

# An Efficient High Speed Convolution Encoder and Viterbi Decoder

Chandan Nagraj<sup>1</sup>, S Muralinarasimham<sup>2</sup>

<sup>1</sup>M. Tech Scholar (VLSI Design & Embedded Systems) Bangalore Institute of Technology, Bangalore, Karnataka, India

<sup>2</sup>Associate Professor, Dept of ECE Bangalore Institute of Technology, Bangalore, Karnataka, India

**Abstract:** *The main aim of the project is execution of convolution encoder and Viterbi decoder by applying HDL (Verilog) coding. The overall objective is to clearly understand in depth the convolution encoder and Viterbi Decoder model, to evaluate the basic functionalities and steps involved in the process, through HDL code and finally to critically analyze the results obtained through HDL code. The main purpose of this study is to yield the gains obtained by the developers with the usage of convolution encoder and Viterbi decoder. Here, we make use of HDL code in order to implement the both encoder and decoder in a simpler way so that we can achieve faster operation. Convolutional codes are non-blocking codes that can be designed to either error detecting or correcting. Convolution coding has been used in communication systems including deep space communication and wireless communication. At the receiver end the original message sequence is obtained from the received data using Viterbi decoder, by making use of trellis diagram. This project helps the students related to the communications and also it helps the people who are in the field of encoders and decoders as it is one of the efficient methods for reducing the errors while communication procedure is in advance.*

**Keywords:** HDL, Verilog, Convolution encoder, Viterbi decoder, trellis diagram

## 1. Introduction

In the present days, data transmission between the systems play a vital role with increasing technologies day-by-day, the number of users is simultaneously increasing. This wide usage leads to major issues in the digital communication systems and results in data corruptions. It's very necessary for the telecommunication to reduce the data corruption by providing a suitable solution to the errors occurred in the communication process. One such method that decodes the process by simultaneously correcting the process effectively is Viterbi decoder. For decoding the convolution codes Viterbi algorithm is the highest recognizable algorithm. This algorithm may be described with RTL implementation and simulation. To engage well organized communications an efficient data is presented by the digital systems. Data corruptions are the important issue confronted by the digital communication systems. To decrease data corruptions error correcting codes is a best technique. Previously the error correction was not done properly by the decoder, hence we implemented the Viterbi decoder. The Viterbi decoder decodes the data efficiently with error correction.

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. IS-95, a wireless digital cellular standard for CDMA (code division multiple access), employs convolutional coding. A third generation wireless cellular standard, under preparation, plans to adopt turbo coding, which stems from convolutional coding. The Viterbi decoding algorithm, proposed by Viterbi, is a decoding process for convolutional codes.

The algorithm can be applied to a host of problems encountered in the design of communication systems. The Viterbi decoding algorithm provides both a maximum likelihood and a maximum a posteriori algorithm. A maximum posteriori algorithm identifies a code word that maximizes the conditional probability of the decoded code word against the received code word; in contrast a maximum likelihood algorithm identifies a code word that maximizes the conditional probability of the received code word against the decoded code word. The two algorithms give the same results when the source information has a uniform distribution,

Two major techniques, clock gating and toggle Filtering, were investigated in this project. Hardware description language called Verilog HDL is used to valuate this project, which is one of the hardware descriptive languages that stand for Verilog Hardware Description Language. This project makes use of two main tools namely ISE-simulation and Xilinx-ISE-synthesis for successfully reaching the objectives.

The probability of error can be reduced by transmitting more bits than needed to represent the information being sent, and convolving each bit with neighboring bits so that if one transmitted bit got corrupted, enough information is carried by the neighboring bits to estimate what the corrupted bit was. This approach of transforming a number of information bits into a larger number of transmitted bits is called channel coding, and the particular approach of convolving the bits to distribute the information is referred to as convolution coding.

## 2. Convolution Encoder

Encoding of convolutional codes can be achieved using simple registers. In convolutional encoder, the message stream continuously runs through the encoder unlike in the block coding schemes where the message is first divided into long blocks and then encoded. Thus the convolutional encoder requires very little buffering and storage hardware. In this paper for a convolutional encoder, the following notations are used.

**c** = number of output bits.

**x** = number of input bits entering at a time.

**m** = number of stages of shift register.

**L** = number of bits in a message sequence.

**j** = number of modulo 2 adders.

**constraint Length: K** = (m + 1) digits.

**Bit Rate: r** = x / c

To convolutionally encode data, start with k memory registers, each holding 1 input bit. Unless otherwise specified, all memory registers start with a value of 0. The encoder has n modulo-2 adders ( a modulo 2 adder can be implemented with a single Boolean XOR gate, where the logic is:  $0+0=0$ ,  $0+1=1$ ,  $1+0=1$ ,  $1+1=0$ ), and n generator polynomials — one for each adder (see figure below). An input bit  $X=d_n$  is fed into the leftmost register. Using the generator polynomials and the existing values in the remaining registers, the encoder outputs n bits. Now bit shift all register values to the right (  $d_n$  moves to  $d_0$ ,  $d_0$  moves to  $d-1$ ) and wait for the next input bit. If there are no remaining input bits, the encoder continues output until all registers have returned to the zero state.

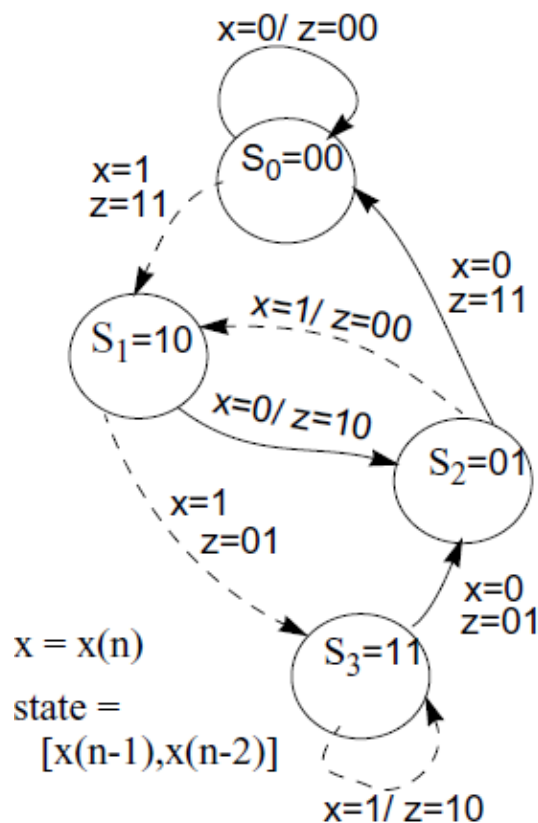


Figure 1: Convolution encoder with output c1, c2.

To generate the output, the encoder uses 7 values of the input signal (1 present input bit and 6 previous input bits). The set of values of input data in the shift register is called a state. The number of input data values used to generate the code is called the constraint length. Each set of outputs is generated by XOR ing a pattern of current and shifted values of input data.

The output  $C1 = d(n) \oplus d(n-1) \oplus d(n-2) \oplus d(n-3)$ . Here  $d(n)$  is the present input bit,  $d(n-1)$  was the previous (yesterdays) bit, etc.

The output  $C2 = d(n) \oplus d(n-2) \oplus d(n-3) \oplus d(n-5)$ .

### a) The Encoder as a Finite-State Machine

The convolution encoder can be described as a Mealy machine. The state is the two bits in the shift register. Let the first input bit to the shift register be  $x(n) = 1$ , and let the flip-flops be reset to zero so  $x(n-1)=x(n-2)=0$ . Then:

State= 00 =  $S_0 = [x(n-1), x(n-2)]$

Output  $z=[z1, z2]$

$z1 = x(n) \oplus x(n-1) \oplus x(n-2) = 1 \oplus 0 \oplus 0 = 1$

$z2 = x(n) \oplus x(n-2) = 1 \oplus 0 = 1$

$z=[z1, z2]= 11$

After the clock, state bit  $x(n-1)=0$  will shift right into  $x(n-2)$ , the input  $x(n)=1$  will shift right into  $x(n-1)$ , and the next state will be 10 =  $S_1$

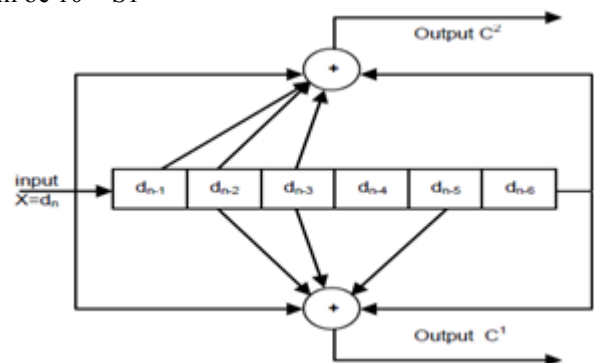


Figure 2: State Diagram for 2 state encoder.

The figure 2 shows the state diagram for two state convolution encoder, since the length of the state diagram grows for higher length encoder, thus for a 7 state convolution encoder state diagram grow, hence here it is shown for 2state convolution encoder.

As the length of the convolution increases the design becomes complex, and the time for simulation as well increases. Hence it is very much required to reduce the execution time by making the code much simpler with the same properties of the existing design. Hence we go for the RTL implementation of the convolution encoder by using less number of instructions coded in Verilog. By doing this a very less execution time is required there by it directly increases the speed of communication.

### 3. Viterbi Decoder

The Viterbi decoder receives the sequence of the data from the encoder and it is required to determine the original sequence of data transmitted from the convolution encoder. The process of identifying original message sequence from the received data can be done using the diagram called "trellis". A Viterbi decoder uses the Viterbi algorithm for decoding a bit stream that has been encoded using Forward error correction based on a convolutional code. Figure 4 shows the block diagram of Viterbi decoder.

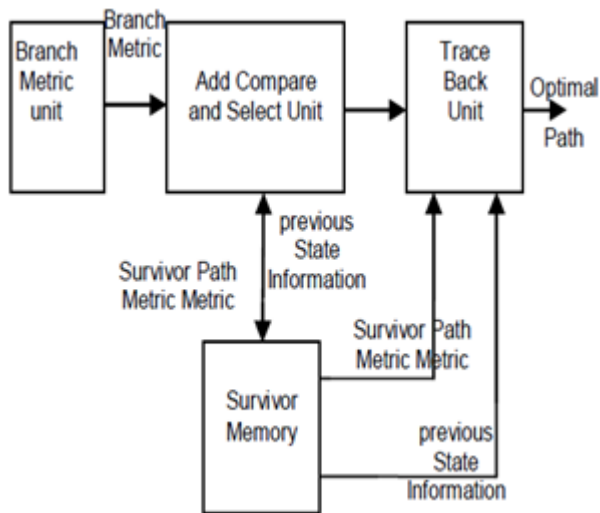


Figure 3: Block diagram of Viterbi decoder.

The Viterbi decoder consists of following units and they are explained in detail.

- Branch Metric Unit (BMU)
- Add Compare and Select Unit (ACS)
- Survivor Memory Unit
- Trace Back Unit (TBU).

#### A.Branch Metric Unit

The two binary bits are parallel given as the input to the Viterbi decoder and then the hamming distance computation module (BMU) calculates sixty four set of hamming distance. Each current state can be reached by two possible paths, hence each set consists of two values.

#### B.Add Compare and Select Unit

Cumulative hamming distance till the last branch is added to hamming distance of the new branches by ACS module. After adding operation each current state gets two new cumulative hamming distances, now ACS module compares the size of the two cumulative distances and selects the smaller one as a survivor. The smaller cumulative hamming distance becomes the benchmark for the next computation. Survivor paths of all the sixty four states are stored in RAM blocks of Survivor memory unit.

#### C.Survivor Memory Unit

Sixty four survivor paths are stored in the RAM blocks at each stage. When there are no more encoded bits to process the detection of a node having minimum path metric is done by comparing all the sixty four cumulative hamming distances at the last stage.

#### D.Trace Back Unit

Using the minimum path metric of last stage the Trace back Unit starts back tracing survivor paths which are stored in the memory unit. According to survivor path values the original transmitted message is determined.

### 4. Trellis Diagram

The original message sequence is encoded following a path which has to be determined at the decoder part of receiver to get the original message sequence from the encoded data. For the representation of this path a 'trellis diagram' is used. An example of trellis structure for  $K=3$  and  $r=1/2$  is given below in figure 4.

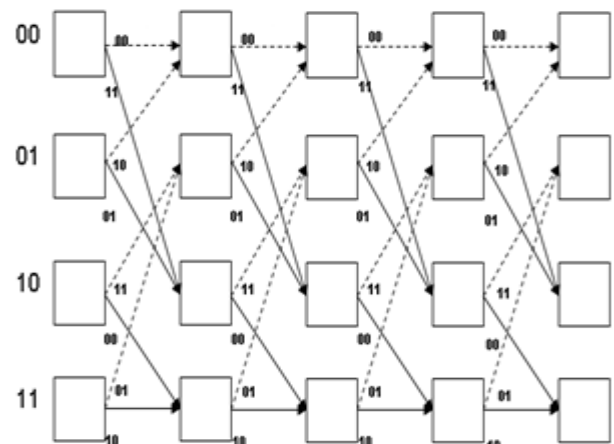


Figure 4: Trellis diagram for  $K = 3$  and  $r = 1/2$

### 5. Experimental Results and Discussion

The input values for convolution encoder are specified in the developed code. Providing input every time for the developed design is time taking process. So, the input values are included directly in the developed code, so that simulation process can be executed directly. The convolution encoder input values given in this project are:

```

0110100110010110100101100110100110010110011010
0101101001100101101001011001101001011010011001
0110011010
  
```

Each input value will be processed and a corresponding output will be provided for the Convolution encoder. The output values will be specified in the form of wave forms in both Xilinx and active HDL simulation environments.

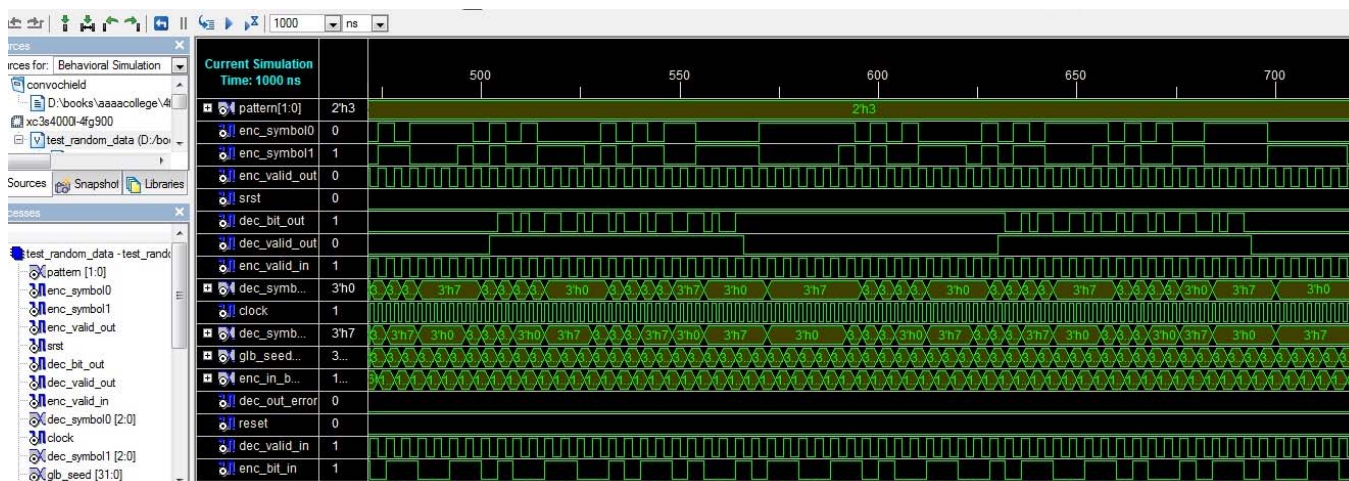


Figure 5: Output waveform of convolution encoder and Viterbi decoder

### A. Analysis of simulation using XILINX

When clock signal is applied to the Viterbi decoder reset button is set as 0, and the system reset button kept as 0, the cyclic encoded data starts after 100ns. by applying the valid encoded pulse train to the decoder enc\_symbol 0 and enc\_symbol 1 varies according to the periodic pulse train. According to the enc\_symbol the encoded output bit also a pulse train. The dec\_symbol0 consists an 3'h0 error bits at decoder process up to 100ns, after 100ns it decodes 3bits at a time changes continuously every 50ns. dec\_valid\_in is a continuous pulse train when dec\_symbol is change their bit stream every after 100ns. thus we get dec\_out bit and dec\_valid\_out bits same as 1.

The encoder input, output, decoder output, are written into the buffers and then stored in a text file, enc\_in.txt, enc\_out.txt and decoder\_out.txt respectively, so that we can verify the output and can also conclude that there are no errors, and can ensure successful completion of simulation.

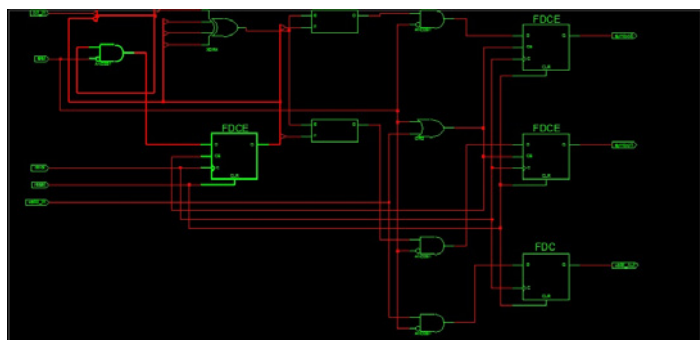


Figure 6: RTL Structure of convolution encoder.

The convolution encoder is designed by using less number of flip-flops and gates hence the operation is faster compared to the normal design the design summary of encoder is shown in figure 7.

Project File:	convochield.iise	Current State:	Synthesized
Module Name:	decoder	Errors:	No Errors
Target Device:	xc3s4000l-4fg900	Warnings:	2 Warnings
Product Version:	ISE 9.2i	Updated:	Wed Jun 4 09:00:46 2014

CONVOCHIELD Partition Summary	
No partition information was found.	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	6	27648	0%
Number of Slice Flip Flops	8	55296	0%
Number of 4 input LUTs	11	55296	0%
Number of bonded IOBs	8	633	1%
Number of GCLKs	1	8	12%

Figure 7: Design summary of the convolution encoder.

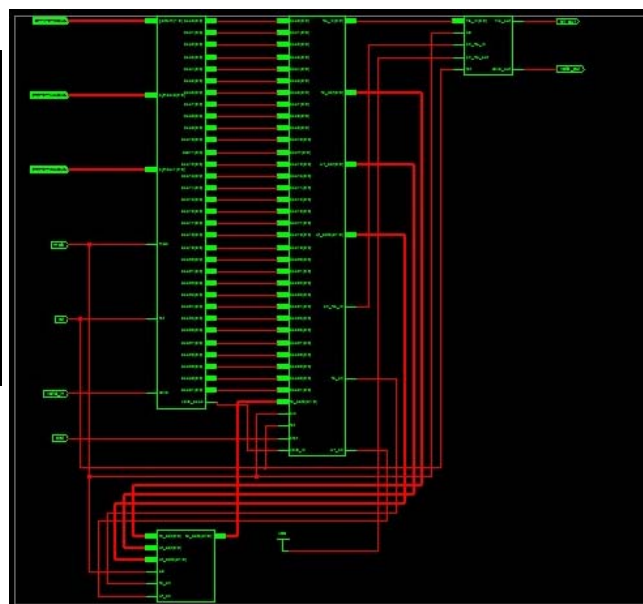


Figure 8: RTL design of Viterbi decoder.

Similarly the design summary has been designed for Viterbi decoder and it is observed that the numbers of gate counts are less and even the numbers of flip-flops used are less. The

figure 8, 9, shows the RTL design and design summary respectively.

CONVOCHIELD Project Status			
Project File:	convochield.isc	Current State:	Synthesized
Module Name:	decoder	• Errors:	No Errors
Target Device:	xc3s4000I-4fg900	• Warnings:	<a href="#">9 Warnings</a>
Product Version:	ISE 9.2i	• Updated:	Wed Jun 4 09:05:18 2014

CONVOCHIELD Partition Summary	
No partition information was found.	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	1015	27648	3%
Number of Slice Flip Flops	703	55296	1%
Number of 4 input LUTs	1903	55296	3%
Number of bonded IOBs	14	633	2%
Number of BRAMs	2	96	2%
Number of GCLKs	1	8	12%

Figure 9: Design summary of Viterbi decoder.

## 6. Conclusion

The implementation of the convolution encoder with other decoders utilizes maximum memory, and computational resource and makes utilization of more power. The implementation part of Viterbi decoder includes the simulation, pipelining, decoding and the interleaving procedures. As Viterbi algorithm is conceived more interesting and challenging for this project topic, it is considered, and also it has wide variety of applications in digital communications field. This project helps to generate more efficient decoder. Hence Viterbi decoder uses less memory.

## Reference

- [1] P. P. Chu (2006) RTL hardware design using VHDL: coding for efficiency, portability, and scalability. USA: John Wiley and Sons
- [2] K. Kawazoe, S. Honda, S. Kubota and S. Kato, "Ultra-high-speed and universal coding-rate Viterbi decoder VLSIC," Proc. ICC'93, pp. 1434-1438, May 1999.
- [3] S. Kubota, K. Ohtani and S. Kato, "A high-speed and highcoding-gain Viterbi decoder with low power consumption employing SST (scarce state transition) scheme," Electron. Lett., 22 (9), pp. 491-493, 2006.
- [4] L. Lang; C.Y. Tsui and R.S. Cheng, "Low power soft output Viterbi decoder scheme for turbo code decoding," Conference-paper, ISCAS '97(Cat. No97CH35987). IEEE, New York, NY, USA, 4 vol. Lxvi+2832 pp. p. 1369-1372 vol.2, 1997.
- [5] HEMA.S, SURESH BABU.V, RAMESH P "FPGA Implementation of Viterbi Decoder", Proceedings of the 6th WSEAS Int. Conf. on Electronics, Hardware, Wireless and Optical Communications, Corfu Island, Greece, February 16-19, 2007.

- [6] DR. Anubhuti Khare, Manish Saxena, Jagdish Patel "FPGA Implementation of Viterbi Decoder.
- [7] "Error Detection and Correction" at [www.mathworks.in](http://www.mathworks.in), 2013.
- [8] Sherif Welsen Shaker, Salwa Hussein Elramly, Khaled Ali Shehata "FPGA Implementation of a Reconfigurable Viterbi Decoder for Wimax Receiver", International Conference on Microelectronics, 2013.

## Author Profile



**Chandan Nagraj** received his B.E degree from SJBIT, Visweswarayya Technological University (VTU) in Electronics and Communication in the year 2012. Presently he is pursuing final year M. Tech from Bangalore Institute of Technology in VLSI Design and Embedded Systems under VTU in 2014. Currently I am working as Project trainee at Wipro GE Health Care Pvt. Ltd. Bangalore.



**S Muralinarasimham** is working as Associate professor at Bangalore institute of Technology Bangalore in the department of Electronics and communication. I would like to express my sincere gratitude to Mr. S Muralinarasimham for his guidance in the course of my research. I am also grateful to Mr. Arun Ramakrishna for his generous discussion and help.