Simulation of Convolution Encoder and Viterbi decoder Using Verilog HDL

Rakhi B. Menon¹, Dr. Gnana Sheela K²

M.Tech Student¹, Professor², Department of ECE, TOCH Institute of Science and Technology, Arakkunnam

Abstract— Convolution encoder and Viterbi decoder of rate 2/3 is simulated using Model Sim. Test benches for encoder and decoder are coded and simulated. Two bits are provided to the encoder as input and three bits are obtained as output of encoder. These encoded three bits are provided as input to the Viterbi decoder. Viterbi decoder consists of different modules such as Euclidean distance calculation module, Subset decode module, Compute metric module, Compare Select Module, Path module, Path memory module, Path in Module, Metric Module, Output decision module and reduce module. All these modules are internally connected by Verilog HDL codes and simulated using Model Sim. The output of the Viterbi decoder will be two bit output. When Input to the Convolution encoder is compared to the output of Viterbi decoder, results shows that they are the same. Convolution Encoder with Viterbi decoder finds applicable in digital broadcasting, Satellite applications, Digital Mobile Applications, Deep Space, Code Division Multiple Access and voice-band data communications.

Keywords- Convolution encoder, Viterbi decoder, Viterbi Algorithm, Verilog HDL, Soft decision decoding, Trace back method, Model Sim

INTRODUCTION

Convolution codes were invented in 1955 by P.Elias. Convolution codes are generally error correcting codes that are used to improve the performance of many digital systems such as digital radio, mobile phones and the Bluetooth implementations. Viterbi decoder together with its improved versions is one of the best applications of convolutional codes. Convolutional coding has been used in communication systems including deep space communications and wireless communications. It offers an alternative to block codes for transmission over a noisy channel. An advantage of convolutional coding is that it can be applied to a continuous data stream as well as to blocks of data. Convolution codes are also used in real time error correction to improve the performance of digital radio, mobile phones, satellite links etc. The simplicity and performance of convolution codes for Gaussian channel is very close to the accurate. They are one of the most widely used channel codes in the practical communication systems.

There are three alternative methods that are often used to describe the convolutional code. These are the tree diagram, state diagram and trellis diagram. There exist four basic convolutional codes decoding techniques: sequential, threshold, maximal-likelihood and the Viterbi algorithm. The sequential algorithm can provide very strong correcting capabilities while it needs relatively large memory, which strongly depends on communication channel error density. The threshold algorithm is extensively good for channels with mid to good signal to noise ratios (SNR). The Viterbi algorithm is an optimum decoding technique. It is optimum as it results in the minimum probability of error. It is also the relatively straight algorithm to implement in hardware and is the best decoding technique. Viterbi algorithm is a maximum likelihood algorithm and performs decoding, through searching the minimum cost path in a weighted oriented graph, called trellis. The basic building blocks of Viterbi decoder are branch metric unit (BMU), path metric unit (PMU), add compare and select unit (ACSU) and survivor memory management unit (SMU).

CONVOLUTION ENCODER

Convolution encoding is widely used for satellite and other noisy communications channels. There are two important components of a channel using Convolution encoding: the Convolution encoder (at the transmitter) and the Viterbi decoder (at the receiver). An encoder includes extra information in the transmitted signal to reduce the probability of errors in the received signal that may be corrupted by noise. Every two bits of data stream are encoded into three bits for transmission. The ratio of input to output information in an encoder is the rate of the encoder; this is a rate 2/3 encoder. The following equations relate the three encoder output bits (Yn2, Yn1, and Yn0) to the two encoder input bits (Xn1 and Xn0) at a time nT:

Yn2 = Xn1	(1.1)
Yn1 = Xn0 xor Df2	(1.2)
Yn0 = Df1	(1.3)



Figure 1. Convolution encoder of rate 2/3

VITERBI DECODER

The Viterbi decoder consists of three main units branch matrices unit, add control select unit and survivor management unit. The branch metric unit (BMU) is responsible for the computation of matrices, second block add compare select unit (ACSU) selects the survivor paths for each trellis state, third block survivor management unit (SMU) performs the selection of output which is based on the most optimum path metric. Figure shown below shows the general structure of Viterbi decoder



Figure2. Block diagram of Viterbi decoder

INTERNAL MODULES OF VITERBI DECODER

The Viterbi decoder consists of various internal modules for Branch metric calculation, Path metric calculation, modules to perform addition of branch metric and path metric, comparison module to select the smallest path and other modules. The first module is Euclidean distance calculation module. The other modules are subset decode module, Compute metric module, Compare Select module, Path in and Path memory modules, Reduce metric and Output decision module.



Figure3. Internal modules of Viterbi decoder

www.ijergs.org

Viterbi distance

The received signal (with noise) is converted into a series of distance measures from the known eight possible transmitted signals: The digitally encoded 3-bit signal, Y, from the encoder is converted directly to the distance measures. d[N] is the distance from signal = N to signal = 0 $d[N] = (2*\sin(N*PI/8))*2$ in 3-bit binary (on the scale 2=100)

Signal	Algebraic distance from signal 0	X =Distance from signal 0	Euclidean distance E = X ²	B = binary quantized value of E	D = decimal value of B	Quantization error Q = D - 1.75 E							
0	$2\sin(0\pi/8)$	0.00	0.00	000	0	0							
1	$2\sin(1\pi/8)$	0.77	0.59	001	1	-0.0325							
2	$2\sin(2\pi/8)$	1.41	2.00	100	4	0.5							
3	$2\sin(3\pi/8)$	1.85	3.41	110	6	0.0325							
4	$2\sin(4\pi/8)$	2.00	4.00	111	7	0							
5	$2\sin(5\pi/8)$	1.85	3.41	110	6	0.0325							
6	$2\sin(6\pi/8)$	1.41	2.00	100	4	0.5							
7	$2\sin(7\pi/8)$	0.77	0.59	001	1	-0.0325							

Table 1.1 Calculation of Euclidean distance

Module Subset decode

This module chooses the signal corresponding to the smallest of each set {||r-s0||**2, ||r-s4||**2}, {||r-s1||**2, ||rs5||**2}, {||r-s2||**2, ||r-s2||**2, ||r-s2||**2}. Therefore there are eight input signals and four output signals for the distance measures. The signals sout0, sout3 are used to control the path memory. The statement dff #(3) instantiates a vector array of 3 D flip-flops.

Module Compute metric

This module computes the sum of path memory and the distance for each path entering a state of the trellis. For the four states, there are two paths entering it; therefore eight sums are computed in this module. The path metrics and output sums are 5 bits wide. The output sum is bounded and should never be greater than 5 bits for a valid input signal. The overflow from the sum is the error output and indicates an invalid input signal.

Module Compare select

This module compares the summations from the Compute metric module and selects the metric and path with the lowest value. The output of this module is saved as the new path metric for each state. The ACS output signals are used to control the path memory of the decoder.

Module path

This is the basic unit for the path memory of the Viterbi decoder. It consists of four 3-bit D flip-flops in parallel. There is a 2:1 mux at each D flip-flop input. The statement dff #(12) instantiates a vector array of 12 flip-flops.

Module Path memory

This module consists of an array of memory elements (D flip-flops) that store and shift the path memory as new signals are added to the four paths (or four most likely sequences of signals). These module instantiates 11 instances of the path module.

Module Path in

This module determines the input signal to the path for each of the four paths. Control signals from the subset decoder and compare select modules are used to store the correct signal. The statement dff #(12) instantiates a vector array of 12 flip-flops.

Module metric

The registers created in this module (using D flip-flops) store the four path metrics. Each register is 5 bits wide. The statement dff #(5) instantiates a vector array of 5 flip-flops.

Module Output decision

This module decides the output signal based on the path that corresponds to the smallest metric. The control signal comes from the reduce module.

Module reduce

This module reduces the metrics after the addition and compares operations. This algorithm selects the smallest metric and subtracts it from all the other metrics

SIMULATION RESULTS OF CONVOLUTION ENCODER



Figure4. Simulation of convolutional encoder



Figure 5. Test bench of convolutional encoder

Vorkspace	10001		±##×	S F:/Ac	ademics/Project Do	Workspace			# # # X	S Filac	ademics/Protect Documents
Name	Statu	s Type Or	Modified	Ln#		* Name	Statu	s Type Or	Modified	Ln#	
CompareSelect.v Corv_Encoder.v D_Flp_Flop.v DesignOut.v Distances.v MetricCompute.v path0.v MreduceMetric.v SubsetDecoder.v Morebilecter.v WiterbiDec_test.v	*******	Verlog 0 Verlog 1 Verlog 3 Verlog 3 Verlog 4 Verlog 4 Verlog 7 Verlog 7 Verlog 7 Verlog 9 Verlog 9 Verlog 10	12/23/15 10:5 12/28/15 04:3 12/28/15 02:4 12/23/15 01:2 12/23/15 02:0 12/23/15 02:5 12/23/15 02:5 12/23/15 02:0 12/23/15 02:1 12/23/15 04:1 12/28/15 10:3 12/29/15 10:3	12345678901112314 1516	00 01 10 11 01 11 00 01 11 00 01 10 00 0	M compareSelect.v M Conv_Encoder.v M D_Flp_Flop.v M DecisionDut.v Distances.v M MetricCompute.v M path0.v M reduceMetric.v M SubsetDecode.v M TopVtDecoder.v W WterbiDec_test.v	***********	Verilog 0 Verilog 1 Verilog 2 Verilog 3 Verilog 4 Verilog 4 Verilog 7 Verilog 7 Verilog 7 Verilog 9 Verilog 10 Verilog 11	12/23/15 10:5 12/28/15 04:3 12/28/15 02:4 12/23/15 01:2 12/23/15 02:0 12/23/15 02:5 12/23/15 02:5 12/23/15 02:5 12/23/15 01:5 12/23/15 04:1 12/28/15 10:3 12/29/15 10:3	1234567890123456	000 010 101 100 011 111 100 001 000 111 011 011 010 010 101

Figure6. Input and Output of Convolution encoder of rate 2/3

Two bit inputs are provided to Convolution encoder. The inputs are taken as text document. Thus the convolution encoder of rate 2/3 is simulated successfully and three bit outputs are obtained which confirms that the simulation is successful. Test bench results prove the same. Two bit input and three bit output are visible on the test bench.

www.ijergs.org

SIMULATION RESULTS OF VITERBI DECODER

Messages																		
	100	000	010	101	100	011	111	100	001	000	111	011)000	101	010	010	101	Ţ
/viterbi_test/v_3/dis/dk	St1																	
/viterbi_test/v_3/dis/rst	St1																	
₽-♦ /viterbi_test/v_3/dis/dis0	111	000	(100	1110	1111	(110	001	111	001	000	001	J110	2000	[110	100	(100	1110	Υ
■-√ /viterbi_test/v_3/dis/dis1	110	001		111	110	(100	100	110	(000	001	(100	1100	001	(III)	001		011	
	100	100	000	110	100	001	110	100	001	100	110	001	(100	1110	000	(000	110	X
₽-⇒ /viterbi_test/v_3/dis/dis3	001	110	001	100	001	(000	111	001	(100	110	1111	000	(110	100	001	001	100	<u> </u>
	000	111	1100	001	000	001	110	000	(110	1112	110	001	111	001	100	100	001	1
	001	110		000	001	(100	100	001	[111	ĺ110	(100	100	(110	000	110		000	
E-⇒ /viterbi_test/v_3/dis/dis6	100	100	1111	001	100	1110	001	100	(110	100	001	1110	<u> 1100</u>	001	111	<u>(111</u>	001	
E-4 /viterbi_test/v_3/dis/dis7	110	001	1110	100	110	<u>[</u> 111	000	110	[100	001	000	[111	001	100	110	(110	100	<u> </u>
	011	111	101	1010	011	(100	000	011	(110	m	000	1100	1111	1010	101	101	010	
/viterbi_test/v_3/dis/Y2N	St1	_																
/viterbi_test/v_3/dis/Y1N	StD														-			
/viterbi_test/v_3/dis/YON	StD																	

Figure7. Calculation of Euclidean distance

Messages	2 3														_		
🐓 /vilanbi_test/v_3/sold/ch	5ti								وي الک								
Anterta_text/v_3/edd/rat	513	1	1000	Ville		14.44			1000	1 0000	1000	V- 10	1000 C	Vereit	View	1.10	1000
MINERS RELEVE JANGENO	HIM .	9000	1100	1110		11.10	1001	1111	,001	1000	1001	1110	1000	/1110	1100	1100	1110
Altero_test/v_vsdd/in1	110	\$001			1110	11,00	1100	1110	2000	1091	1100	1100	2001		1001		100
Aviteria_test/v_3/sdd/m2	100	(100	2000	1110	100	1001	1110	100	2001	1100	1110	1001	100	1110	000	1000	1110
/viterbi_test/v_3/sidd/n3	001	(110	1001	100	001	2000		001)100	1110	111	2000	1110	100	1001	1001	1100
Aviterb_test/v_3/sdd/vv4	000	- (111	1100	001	000	2001	1110	000	1110	- 101	1110	001	1111	2001	[100	100	1001
Avitarbi_text/v_3/edd/n5	001	010		000)001	1100	100	2001	1111	1110	1100	X100	1110	000	<u> 1110</u>		000
Image And Antipation (1998) Image Antipation (1998)	100	(100	Ittit	001	100	1110	1001	100	1110	100	001	Ĭ110	1100	001	(111	1111	001
Miterbi_test/V_3/sdd/in7	5.10	\$901	1110	1100	1110	1111	000	1110	100	001	000	йн I	1001	1.00	1110	1110	1100
And Anteria Link / Shidd/s0	000			(100)001	5000	1001		1000)001	000	001		X000	001	Xi00	
D dy /viterbi_test/v_3/sdd/s1	001	000	2001		(466	1001	.X100		201	1000	2001	1100		2001	2000	(201	
Initerbi_test/v_3/sdd/k2	100	000	1100	1000	2001	1100	1001		1100	2001	1100	2001		1100	2001	1000	
Anterts_text/v_3/edd/u3	001	000	1001		1100	2001	1000		201	1100	001	000		2001	1100	1001	
/viterts_test/v_3/sdd/scntr0	Sti																
/viterbi_test/v_3/sdd/scntr1	5t1																
/viterts_text/v_3/bdd/hcntr2	510							1									
/viterbs_text/v_3/sdd/scntr3	SH0							1			1						
	111			100	[110	1111	I110	1001	1111	1001	000	2001	1110	000	1110	1100	
Anterbi_text/v_3/edcl/indo I	110	000	1001		1111	1110	1100		1110	1000	001	1100		1001	1111	001	
D /vitertsi test/v_3/sdd/sub2	100		1100	000	1110	100	1001	J1 10	1100	1001	1100	Ĭ110	3001	1100	Y110	1000	
Alterta test/V 3/sdd/aub3	001	000	1110	001	1100	2001	1000	5111	3001	100	1110	111	Jood	1110	1100	3001	
Atterts test/v Shald/mab4	000	000	litt	Itoo	1001	500	loot	1110	boo.	1110	litt	1110	5001	Ditt	1001	Tino	
n d hiterts teethy 3/add/mab5	001		1110	1002	Vice	Vint	Tion.		201	1111	1110	1100		1110	Yada	Yi to	
Adviterts test/s Riskinghe	100	000	1100	TITLE.	1001	100	1110	1001	1100	1110	1100	Yoot	1110	100	Vint	YIII	
Anteris weth Shekilada?	110	000	6001	Tu io	Tion	1110	Titt	5000	YL 10	1100	001	000	- IIII	1001	1100	Yitio	
a second second	1000	100	- 20/			410			110						A4052		

Figure8. Subset Decode Module

Messages																		
🖪 👍 /viterbi_test/v_3/cmc/m_out0	00000	00000			00001	00010		00011			00100		00101		00100	00101)00000	5
iviterbi_test/v_3/cmc/m_out1	00100	00000				00010	00000	00011		00000	00100	00000	00101		00000	00101	00 100	j
	00101	00000		00001		00000	00010		00000	00100	00000	00101	00100	00000	00101	00000	00101	
₽-♦ /viterbi_test/v_3/amc/m_out3	00101	00000		00001			00010	100000	00011	00100		00101	00000	00100	00101	00100	00101	
	000	000		(100	001	000	001		000	001	000	001		000	001	100		
	001	000	001		(000	001	100		001	(000	001	(100		001	000	001		
E-4/viterbi_test/v_3/onc/s2	100	000	100	(000	001	100	001		100	001	100	001		100	001	000		
	001	000	001		(100	1001	000		001	(100	001	000		001	[100	001		
₽-� /viterbi_test/v_3/onc/p0_0	00000	00000		00100	00010	00010	00011	00100	00011	00100	00100	00101	00110	00101	00101	01001	00 100	
	01001	00000	00100	00001	00010	00100	00011		00100	00101	00100	00110	00101	00100	00110	00000	00101	
	00100	00000	00100	00000	00010	00110	00011	100100	00111	00100	01000	00101	00110	01001	00101	00101	00000	
	00101	00000		00101	00010	(00000	00011		00000	00101	00000	00110	00101	00000	00110	00100	01001	
	00101	00000	00001		00000	00011	00100	00111	00100	00000	00101	00100	01001	00110	00000	00110	00101	
	00110	00000	00001	00010	00101	00010	00010	00000	00 100	01000	00101	00101	00000	00101	01001	00101	00110	j
	00101	00000	00001		00100	00011	00000	00011	00100	00100	00101	(00000	00101	00110	00100	00110	00101	
• /viterbi_test/v_3/anc/p3_3	00110	00000	00001	00010	00001	00010	00110	00100	00100	00100	00101	01001	00100	00101	00101	00101	00110	
/viterbi_test/v_3/cmc/error	StO																	

Figure9. Compute Metrics Module

Messages																	
₽-� /viterbi_test/v_3/cs/p0_0	00000	00000		00100	00010	00010	00011	00100	00011	00100	00100	00101	00110	00101	00101	01001	(00100)
	01001	00000	00100	00001	00010	00100)00011		00100	00101	00100	00110	00101	00100	00110	(00000	00101
	00100	00000	00100	00000	00010	00110	00011	00100	00111	00100	01000	00101	00110	01001	00101	00101	() 00000
₽-� /viterbi_test/v_3/cs/p2_1	00101	00000		00101	00010	100000	00011		00000	00101	00000	00110	100101	00000	00110	00100	01001
	00101	00000	00001		00000	00011	00100	00111	00100	00000	00101	00100	01001	00110	00000	00110	00101
₽-� /viterbi_test/v_3/cs/p3_2	00110	00000	00001	00010	00101	00010	00010	00000	00100	01000	00101	00101	00000	00101	01001	00101	00110
₽-� /viterbi_test/v_3/cs/p1_3	00101	00000	00001		00100	00011	(00000	00011	00100	00100	00101	00000	00101	00110	00100	00110	00101
G-√ /viterbi_test/v_3/cs/p3_3	00110	00000	00001	00010	00001	00010	00110	00100	00 100	00100	00101	01001	00100	00101	00101	00101	00110
₽-� /viterbi_test/v_3/cs/out0	00000	00000		00001	00010	00010	00011		00011	00100	00100	00101		00100	00101	00000	00100
	00100	00000		00000	00010	00000	00011		00000	00100	00000	00101		100000	00101	00100	(00000
🖅 🎝 /viterbi_test/v_3/cs/out2	00101	00000	00001		00000	00010	00010	00000	00100	00000	00101	00100	00000	00101	00000	00101	(
	00101	00000	00001			00010	(00000)	00011	00 100	00100	00101	00000	00100	00101	00100	00101) (
🐓 /viterbi_test/v_3/cs/acs0	ទស					1											
/viterbi_test/v_3/cs/acs1	St0																
/viterbi_test/v_3/cs/acs2	St0																
/viterbi_test/v_3/cs/acs3	StD																

Figure10. Compare Select Module



Figure11. Path in Module

Messages	
🖬 🎻 /viterbi_test/v_3/pm/p0 011	000 1010 1101 1
Image: Anterbilitest/v_3/pm/p1 011	6000 1010 X101 X
	000 X101 X
Aviterbi_test/v_3/pm/p3 011	000 101 Juli X
	Baccorpio (2000 (po) (2000 (po) (2100 (po) (200 (po) (100 (po) (1000 (po) (100 (po) (100 (po) (210 (po)
/viterbi_test/v_3/pm/dk St1	ر ان ہو
/viterbi_test/v_3/pm/rst	
/viterbi_test/v_3/pm/acs0 St0	
/viberbs_text/v_3/pm/acx1 S10	نظلت ہے پر کا تن پر جی و کا تن پر کا تن ہے ہیں و کا کا برج کا تخط کا ہے و کا ت
/viterbr_test/v_3/pm/acs2 510	
/viterbi_test/v_3/pm/acs3 St0	
Avitesta_test/v_3/pm/aut 1 100 1000 100 10	- Goscodposecee 🔰 Seccedpia Sectedpa Jate 196
Miterbi_test/v_3/pm/out2 100100100100	000000000000 101000000
/viterbi_test/v_3/pm/out3 100100100100	000000000000 10100 1000 10010000 10110101 1011001 1011001 1011011 10101001 10010000 100100000 101111111 10110100 10
/viterbi_test/v_3/pm/out4 100.000.000.000	60000000000 10100100. 10010010 10110110 10110110 10110110 10110110 1010100 10010010 10010010 100000000
/viterbi_test/v_3/pm/out5 110110110110	000000000000 001000. 10000000. 10110100. 101101001 1011001 1011001 1011011 11011001 1001000 1001000 10000000 1
	000000000000 10100100110110111011011101101
	Bostoppo 101000000000
•	600000000000 100 100 110 110 1 100 100
Avterbi_test/v_3/pm/out9 101101101101	00000000000 B0100000 B0100100 B11011011 B11011001 B11011001 B11011001
/viterbi_tent/v_3/pm/out10 000000000000	6000000000 0000000 0000000 10100 (00 10110) 1 10000000
/viterbi_test/v_3/pm/out11 011011011011	

Figure12. Path Memory Module

ISSN 2091-2730																	
Messages																	
🐓 /viterbi_test/V_3/mc/clk	St1	اولا															
🐓 (viterbi_test/v_3/mc/rst	St1																
₽-∲ (viterbi_test/v_3/mc/m_in0	00000	00000		00001	00010	00010	00011		00011	00100	00100	00101		00100	00101	(00000	00100
₽-🍫 (viterbi_test/v_3/mc/m_in1	00100	00000		00000	00010	00000	00011		0000	00100	00000	00101		00000	00101	00100	00000
/viterbi_test/v_3/mc/m_in2	00101	00000	00001		00000	00010	00010	00000	00100	00000	00101)00100	(00000	00101	00000	00101	
/viterbi_test/v_3/mc/m_in3	00101	00000	00001		00001	00010	00000	00011	00100	00100	00101	00000	00100	00101	00100	00101	
Iviterbi_test/v_3/mc/m_out0	00000	00000			00001	00010		00011			00100		00101		00100	00101	(00000)
• /viterbi_test/v_3/mc/m_out1	00100	00000				00010	00000	00011		00000	00100	00000	00101		00000	00101	00100
/viterbi_test/v_3/mc/m_out2	00101	00000		00001		00000	00010		00000	00100	00000	00101	00100	00000	00101	(00000	00101
• /viterbi_test/v_3/mc/m_out3	00101	00000		00001			00010	(00000	00011	00100		00101	00000	00100	00101	00100	00101
					Fig	ure13	. Metri	ic Mod	ule								
Meccanec																	
					· · · ·	-	- v		.		y			V			
• /viterbi_test/v_3/rd/in0	00000	0000		00001	00010	00010	00011		100011	00100	100100	00101		00100	00101	100000	00100
• /viterbi_test/v_3/rd/n1	00100	0000	Y	(00000	100010	00000	00011	·	100000	00100	00000	00101		00000	00101	100100	00000
	00101	0000	00001		,0000	100010	00010	20000	200100	00000	00101	00100	00000	00101	00000	100101	
How And A state in the second state in the	00101	00000	100001	Vanna	600040	00010	00000	200011	00100	00100	100101	00000	00100	00101	00100	100101	Tanuna
Vitero_test/v_s/ro/m_nu	0000	0000		00001	00010	00010	00011		00011	00100	00100	00101		00100	00101	0000	00100
Interpretation of the second secon	00100	0000	GAAAAA	,0000	00010	0000	00011	60000	0000	00100	0000	00101	64444	0000	00101	100100	0000
• /viero_esuv_arom_nz	10101	0000	0001		0000	00010	00010	0000	100100	0000	00101	100100	0000	00101	0000	10101	
Alterio_cesqv_ajidpin_ina	00101	11 00		61	110	100010	111	710 Y10	10100	110	10101	10000	10	10101	10100	10101	61
Niterhi tect/v 3ird/cmallect	00000	0000		<u></u>	100000	Vincon	(11	110	100000	60000	01	111	(10	100000	100	100000	
R- A LuccoTerrel - Sholynomer	*****	0000			0000	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	0000		0	10000	0000	10000		ĮJUUUU		0000	
		-			Fi	oure1/	1 Red	uce Ma	dule								
					1.1	guiti	. Reu		June								
Maccanae																	
naaya																	
🖬 🖞 (viterbi test/v 3/do/o0	100	-000														2010	101
a that had block	100			_		=										la 44	
••• (vitero_test/v_s/oo/p1	100	000														1010	
🕵 🌖 (viterbi test/v 3/dojb2	100	000														1010	101
Alberti tertle 21da.h2	W	AMA														hin	
B-A luncin Testla "sholbs		W			1111				_		100					1010	
/viterbi_test/v_3/do/control	00	11 00		01	10	01	11	10	01	10	01	11	10	01	10	00	01
📢 hiterhi testiv 3ikolout	m	- 00														nia.	101
T A lumo "end i "dadage	110	W														(010	_(101

Figure15. Output Decision Module

www.ijergs.org

	Television (
	formers and 201	100		3122	101	100	1011	111	103	.021		m	011	000	2013	0.12	010	1221	1
0	Atteria Jest/v 34bat	1	- 02.															101	
	Anterio Justife 3th	5(1																	
	/cmarte_text/s_S/Vel	311	1																
	Petarla_tost/v_3/error	50																	
	Hants Jean A Anosti	911				1					1.								
	/itel/_tell/_3/kell	311																	
	Alleta_test/v_3/avit2	517																	-
	Ameria_meth_3/mut3	515					1.00												
04	/uthartis_texts/v_3/x0	000	035		1120	1021	1000	1001		1050	701	1000	1201	_	200	1901	1150		
01	Artante_bant/s_3/e1	der 1	- 630	001		000	1001	1100		001	2000	1.01	1100		201	800	051	-	
0	femaria_bash/s_3/62	300	020	1:00	800	100 t	1,000	1.01		Libo	2011	1100	D01		G100	\$101	2000		
	/etarte_leat/s_3/v3	001	035	001		Lino.	0001	\$200		2001	1900	201	600		201	LU00	101		
	Distantia test (1-3/10_010)	00000	00000		£0000 i	200010	200910	110011		222011	001100	E10,1005	Bourg		200300	100101	100000	100100	1
	Friterite 38800-344 JHL	99106	00000		00000	170010	000000	200011		00000	Do 100	20000	100101		20002	20101	200100	00000	
	/viteds_test/v_3/m_m2	00101	00000	00001		100000	1000 U	1000 10	5 5000	00100	1,0000	120101	25100	100000	20111	00000	100101		
	Autanta statului jota jara	02101	00000	000001		100001	1000010	\$10003	00011	00100	200100	1.0101	120000	001100	201011	20103	10100		
	/emeria_next/v_3/m_markit	3155281	- 00000			00001	000010		00011			1.01001		00101		8.0000	80101	0000	
	Americ hert/v_3ht.pat1	80100	00000				100010	00005	00011		100000	10005	101000	D0101		00000	00101	0.0100	
•	Enterte Astath 300 parts	001111			E0001		100000	1000 30		100000	100100	1.0002	100103	00100	20000	00101	00000	00101	
0	Antarta_Inst./usi/v_3/in_out3	07101:	03000		D0001	_		1000 10	\$20000	100011	200 100		00101	100000	20102	100101	200100	00101	
	/+meta_leat/+_3(pd_3)	00000	00000		\$20 LOT	\$300 LG)0000 L0	\$20011	(20100	110003	001100	20103	00101	00110	20101	20101	\$2100 i	(ac too	- 1
	/virents_text/v_3(p2_1)	01002	00000	00100	100001	\$10010	00100	110011		00100	01101	2001002	00100	100102	200 100	00132	\$20000	\$10101	
	formers_test/v_3/p0_1	00100	00000	001100	100001	\$1000LQ	00110	\$20011	\$301.00	610111	00100	01000	100101	60110	\$21001	60101	\$20101	00000	
	/viterts_text/v_3/in2_1	DEIDI	00000		00001	000010	00000	100011		100001	00101	.0000	00110	100103	.00025	80193	100100	01001	
•	Atterts_text/v_3(s1_7	00101	03300	00001		100000	100011	20102	00111	0.5100	100000	1.01015	00100	01001	20112	20001	00110	00101	
	/cherte_heat/v_3/st3_2	00110	0000	00001	00010	00101	000010	200055	05.000	00100	0:00	10:01	00101	0300	201011	£ 1001	80103	00110	
	Ameria Just Jo Shit J	00101	00000	00001		100100	192011	\$10005	00011	DO101	00100	1.0101	00000	00101	20110	100,000	00110	00101	
۲	/viterbi_test/v_3/p3_3	00110	03000	100001	(D0010	10000	303010	\$201.00	00100	00100	00100	10001	10010	00100	20101	100101	100101	100110	
and the second second	A CONTRACTOR OF	100	The second s	COLUMN STATE			THE OWNER WATER OF		THE OWNER OF TAXABLE PARTY.		TTTTTTTTTTT				CONTRACTOR OF	TAXABLE INC.	The second s		
1	Filan	3100 pm	s	205.0	5	+ 30.pt		6021	20	800 pt		1000 (1	1200.0		1400 p		1600 p	
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1																			

Figure16. Test bench of Viterbi decoder

		E TIAC	ademics/Project	workspace			- D E A	F:/Aca	ademics/Project Documents/Cl		
Name	Statu	s Type Or	Modified	Ln#		▼ Name	Status	Type C	n △ Modified	Ln#	
Name CompareSelect.v Conv_Encoder.v D_Flip_Flop.v DecisionOut.v Distances.v MetricCompute.v path0.v reduceMetric.v SubsetDecode.v TopVitDecoder.v viterbiDec_test.v	Statu V V V V V V V V V V V V	s Type Or Verlog 0 Verlog 2 Verlog 3 Verlog 4 Verlog 5 Verlog 6 Verlog 7 Verlog 8 Verlog 9 Verlog 10 Verlog 11	Modified 12/23/15 10:5 12/28/15 02:4 12/28/15 02:4 12/23/15 01:2 12/23/15 02:5 12/23/15 02:5 12/23/15 02:5 12/23/15 02:5 12/23/15 01:5 12/23/15 01:5 12/23/15 04:1 12/28/15 10:3 12/29/15 10:3	En# 1 2 3 4 5 6 7 8 9 10 11 12 13 14	0000 010 101 100 001 111 111 100 001 000 111 000 101 010	Violicipade Violicipade Violicipade Violicipade Conv_Encoder.v D_Filp_Flop.v DecisionOut.v Distances.v MetricCompute.v path0.v reduceMetric.v SubsetDecode.v TopVitDecoder.v viterbi_test.v ViterbiDec_test.v	Status V V V V V V V V V	Type C Verlog 0 Verlog 1 Verlog 2 Verlog 3 Verlog 4 Verlog 5 Verlog 5 Verlog 7 Verlog 8 Verlog 9 Verlog 1 Verlog 1	± b ± b ≥ 12/23/15 10:5 12/28/15 04:3 12/28/15 02:4 12/23/15 02:4 12/23/15 01:2 12/23/15 01:2 12/23/15 02:5 12/23/15 02:5 12/23/15 02:0 12/23/15 02:0 12/23/15 01:5 12/23/15 01:5 12/23/15 01:5 12/23/15 01:5 12/23/15 01:5 12/23/15 01:3 1 12/29/15 10:3 1	★ F:Acc Ln# 1 2 3 4 5 6 7 8 9 10 11 12 13 14	ademics, Project Documents, (CX 00 01 10 11 01 10 11 00 01 11 00 01 11 00 01 11 00 01 11 00 01 01
				14 15 16	010					15 16	01 01

Figure17. Input and Output of Viterbi decoder

CONCLUSION

A Viterbi algorithm based on the strongly connected trellis decoding of binary convolutional codes has been presented. The use of errorcorrecting codes has proven to be an effective way to overcome data corruption in digital communication channels. The Viterbi decoder is modelled using Verilog, and Simulated by Xilinx ISE. We can implement a higher performance Viterbi decoder with such an algorithm. So in the future, with this algorithm with larger code rates we can get better results.

REFERENCES:

- [1] Mandwale A. J and Mulani A O" Different Approaches For Implementation of Viterbi decoder on reconfigurable platform"International Conference on Pervasive Computing (ICPC) 2015
- [2] P R Pachlegaonkar, S S Patki "Implementation of Reconfigurable Convolutional Encoder and optimum Adaptive Viterbi Decoder with Multi booting and Error Detection on FPGA", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 3 Issue 10, October- 2014
- [3] Mahe Jabeen, Salma Khan," Design of Convolution Encoder and Reconfigurable Viterbi Decoder" International Journal of Engineering and Science ISSN: 2278-4721, Vol. 1, Issue 3 (Sept 2012), PP 15-21
- [4] Yan Sun, Zhizhong Ding "FPGA Design and Implementation of a Convolutional Encoder and a Viterbi Decoder Based on 802.11a for OFDM", Wireless Engineering and Technology, 2012, 3, 125-131.

- [5] Karim, M.U., Khan, M.U.K. and Khawaja, Y.M. "An area reduced, speed optimized implementation of Viterbi decoder", International Conference on Computer Networks and Information Technology (ICCNIT), pp. 93-98, 2011
- [6] Guan, M. and Yang, M. "Comparison and design of decoder in B3G Mobile communication system", Communications and Network, pp. 20-24, 2009
- [7] Arun and V. Rajamani, "DESIGN AND VLSI IMPLEMENTATION OF A LOW PROBABILITY OF ERROR VITERBI DECODER" IEEE, First International Conference on Emerging Trends in Engineering and Technology, 2008
- [8] Russell Tessier, Sriram Swaminathan, Ramaswamy, Dennis Goeckel and Wayne Burleson. "A Reconfigurable, power-efficient adaptive Viterbi decoder", IEEE Transactions on Very Large Scale Integration (VLSI) systems, Vol. 13, No. 4, pp. 484-488, 2005
- [9] Engling yeo, Augsburger, S.A., Rhett Davis, W. and Borivoje Nikolic . "A 500-Mb/s soft-output Viterbi decoder", IEEE Journal of solid-state circuits, Vol. 38, No. 7, pp. 1234-1241,2003
- [10] M. Kivioja, J. Isoaho and L.Vanska, "Design and implementation of Viterbi decoder with FPGAs," Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology, Kluwer Academic Publishers, vol. 21, no. 1, pp. 5-14, May 1999.
- [11] C. Berrou, P. Adde, E. Angui and S. Faudeil "A Low Complexity Soft-Output Viterbi Decoder Architecture," *IEEE Int. Conf. Communications*, vol. 2, pp. 737–740, May. 1993
- [12] Viterbi.A.J, "Convolution codes and their performance in communication systems," *IEEE Transaction on Communications*, vol.com-19, pp. 751 to 771, October 1971
- [13] Samirkumar Ranpara, "On a Viterbi decoder design for low power dissipation," towards his master's thesis submitted to virginia polytechnic institute and state university.
- [14] B. Sklar, "Digital Communications, Fundamentals and Applications", Second Edition, New Delhi, Pearson Education, 2004
- [15] Chip fleming, "A tutorial on Convolutional coding with viterbi decoding".
- [16] Samir Palnitkar, "Verilog HDL: A Guide to Digital Design And Synthesis, Second Edition", Prentice Hall PTR
- [17] Text book by Simon Haykin and Michael Moher : Modern Wireless Communications; Pearson Prentice Hall, 2005
- [18] Text book on Applied Coding and Information Theory for Engineers by Richard B. Wells, Prentice Hall, Information and system science series