a switch. So this convolutional encoder consists of one input and two outputs and two memory elements. The constraint length can be defined as the number of memory shifts required for the input bit to influence the output bit. So K- constraint length is the number of memory shifts required for the input bit to influence the output. Now if an input bit is present then for one clock pulse there will be a shift that means for one shift the input will be in the first memory element. For second clock pulse it will be shifted to the second memory element. For third clock pulse it will be shifted to influence the output. So for this structure of rate $\frac{1}{2}$ with two memory elements for the input bit to influence the output three memory shifts are required. So the constraint length is $K=3$. In general for a single stage structure of convolutional encoder the constraint length $K=M+1$ where, M =number of memory elements. Convolutional encoder can be described in terms of state table, state diagram and trellis diagram.

The State is defined as the contents of the shift register of the encoder. In state table output symbol can be described as a function of input symbol and the state.

Input	Current state	Next state	Output
0	00	00	00
1	00	10	11
0	10	01	10
1	10	11	01
0	01	00	11
1	01	10	00
0	11	01	01
1	11	11	10

Table 1: State table of a convolutional encoder

State diagram shows the transition between different states.

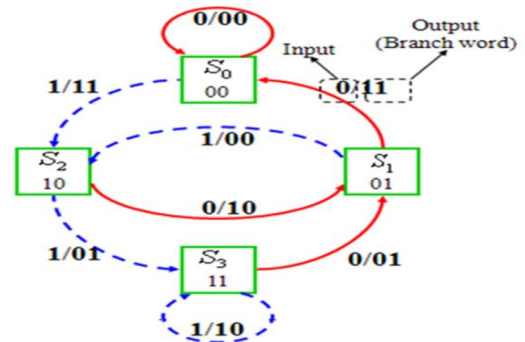


Figure 3: State diagram of a convolutional encoder

Trellis is a description of the state diagram and it is represented by different time lines.

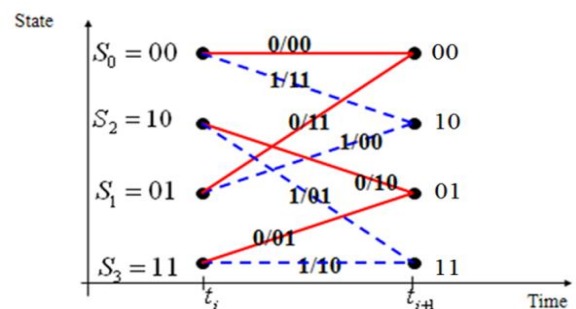


Figure 4: Trellis of a convolutional encoder

III. VITERBI ALGORITHM

Viterbi algorithm was devised by Andrew J. Viterbi (1967). The optimality and the relatively modest complexity for small constraint lengths have served to make the Viterbi algorithm the most popular in decoding of convolutional codes with constraint length less than 10. Viterbi algorithm is called as optimum algorithm because it minimizes the probability of error. The Viterbi algorithm is one of the standard sections in number of high-speed modems of the process for information infrastructure applicable in modern world. The unit of branch metric will calculate all the branch metrics and then processed to add compare for selecting the surviving branches as per the branch metrics finally the decoded data bits are generated by the trace back unit. The algorithm can be broken down into the following three steps.

1. Weight the trellis; that is, calculate the branch metrics.

2. Recursively computes the shortest paths to time n , in terms of the shortest paths to time $n-1$. In this step, decisions are used to recursively update the survivor path of the signal. This is known as add-compare-select (ACS) recursion.

3. Recursively finds the shortest path leading to each trellis state using the decisions from Step 2. The shortest path is called the survivor path for that state and the process is referred to as survivor path decode. Finally, if all survivor paths are traced back in time, they merge into a unique path, which is the most likely signal path.

IV. ARCHITECTURE OF AN ADAPTIVE VITERBI DECODER

An adaptive Viterbi decoder consists of a branch metric unit, add compare and select unit, survivor management unit which includes the best winner search unit and a non-survivor purge unit.

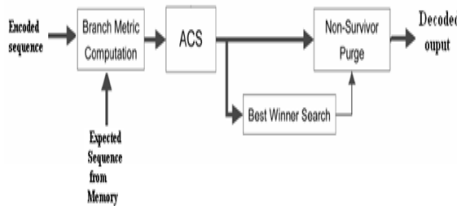


Figure 5: Block diagram of an adaptive Viterbi decoder

1. Branch Metric Unit

The first unit is called Branch metric unit. The Hamming distance (or other metric) values we compute at each time instant for the paths between the states at the previous time instant and the states at the current time instant are called branch metrics. Hamming distance or Euclidean distance is used for branch metric computation. Figure 6. shows the basic block diagram for a BMU.

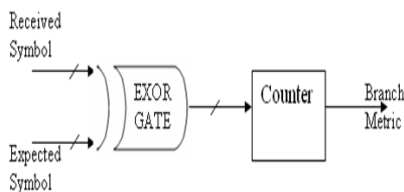


Figure 6: Block diagram of a branch metric unit

2. Add and Compare Select 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.

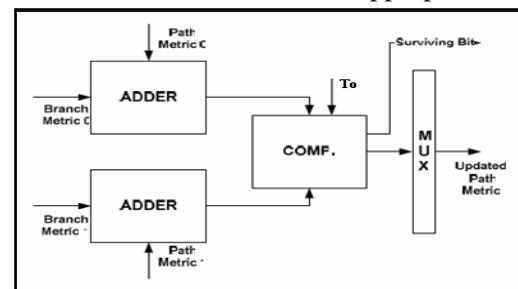


Figure 7: Block diagram of an ACS unit

3. Survivor Management Unit Using Modified Register Exchange Based Method

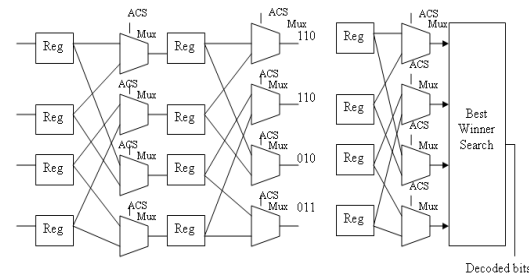


Figure 8: Architecture of a modified register exchange based survivor unit

Figure 8. Shows the modified register exchanged based survivor unit. The trellis coding structure is divided into two segments and the Trellis Window is constructed by 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 are out at the right of the window. First local winner is chosen and then survivor bit of local winner from ACS unit is given to register for storing. Decoding is done at the end of the trellis once best winner search is chosen as final

winner. This modified architecture reduces the size and processing time of the decoder.

4. T-Algorithm

T-Algorithm reduces the number of states so the complexity of computation decreases. It is a sorting process or comparison operation for searching the best path or minimum path metric in decoding state. T-Algorithm is used for finding the optimal path for reducing the number of states. For this it uses a parameter called as the truncation length (τ). For a $K=3$ and $\text{rate}=\frac{1}{2}$ Convolutional encoder the truncation length is given by 3. Truncation length (τ) controls the average number of path stored per trellis at each stage and the frequency with which trace back length through the trellis is performed. The reason why power consumption can be significantly reduced for adaptive by reducing truncation length is mainly because these parameters greatly affect the number of times path memory is accessed. Actually, the number of times nearly all calculations are performed per trellis stage is proportional to average number of surviving path per stage, which is controlled by truncation length. Adaptive Viterbi decoder is functionally same as Viterbi decoder when no error is introduced. Once error is introduced in input Adaptive part come into picture. It checks every node for path metric value equal to or higher than τ . If path is found them it is eliminated, else usual action takes place on node like normal Viterbi decoder. So data travelling in trellis of Adaptive Viterbi decoder is entirely different to its counterpart Viterbi decoder in presence of errors. In Trellis path with value greater or equal to 3 are eliminated.

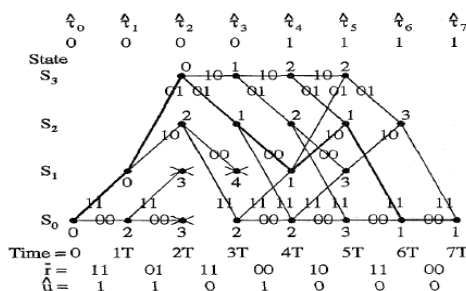


Figure 9: Trellis diagram of AVD for $k=3$ at rate $\frac{1}{2}$ with $\tau=3$

V. SIMULATION AND SYNTHESIS RESULTS

The Simulation Waveform of adaptive Viterbi Decoder is shown in Fig10. The device utilization summary is shown in which it gives the details of number of devices used from the available devices and it is represented in %. To observe the speed and resource utilization, RTL is generated, verified and synthesized using Xilinx Synthesis Tool (XST) and the power details are obtained by using the analyze power distribution option.

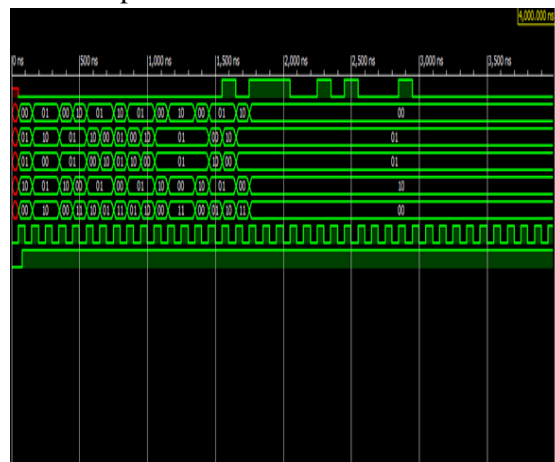


Figure 10: Simulation result

viterbi Project Status (11/06/2016 - 18:21:17)			
Project File:	shiny.vise	Parser Errors:	No Errors
Module Name:	viterbi	Implementation State:	Synthesized
Target Device:	xc3s100e-5iq100	Errors:	No Errors
Product Version:	ISE 14.2	Warnings:	14 Warnings (14 new)
Design Goal:	Balanced	Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	Timing Constraints:	
Environment:	System Settings	Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices		113 / 960	11%
Number of Slice Flip Flops		62 / 1920	3%
Number of 4 input LUTs		199 / 1920	10%
Number of bonded IOBs		11 / 66	16%
Number of GCUs		1 / 24	4%

Figure 11: Device utilization summary (estimated values)

Timing Summary:

Speed Grade: -5

Minimum period: 11.885ns (Maximum Frequency: 84.139MHz)
 Minimum input arrival time before clock: 13.634ns
 Maximum output required time after clock: 4.040ns
 Maximum combinational path delay: No path found

Figure 12: Timing analysis

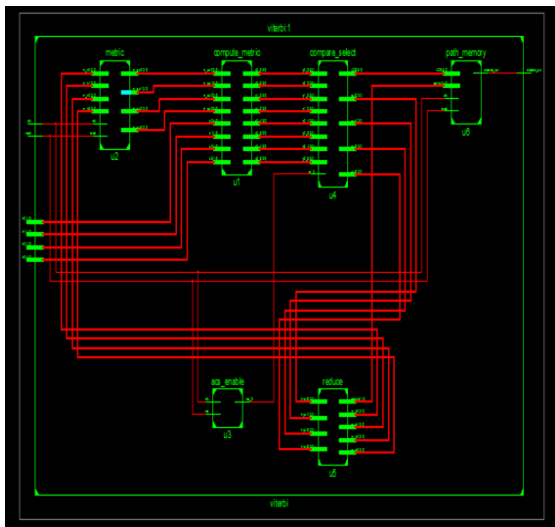


Figure 13: RTL Schematic

Device	On-Chip Power (W)	Used	Available	Utilization (%)	Supply Summary	Total	Dynamic	Quiescent
Family	Spartan6e	Clocks	0.000	2	Source	Voltage	Current (A)	Current (A)
Part	xc6s100e	Logic	0.000	190	Vcore	1.200	0.000	0.000
Package	vq100	Signals	0.000	232	Vccaux	2.500	0.000	0.000
Temp Grade	Commercial	I/Os	0.000	11	Vcc25	2.500	0.002	0.000
Process	Typical	Leakage	0.004	66				
Speed Grade	-5	Total	0.004	17				
					Supply Power (W)			
					Total	Dynamic	Quiescent	
					0.004	0.000	0.004	

Environment	Thermal Properties	Effective T _J	Max Ambient	Junction Temp
Ambient Temp (C)		(C/W)	(C)	(C)
25.0		49.0	83.4	26.6

Figure 14: Power analysis

VI. CONCLUSION & FUTURE WORK

Conclusion:

In this project we have successfully enhanced the decoder in such a way that it works better in several aspects as compared to the previous one. From the study of device utilization summary section in synthesis report it can be seen that the major goal achieved is minimum device utilization so the area required is reduced and from timing analysis it is understood that it took less processing time so thereby the speed is increased. Also by following the new technique i.e. T-algorithm for decoding, we are able to reduce the power consumption. The modified design of the new decoder helped us with reduction in computational complexity too. And the use of error-correcting codes has proven to be an effective way to overcome data corruption in digital communication channels.

Future work:

Implementation of hard decision based adaptive Viterbi decoder with modified register exchange method for constraint length $k=3$ at rate $1/2$ is discussed here. The Viterbi decoding is limited to lower constraint lengths. Same work can be extended for soft decision based adaptive Viterbi decoder which suits multiple quantization level by modifying Branch metric computation unit accordingly. Adaptive Viterbi decoder with Modified Register Exchange Method can be further investigated for higher constraint lengths.

VII. REFERENCES

- [1] Bernard Sklar, "Digital communication Fundamentals and Applications", 2nd edition, Prentice Hall, ISBN: 81-7808-373-6, 2001.
- [2] Michael Purser, "Introduction to Error- correction codes", Artech House INC, ISBN: 0-89006-784-8, 1996.
- [3] Shu Lin and Daniel J. Costello, "Error Control Coding Fundamentals And Applications", 2nd edition, Prentice Hall, 1984.

[4] Fei Sun and Tong Zhang , “Low-Power State-Parallel Relaxed Adaptive Viterbi Decoder” , IEEE Transactions on Circuits and systems , Vol. 54, Page(s)-1060-1069, No. 5, May 2007.

[5] Rex Andrew Antony,” An Adaptive threshold strategy for soft decision Viterbi Decoder”, Dalhousie university, December 2002.

[6] QIN Xiang-Ju', ZHU Mmg -Cheng', WEI Zhong-Yi2, CHAO Du,, , “An Adaptive Viterbi Decoder Based on FPGA Dynamic Reconfiguration Technology”, IEEE International Conference on Field-Programmable Technology 2004, Vol. 10, Page(s)-6-8 December , 2004.

[7] Ming-Hwa Chan, Wen-Ta Lee, Mao-Chao Lin and Liang-Gee Chen , “IC Design of an Adaptive Viterbi Decoder”, IEEE Transactions on Consumer Electronics, Vol. 42, Page(s)-52-62 ,No. 1, February 1996

[8] Man Guo, M. Omair Ahmad, M.N.S. Swamy, and Chunyan Wang , “A Low-Power Systolic Array-Based Adaptive Viterbi Decoder and its FPGA Implementation”, International Symposium on Field-Programmable Technology 2003, Vol 2, Page(s)- 276 - 279, 25-28 May 2003.

[9] 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.

[10]http://ftp.cs.man.ac.uk/pub/amulet/theses/Shao07_phd.html (last accessed on 2/2/09).

[11]http://www.vocal.com/data_sheets/802.11a5.html (last accessed on 12/11/08)

[12]http://www.doe.carleton.ca/~jknight/97.478/97.478_01F/Conv1_2LabsB.html (last accessed on 21/12/08).