International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 10, October- 2014

Implementation of Reconfigurable Convolutional Encoder and optimum AdaptiveViterbi Decoder with Multibooting and Error Detection on FPGA

Prashant Renukadas Pachlegaonkar[1]
Manoj Annashaheb Jagtap[2]
Electronics and Telecommunication Department,
Maharastra Institute of Technology,
Aurangabad,India

Ms. S. S. Patki
Electronics and Telecommunication Department,
Maharastra Institute of Technology,
Aurangabad,India

Abstract*—The main goal of this paper was resource-optimized implementation of the decoder on the target platform. It is well known that data transmissions over wireless channels are affected by attenuation, distortion, interference and noise, which affect the receiver's ability to receive correct information. Convolutional encoding with Viterbi decoding is a powerful method for forward error correction. Convolutional encoders and viterbi decoders are play important role in digital communication especially, When channel is noisy and introduces errors in transmitted signal, Wireless communication systems such as the third generation (3G) mobile system, DVB, Hiperlan 3GPPLTE, IEEE 802.11a WLAN, IEEE 802.26 intelsat IES-308/309 utilize some formulation of convolutional encoding usually decoded via viterbi decoder using multiple booting technique is designed. Today's data reconstruction in digital communication systems requires designs of highest throughput rate. Look ahead technique is studied for extracting vectorized output bits without taking into consideration the hardware cost involved. It improves the throughput rate. Implementation parameters for the decoder have been determined through simulation and the decoder has been implemented on a Xilinx FPGA SPARTAN 3E Kit. VHDL language is used as a design entry. The architecture can be reconfigured to decode a range of convolutionally encoded data with ½ rate, constraints lengths varying from 2 to 8 and various generator polynomials. In the starter kit mentioned above, two designs are implemented on the flash memory using the multiple booting techniques, the convolutional encoder and the viterbi decoder. The FPGA is configured with the specified design depending on the loaded program from the parallel NOR PROM flash memory. Also, configurability allows use of same hardware to satisfy needs of channel. And also error correction and detection, With this way of configuration, the FPGA itself can operates as a convolutional encoder or viterbi decoder of variable constraint length depending on SNR conditions of received signal.

Keywords— Convolutional Encoder; hard Decision; Viterbi Decoder; FPGA; Adaptive Decoder; Dynamic Reconfiguration.

I. INTRODUCTION

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. Convolutional codes [2] are a type of Error Correcting Codes (ECC) widely used in channel coding since the late 1960's [3]. All communication channels are subject to additive white Gaussian noise (AWGN) around the environment. The block codes can be applied only for the block of data whereas convolution coding has can be applied to a continuous data stream as well as block of data. They are preferred for their powerful correcting capability with high speed at low cost compared with their competing block codes [4]. Other types include error detecting codes, which only detect errors and request for retransmission; however, these types are more complex and expensive, as they require a two-way transmission system. 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) [5,1]. Viterbi decoding was proposed in 1967 by Andrew, J,Viterbi, the founder of Qualcomm Corporation. Evaluation process of convolutional encoder and decoder (CODEC), for the various constraint lengths and for different generator polynomials is discussed considering only few bytes of data [9]. The performance is analysed for the burst errors and distributed errors. Impact of constrain length for the different input images for SOVA algorithm is presented in [9]. A high performance generic soft input hard output Viterbi decoder is presented [10] and prototyped on an FPGA board. The presented Viterbi decoder is intended to be used in a complete wireless LAN transceiver prototype. Viterbi decoding method uses the Maximum Likelihood Decoding
(MLD) algorithm, this method find a most likely pattern from the received data, and is known as the most optimum decoding method. In the proposed work the code rate½ is used. The performance of Viterbi decoders are analyzed for the different constraint lengths (CL) as well as for different generator polynomials. To implement convolutional encoder and viterbi decoder we use Xillinx SPARTEN 3E FPGA kit.

A. Motivation of the Paper

Recently, convolutional codes have become more and more important in digital transmission. A convolutional code with viterbi decoding is used in wireless communication and

satellite communication. The various examples are cellular phone i.e. GSM IS-54,IS-95 CDMA standerd, Most of the Viterbi decoders in the market are a parameterizable intelligent property(IP) core with an efficient algorithm for decoding of one convolutionally encoded sequence only. In addition, the cost for the Convolution Encoder and Viterbi decoder are expensive for a specified design because of the patent issue[2].Therefore, to realize an adaptive Convolution Encoder and Viterbi decoder on FPGA is very demanding. Complexity of Viterbi decoding algorithm increases in terms of trellis length. Increasing the trellis length causes the algorithm to take more time to decode. This will cause transmission speed lower but make the transmission more reliable. Reducing the trellis length will increase the transmission speed.

II. SURVEY

A. History of Codes

The pioneering work on coding and coded waveform for digital communication was done by Shannon (1948), Hamming (1950) and Golay (1949). These works were followed with papers on code performance by Gilbert (1952), new codes by Muller (1954) and Reed (1954), and coding techniques for noisy channels by Elias (1954,1955) and Slepian (1956 Bupesh Pandita and Subir K Roy (1998) described the design and implementation of Viterbi decoder using FPGA for constraint length 0 to 8. Erik Paaske and Jakobnm Dahl Andersen (1998) proposed high speed Viterbi decoder architecture. M. Kivioja, J. Isoaho and L.Vanska (1999) presented the implementation of Viterbi algorithm on FPGA. Speed performance, easy rout ability and minimization of inter chip connections were main design criteria. Yun-Nan Chang, Hiroshi Suzuki, and Keshab K.Parhi (2000) proposed low power bit serial Viterbi decoder architecture. YOU yu-xin, WANG Jin-xiang, LAI Feng-changand YE Yi-zheng (2002) proposed VLSI design and implementation of high speed Viterbi decoder. Low power dissipation, modified T-algorithm and modified trace back methods were main considerations .Dalia A. El-Dib and M. I. Elmasry (2005) proposed memory less Viterbi decoder.This implements the pointer concept with modified register exchange method and bit serial architecture. The convolutional Encoder and viterbi Decoder used in the digital communications system is shown below.

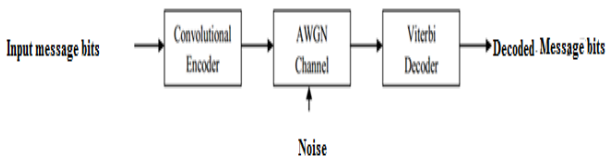

Fig 2.1 Digital Communication System

B. *Convolution Encoder*

The convolutional encoder is basically a finite state machine and is a linear system. A binary convolutional encoder can be represented as a shift register. The outputs of the encoder are modulo 2 sums of the values in the certain register's cells. The input to the encoder is either the unencoded sequence (for non-recursive codes) or the unencoded sequence added with the values of some register's cells (for recursive codes).Convolutional codes can be systematic and nonsystematic.

The k bit input is fed to the constraint length K shift register and the n outputs are calculated from the generator polynomials by the modulo-2 addition. The generator polynomial specifies the connections of the encoder to the modulo-2 adder. The 1 in the generator polynomial indicates the connections and zero indicates no connections between the stage and the modulo 2 adder. The figure below illustrates a simple convolutional coder with k=1,K=3, n=3, g1 (n) = (1 0 1), g2 (n) = (1 1 1), g3 (n) = (0 1 1) and R=1/2

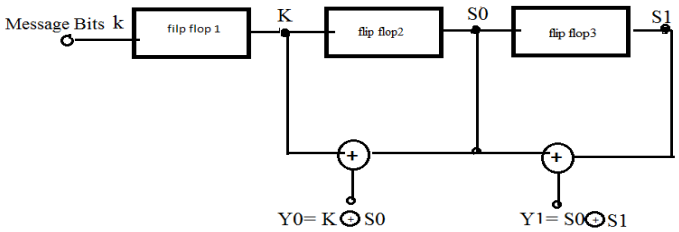

Fig 2.2 Convolutional Encoder

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. State diagram shows the transition between different states. Trellis diagram is the description of state diagram of the encoder by a time line i.e. to represent each time unit with a separate state diagram [10].

Table I- State Table of Convolutional Encoder

Input u	Present state (S_1,S_0)	Next state (S_1,S_0)	Output (v_1,v_2)
0	0 0	0 0	0 0
1	0 0	0 1	1 0
0	0 1	1 0	0 1
1	0 1	1 1	1 1
0	1 0	1 1	1 1
1	1 0	1 0	0 1
0	1 1	0 1	1 0
1	1 1	0 0	0 0

Trellis diagrams are messy but generally preferred over both the tree and the state diagrams because they represent linear time sequencing of events. The x-axis is discrete time and all possible states are shown on the yaxis. We move horizontally through the trellis with the passage of time. Each transition means new bits have arrived.The trellis diagram is drawn by lining up all the possible states (2L) in the vertical axis. Then we connect each state to the next state by the allowable codeword's for that state. There are only two choices possible at each state. These are determined by the arrival of

either a 0 or a 1 bit. The arrows show the input bit and the output bits are shown in parentheses. The arrows going upwards represent a 0 bit and going downwards represent a 1 bit. The trellis diagram
is unique to each code, same as both the state and tree diagrams are. We can draw the trellis for as many periods as we want. Each period repeats the possible transitions.We always begin at state 00. Starting from here, the trellis expands and in L bits becomes fully populated such that all transitions are possible. The transitions then repeat from this point on. The output of each transition is written on the line within brackets as shown. The state transitions for a given '1' input are denoted by dotted lines while state transitions for a given '0' input are denoted by solid lines. A trellis diagram featuring the present state and next state values is shown in the figure2.3

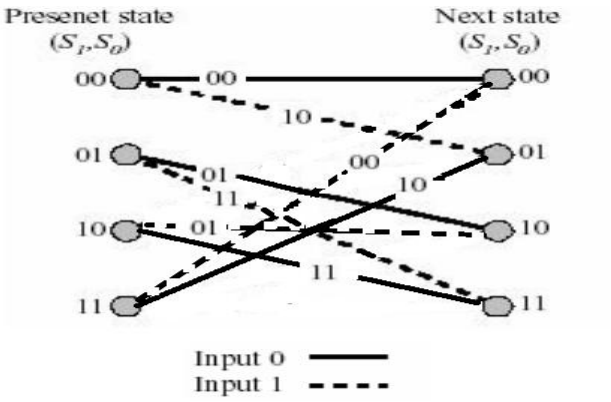

Fig 2.3- Trellis diagram

The encoding for the sequence 0 1 1 0 1 0 0 and the output sequence is 00 10 11 10 01 11 10 from the above figure.

C. Viterbi Decoder

The Viterbi decoding algorithm is a decoding process for Convolutional codes. This algorithm is based on maximum likelihood algorithm. When the received signal is sampled and quantized into two levels, either zero or one, the hard decision decoding will be used. In such decoding, the path through the trellis diagram is determined using hamming distance measure. Where the trellis represents the extension of the state diagram that explicitly demonstrates the state diagram with the time. The hamming distance is defined as a number of bits that are different between the observed symbol at the decoder and the sent symbol from the encoder [11].The length of the trellis is equal to the length of the input sequence, which consists of the information bits followed by the reset sequence. The reset sequence forces the trellis in to the initial state, so the trace back can be started at the initial state. Figure shows the simplified Viterbi decoder block diagram. The Viterbi decoding process consists of the following main tasks: branch metric computation, state metric update, survivor path recording and output decision generation. The branch metric computation compares the received code symbol with the expected code symbol using bit wise XOR and counts the number of different bits. Although convolution encoding is a simple procedure, decoding of a convolution code is much more complex task. Several classes of algorithms exist for this purpose:

- Threshold decoding is the simplest of them, but it can be successfully applied only to the specific classes of convolution codes. It is also far from optimal.
- Sequential decoding is a class of algorithms performing much better than threshold algorithms. Their serious advantage is that decoding complexity is virtually independent from the length of the particular code. Although sequential algorithms are also suboptimal, they are successfully used with very long codes, where no other algorithm can be acceptable. The main drawback of sequential decoding is unpredictable decoding latency.
- Viterbi decoding is an optimal (in a maximum-likelihood sense) algorithm for decoding of a convolution code. Its main drawback is that the decoding complexity grows exponentially with the code length. So, it can be utilized only for relatively short codes [12].

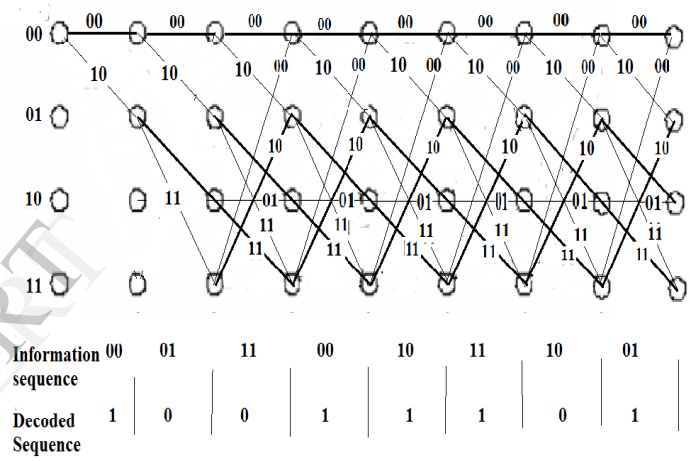

Fig 2.4 Viterbi Algorithm

D. SPARTAN XC3S500E FPGA

The Spartan®-3E family of Field-Programmable Gate Arrays (FPGAs) solves the design challenges in most high volume ,cost-sensitive, I/O-intensive electronic applications .The five-member family offers densities ranging from 50,000 to 1.4 million system gates, as shown in Table 1. The Spartan-3E FPGAs are part of the Extended Spartan-3E family, which also include the non-volatile Spartan-3EN and the higher density Spartan-3EDSP FPGAs. The Spartan-3E family builds on thse success of the earlier Spartan-3E andSpartan-3 FPGA families. New features improve system performance and reduce the cost of configuration. These Spartan-3E family enhancements, combined with proven 90nm process technology, deliver more functionality and bandwidth per dollar than ever before, setting the new standard in the programmable logic industry. Because of their exceptionally low cost, Spartan-3 FPGAs are ideally suited to a wide range of consumer electronics applications ,including broadband access, home networking, display/projection, and digital television equipment. TheSpartan-3A family is a superior alternative to mask programmed ASICs. FPGAs avoid the high initial cost, lengthy development cycles, and

the inherent inflexibility of conventional ASICs, and permit field design upgrades.

Table II Comparative Study between different implementations

Parameters	ASIC	DSP	FPGA
Flexibility	Better	Good	Best
Performance	Good	Good	Best
Area Utilization	Not	Good	Best
Optimization	Not possible	Not possible	Possible

E. Hardware Design And Development

The figure 2.5 show the hardware design of a viterbi decoder the convolutional encoder is realized by a hardware using XOR Gate, flip flop. The 8 bit input is given through a DIP switches the shift register is used to provide this input to convolutional encoder the output of encoder is given to a SPARTAN FPGA through a 2:1 multiplexer.

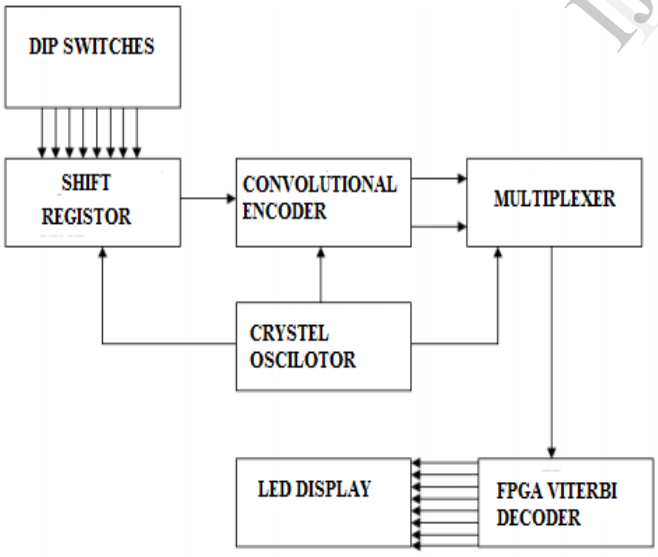

Fig 2.5 System Block Diagram

A description of the hardware's structure and behavior is written in a high-level hardware description language (usually VHDL or Verilog) and that code is then compiled and downloaded prior to execution. Of course, schematic capture is also an option for design entry, but it has become less popular as designs have become more complex and the language-based tools have improved. The overall process of hardware development for programmable logic is shown in Figure [11].

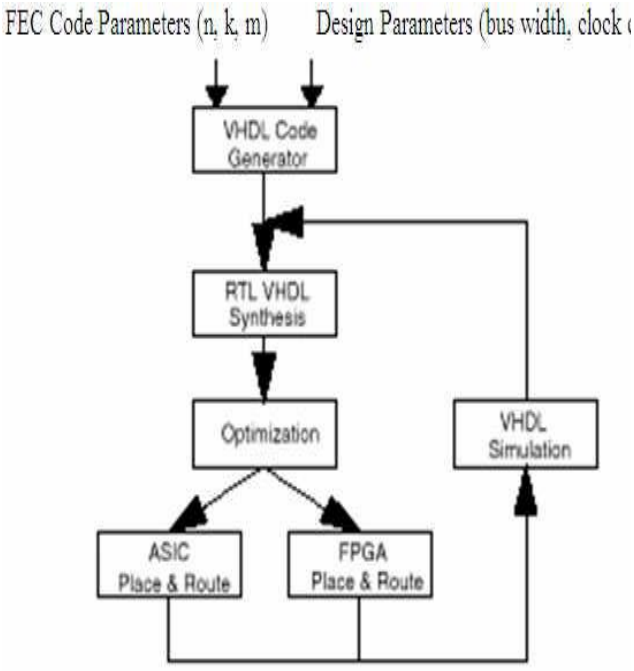

Fig 2.6 Design Flow

Xilinx ISE is a software tool produced by Xilinx for synthesis and analysis of HDL designs, which enables the developer to synthesize ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer. This design is simulated using Model Sim Starter Edition and synthesized using Xilinx 10.1 ISE. Model Sim SE entry-level simulator, offers VHDL, Verilog, or mixed-language simulation. Using Hardware Description Languages (HDLs) to design high-density FPGA devices has the following
advantages:

1. Top-Down Approach for Large Projects: Designers use HDLs to create complex designs. The top-down approach to system design works well for large HDL projects that require many designers working together. After the design team determines the overall design plan, individual designers can work independently on separate code sections.
2. Functional Simulation Early in the Design Flow: You can verify design functionality early in the design flow by simulating the HDL description. Testing your design decisions before the design is implemented at the Register Transfer Level (RTL) or gate level allows you to make any necessary changes early on.
4. Early testing of Various Design Implementations
5. Reuse of RTL Code [13]

Then, using an electronic design automation tool, a technology-mapped net list generates. The net list can then be fitted to the actual FPGA architecture using a process called place-and-route, usually performed by the FPGA Company's proprietary place-and-route software. The user will validate

the map, place and route results via timing analysis, simulation, and other verification methodologies. Once the design and validation process I s complete, the binary file generated (also using the FPGA company's proprietary software) is used to (re) configure the FPGA device. Figure 2.7 show the design flow of Xilinx [15].

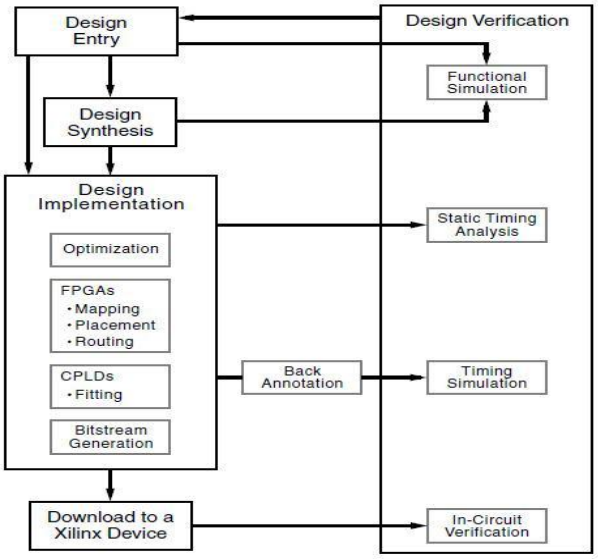

Fig 2.7 Xilinx Design Flow

E. Multiple Booting Technique

The Spartan-3E Starter Kit board [14] supports a variety of FPGA configuration options. One of these options is programming the on-board 128Mbit Intel Strata Flash(parallel NOR Flash PROM), then configuring the FPGA from the image stored in this Flash PROM using BPI Up or BPI Down configuration modes. Moreover, an FPGA can be dynamically loaded with two different FPGA configurations using the Spartan-3E FPGA's Multi Boot mode. Figure demonstrates the Multiple Booting Technique for two different applications. The multiple booting requires some steps that are different from those in the case of a single design implementation. In brevity, these steps require building more than one VHDL designs that include an additional selector input bits in the entity of each VHDL design program. The additional input bits are constrained for some of the slide switches in the Spartan 3E FPGA kit through the user constraint file (UCF). This gives the advantage for selecting the design configuration on the FPGA; for example: if n is the number of selection inputs, then 2n will be the number of the total different designs that can be configured on the FPGA through the Intel Strata Flash. In this research, two different designs were considered on the mentioned flash and then applied on the FPGA separately to be either Convolutional encoder or Viterbi decoder, configuration is different from the traditional method in the programming file type and the steps of programming. The PROM is retained with the design after power- off, then the FPGA will load the design from the PROM after power- on by pressing the programming pulse switch on the kit. Additionally, before final implementation; two bit files generated from the two designs must be merged in one MCS file which is downloaded on the parallel NOR flash that contains the two different designs configurations.

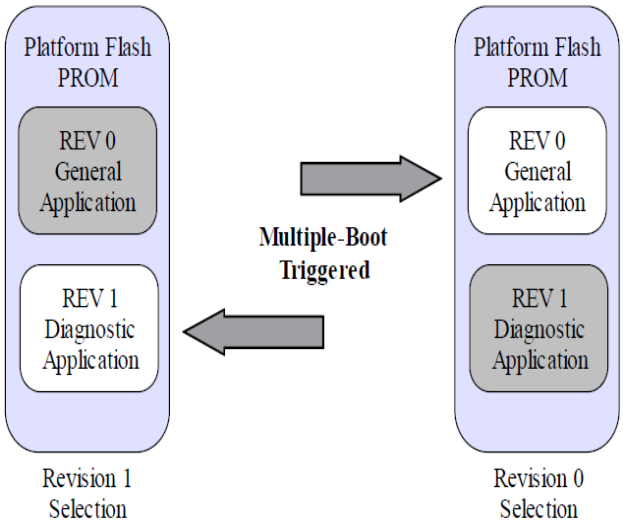

Figure 2.8 The Multiple Booting Technique

III. SIMULATION AND SYNTHESIS RESULTS

Synthesis is a process of constructing a gate level net list from a register transfer level model of a circuit described in Verilog HDL. Increasing design size and complexity, as well as improvements in design synthesis and simulation tools, have made Hardware Description Languages (HDLs) the preferred design languages of most integrated circuit designers. The two leading HDL synthesis and simulation languages are Verilog and VHDL. Both have been adopted as IEEE standards. The Xilinx ISE™ software is designed to be used with several HDL synthesis and simulation tools that provide a solution for programmable logic designs from beginning to end. [7] RTL View is a Register Transfer Level graphical representation of your design. This representation (.ngr file produced by Xilinx Synthesis Technology (XST)) is generated by the synthesis tool at earlier stages of a synthesis process when technology mapping is not yet completed. After the HDL synthesis phase of the synthesis process, the RTL Viewer can be used to view a schematic representation of the pre-optimized design in terms of generic symbols that are independent of the targeted Xilinx device, for example, in terms of adders, multipliers, counters, AND gates, and OR gates. After the optimization and technology targeting phase of the synthesis process, the Technology Viewer can be used to view a schematic representation of the design in terms of logic elements optimized to the target Xilinx device or "technology," for example, in terms of LUTs, carry logic, I/O buffers, and other technology specific components. Typical input and output are as indicated below. Encoded noise
Input bits K = 10010110
Encoded bits Y= 10011100000010111

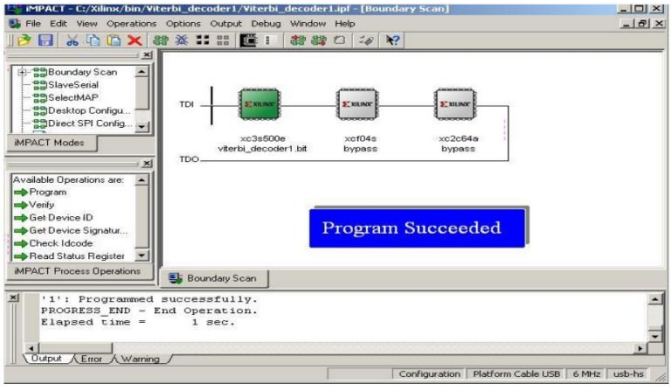

Fig 3.1 situation of programming window

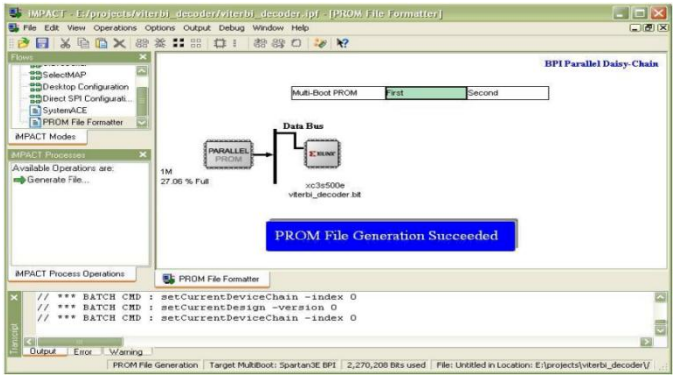

Fig 3.2 The Final Step in The Generation of MCS File

Each design is simulated separately and then implemented by applying the multi- boot technique. The generation of MCS file was necessary to combine the two designs. Then this file has been transferred to the Spartan-3E FPGA kit and stored in the PROM through the serial link using the Hyper Terminal windows program. The technology schematic of encoder is shown below

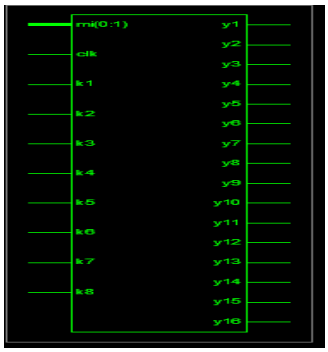

Fig 3.3 Technology Schematic of Convolutional Encoder

The utilization summary of Convolutional encoder and Viterbi decoder architecture given below. The Viterbi decoder occupied more area than Convolutional encoder due to the complexity increasing of this design. The maximum operating frequency (last row in this table) depends on the target technology of the design. In the used Spartan 3E, the maximum operating frequency is equal to the clock frequency provided by the kit, that is 50 MHz. Simulation Waveforms of Convolutional Encoder The Simulation Waveform of Viterbi Decoder is shown in Fig.3.4. To observe the speed and resource utilization, RTL is generated, verified and synthesized using Xilinx Synthesis Tool (XST).

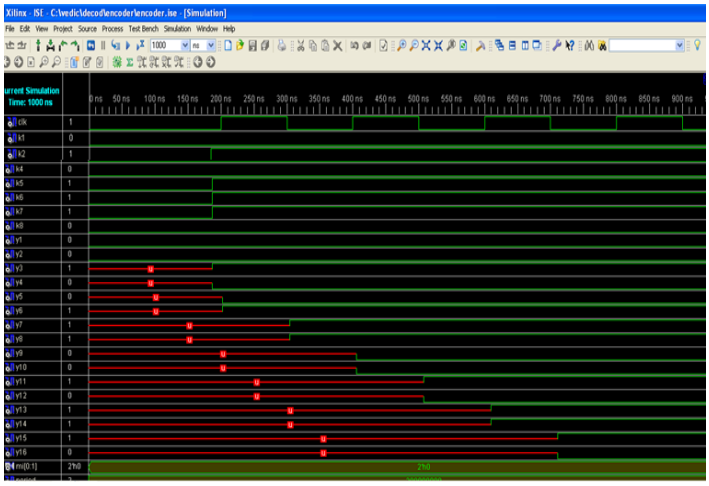

Fig 3.4 waveform Simulation of Convolutional Encoder

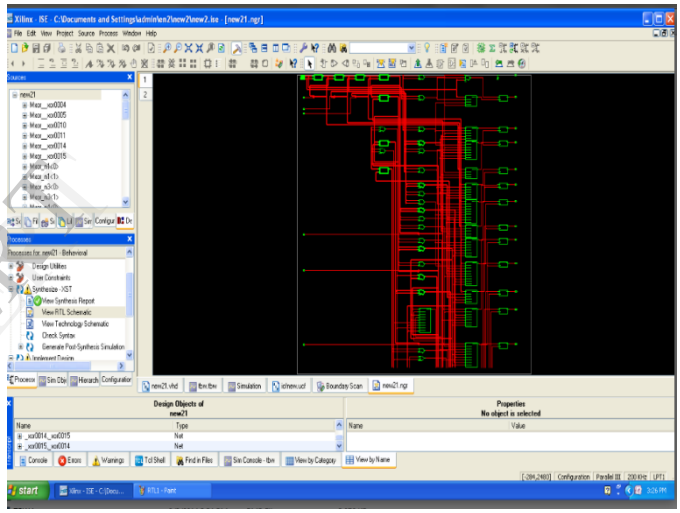

Fig 3.5 RTL schematic of convolutional encoder

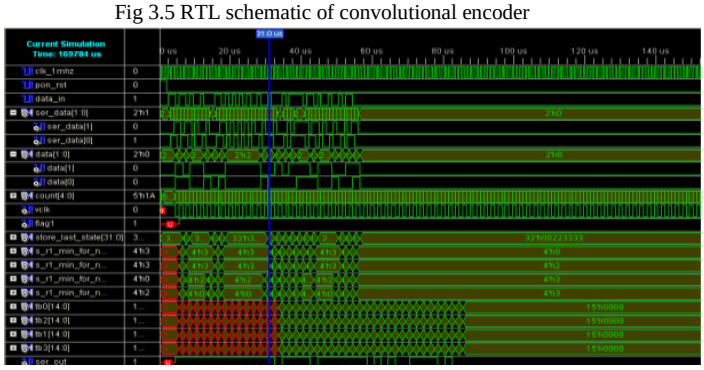

Fig 3.6 waveform Simulationof Viterbi Dicoder

a. Design Summary of Convolutional encoder

```
=========================================
=============================
*          Synthesis Options Summary          *
=========================================
=============================
```

---- Source Parameters

Input File Name	: "new21.prj"
Input Format	: mixed
Ignore Synthesis Constraint File	: NO

---- Target Parameters

Output File Name	: "new21"
Output Format	: NGC
Target Device	:xc3s500e-4-pq208

---- Source Options

Top Module Name	: new21
Automatic FSM Extraction	: YES
FSM Encoding Algorithm	: Auto
Safe Implementation	: No
FSM Style	: LUT
RAM Extraction	: Yes
RAM Style	: Auto
ROM Extraction	: Yes
Mux Style	: Auto
Decoder Extraction	: YES
Priority Encoder Extraction	:Yes
Shift Register Extraction	: YES
Logical Shifter Extraction	: YES
XOR Collapsing	: YES
ROM Style	: Auto
Mux Extraction	: Yes
Resource Sharing	: YES
Asynchronous To Synchronous	: NO
Multiplier Style	: Auto
Automatic Register Balancing	: No

---- Target Options

Add IO Buffers	: YES
Global Maximum Fanout	: 500
Add Generic Clock Buffer(BUFG)	: 24
Register Duplication	: YES
Slice Packing	: YES
Optimize Instantiated Primitives	: NO
Use Clock Enable	: Yes
Use Synchronous Set	: Yes
Use Synchronous Reset	: Yes
Pack IO Registers into IOBs	: Auto
Equivalent register Removal	: YES

---- General Options

Optimization Goal	: Speed
Optimization Effort	: 1
Keep Hierarchy	: No
Netlist Hierarchy	: As_Optimized
RTL Output	: Yes
Global Optimization	: AllClockNets
Read Cores	: YES
Write Timing Constraints	: NO
Cross Clock Analysis	: NO
Hierarchy Separator	: /
Bus Delimiter	: <>
Case Specifier	: Maintain
Slice Utilization Ratio	: 100
BRAM Utilization Ratio	: 100
Verilog 2001	: YES
Auto BRAM Packing	: NO
Slice Utilization Ratio Delta	: 5

b. Design Summery of Viterbi Decoder

```
=========================================
=============================
*          Synthesis Options Summary          *
=========================================
=============================
```

---- Source Parameters

Input File Name	: "decod1.prj"
Input Format	: mixed
Ignore Synthesis Constraint File	: NO

---- Target Parameters

Output File Name	: "decod1"
Output Format	: NGC
Target Device	:xc3s500e-4

---- Source Options

Top Module Name	: decod1
Automatic FSM Extraction	: YES
FSM Encoding Algorithm	: Auto
Safe Implementation	: No
FSM Style	: LUT
RAM Extraction	: Yes
RAM Style	: Auto
ROM Extraction	: Yes
Mux Style	: Auto
Decoder Extraction	: YES
Priority Encoder Extraction	: Yes
Shift Register Extraction	: YES
Logical Shifter Extraction	: YES
XOR Collapsing	: YES
ROM Style	: Auto
Mux Extraction	: Yes
Resource Sharing	: YES
Asynchronous To Synchronous	: NO
Multiplier Style	: Auto
Automatic Register Balancing	: No

---- Target Options

Add IO Buffers	: YES
Global Maximum Fanout	: 500
Add Generic Clock Buffer(BUFG)	: 24
Register Duplication	: YES
Slice Packing	: YES
Optimize Instantiated Primitives	: NO
Use Clock Enable	: Yes
Use Synchronous Set	: Yes
Use Synchronous Reset	: Yes
Pack IO Registers into IOBs	: Auto

Equivalent register Removal : YES

---- General Options

Optimization Goal	: Speed
Optimization Effort	: 1
Keep Hierarchy	: No
Netlist Hierarchy	: As_Optimized
RTL Output	: Yes
Global Optimization	: AllClockNets
Read Cores	: YES
Write Timing Constraints	: NO
Cross Clock Analysis	: NO
Hierarchy Separator	: /
Bus Delimiter	: <>
Case Specifier	: Maintain
Slice Utilization Ratio	: 100
BRAM Utilization Ratio	: 100
Verilog 2001	: YES
Auto BRAM Packing	: NO
Slice Utilization Ratio Delta	: 5

IV. Conclusions

Applying the multiple booting technique can be done on any FPGA kit that supports the use of such technique like Spartan 3E FPGA Starter kit (supported with XC3S500E). In this research, the architectures of the Convolutional encoder (2, 1, 3) and the Viterbi decoder were designed and implemented on XC3S500E FPGA chip built in Spartan 3E FPGA Starter kit using multiple booting technique. The reuse of the same hardware for different applications is the main benefit of multiple booting technique which gives an efficient re-configurability to the chip. This scheme gives the flexibility to change the function of the chip between the Convolutional encoder and Viterbi decoder which is very useful in modern systems. Multiple booting technique allows other types of encoders with different parameters and the corresponding decoders to be added. Consequently, the device can be operated in various modes as needed.

ACKNOWLEDGMENT

The portion of success is brewed by the efforts put in by many individuals. It is constant support provided by people who give you the initiative, who inspire you at each step of your endeavour that eventually helps you in your goal. I wish to express my deep gratitude and heartily appreciation for the invaluable guidance of our professors throughout the span of preparing this seminar. We are indebted to our college **Principal Dr. S. P. Bhosle.**

I am also thankful to our **HOD Mrs. V. M. Kulkarni** and my Project **Guide (Ms. S. S. Patki)** for his invaluable and elaborate suggestions. Their excellent guidance made me to complete this task successfully within a short duration. The inspiration behind the every aspect of life constructs a way to get success, which I have got from all the professors of the department. No thanks giving would be complete without mentioning our parents and friends, without their constant support and encouragement, this assignment would have not been successful.

REFERENCES

[1] A. J. Viterbi and J. K. Omura, "Principles of Digital Communications and Coding", McGraw-Hill, NY, 1979.
[2] A. J. Viterbi, "Convolutional codes and their Performance in communication systems", IEEE Transaction Communication Technology, Vol.19, pp.751-77, Oct. 1971.pp 260-269
[3] J. H. Yuen,"Modulation and Coding for Satellite and Space Communications", IEEE Procedings, Vol. 78, No.7, pp. 1250-1266, July1990.
[4] B. Sklar, Digital Communications: "Fundamentals and Applications",2nd edition. Prentice-Hall, Upper Saddle River, N J, 2001.
[5] G. C. Clark Jr. and J. B. Cain, "Error-Correction Coding for Digital Communications", Plenum Press, NY, 1981.
[6] S.V.Viraktamath, Dr.G.V.Attimarad, V.P.Gejji, Ravi. H "Error Control Mechanism Using CODEC","International Conference on Communication Software and Networks 2009" (ICCSN 2009), February 27 - 28, 2009. Macau, China. Page No 549; 978-0-7695- 3522-7/09
[7] S.V.Viraktamath, Dr.G.V.Attimarad, "Impact of constraint length on performance of convolutional CODEC in AWGN channel for Image applications" has been published in "International Journalof Engineering Science and Technology (IJEST)", Vol 2(9),2010,46974701.
[8] Abdul fattah Mohammad Obeid, Alberto Garc´ıa Ortiz, Ralf Ludewig and Manfred Glesner, "Prototyping of a High P erformance Generic Viterbi Decoder", 13th IEEE International Workshop on Rapid System Prototyping (RSP'02).
[9] Hema .S, Suresh babu .V, Ramesh .P "FPGA Implementation of Viterbi Decoder" Proceedings of the 6th WSEAS Int. Conf. on Electronics,
[10] Stefan Bitterlich and Heinrich Meyr (1993)."Efficient Scalable Architectures forViterbi Decoders" Aachen University of Technology,Templergraben, Germany.pp 89-100.
[11] Christian Schuler and GMD IOKH. "Code Generation Tools for hardwareimplementation of FEC Circuits".
[12] http://www.1-core.com
[13] http://www.xilinx.com/itp/xilinx10/books/docs/sim/sim.pdf
[14] Xilinx, Inc., San Jose Calif., "Spartan-3E Field Programmable Gate Arrays",2006. http://www.xilinx.com.
[15] "A Parallel Viterbi Decoder for Block Cyclic and Convolution Codes",Department of Electronicsand Computer Science, University of Southampton.April 2006

APPENDIX

Fig : Pin Configuration of Dynamically Reconfigurable Pipelined Adaptive Error Corrector on FPGA Spartan 3E