IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 2, Apr-May, 2014
ISSN: 2320 – 8791(Impact Factor: 1.479)
www.ijreat.org

Implementation Of Convolutional Encoder And High Performance Viterbi Decoder Using Eda Tools

A.Srujani[1], MD.SuhaliAfroz[2]

[1]PG Scholar, Department of Electronics and Communication Engineering, Teegala Krishnareddy Engineering college Hyderabad, A.P-500 097, India

[2]Assistant Professor, Department of Electronics and Communication Engineering, TeegalaKrishnareddy Engineering college, Hyderabad, A.P-500 097, India

Abstract

The comprehension of an capable logic propose of a crypto scheme. The type of crypto Scheme considered in the convolutional en policy and acclimatizeive Viterbi de policy (AVD) with a Restraint extent, P of 3 and a policy rate (P/n) of 1/2 using pasture programmable gate mixture (PPGM)technology. Here, the features of Convolutional en policy and depolicy architecture are introduced and he way it can be implementable as an ASIC. Here the Viterbi Depolicy is designed for faster decoding speed and less routing area with a special path management unit. The Scheme is realized using Verilog HDL. It is simulated and synthesized using Modelsim Altera Starter Edition 6.6d and Xilinx 6.9 for RTL Design.

Keywords--*Convolutional Enpolicy, Viterbi Depolicyr, Verilog HDL,FPGA, PNPH Unit*

1.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. The encoder adds redundant bits to the sender's bit stream to create a codeword. 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 [16]. 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 [13] include digital modems and digital cellular telephone, where low latency, component cost and power consumption are must.

2.VITERBI DECODER

Fig. 1 shows Basic Block Diagram of Convolution Encoding and decoding which basically consists three main blocks: Convolutional Encoder, AWGN Channel and Viterbi Decoder [12]

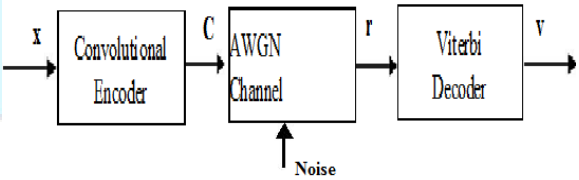

Fig. 1 Convolution Encoding and decoding

2.1 Convolutional Encoder

In convolutional encoding n-tuple of data is generated for every k-tuple of inputs based on both current and K-1 previous k-tuples where K is called constraint length of the code. A (n, k, m) convolutional code can be implemented with a k-input, n-output linear sequential circuit with input memory 'm'. Typically, 'n' and 'k' are

small integers with k<n, but the memory order 'm' must be made large to achieve low error probabilities. The constraint length K of the code represents the number of bits in the encoder memory that effect the generation of the n output bits and is defined as K = m+1. The code rate r of the code is a measure of the code efficiency and is defines as r= k/n. A Convolutional Encoder is a Finite state machine [13] i.e. a model of behavior composed of states, action and transition. Contents of first K-1 shift register stages defines the encoder state. Memory register start with 0 and modulo-2 adders among the registers and input generate the encoded data. Generator polynomial defines how the adders (XOR gates) are placed.The proposed Encoder has the following specifications below and schematic in Fig. 2.

Constraint Length: K = 3, Input bit: k = 1, Output bit: n = 2 Generator Polynomials:

$G1 = 1 + X + X^2$, $G2 = 1 + X^2$

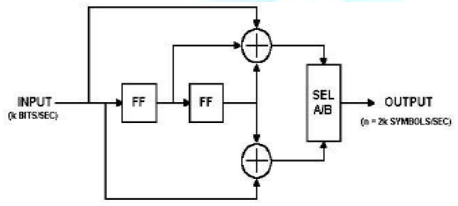

Figure 2: Convlolutional Encoder (Rate ½, K = 3)

2.2 Viterbi Algorithm

Viterbi decoding was developed by Andrew J. Viterbi, is an Italian-American electrical engineer and businessman who co-founded Qualcomm Inc. His seminar paper titled "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", published in IEEE Transactions on Information Theory, in April, 1967 [16]. Viterbi algorithm is a maximum likelihood method to find the most probable sequence of hidden states based on a given sequence of observed outputs in Hidden Markov model. However it reduces the computational load by taking the advantage of special structure in code trellis. The algorithm involves calculating a measure of distance between the received signal at the time t_1 and the entire trellis path entering each state at time t_1. The most likely path through the trellis will maximize this metric. The early rejection of the unlikely paths reduces the decoding complexity. Advantage of Viterbi algorithm is that it has self-correction of the code, minimization of transmitting Energy, minimization of BW and very good ability to correct wrong transmitted bits.

2.3 Viterbi Decoder

Error correction is an integral part of any communication system and for this purpose, the convolution codes are widely used as forward error correction codes. . The two decoding algorithms used for decoding the Convolutional codes are Viterbi algorithm and Sequential algorithm. Sequential decoding has advantage that it can perform very well with long constraint length Convolutional codes, but it has a variable decoding time. Viterbi decoding is the best technique for decoding the Convolutional codes but it is limited to smaller constraint lengths (K<10) [5]. It has fixed decoding time compared to sequential decoding. With the Viterbi algorithm, storage and computational complexity are proportional to 2^K. To achieve very low error probabilities, longer constraint lengths are required, and sequential decoding becomes attractive. The performance of a decoder is characterized by the number of decoded output bits which are in error, the Bit Error Rate or BER. The Viterbi algorithm [13], the most popular decoding approach for convolutional codes, determines a minimum distance path with regards to Hamming distances applied to each received symbol. A limiting factor in Viterbi decoder implementations is the need to preserve candidate paths at all $2K-1$ trellis states for each received symbol. This requirement leads to an exponential growth in the amount of computation performed and in the amount of path storage retained as constraint length K grows.

A Viterbi algorithm consists of the three major parts [13]: Branch metric unit, Path metric unit and trace back as shown in figure 3.

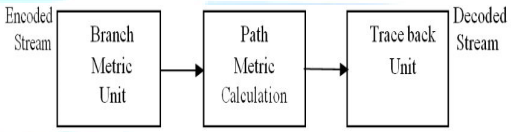

Figure 2: Viterbi algorithm

2.3.1 Branch metric calculation

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.

2.3.2 Path metric calculation

An accumulated Error metric called path metric (PM) contains the 2^{K-1} optimal paths. The current Branch Metric is added to previous PM and each the two distances are compared for all Add- compare select unit

In terms of speed the performance of Viterbi Decoder is mainly determined by the number of ACS (2^{K-1}) units and their computation time. As shown in figure each ACS unit comprises two adder blocks, a comparator and a selector block.

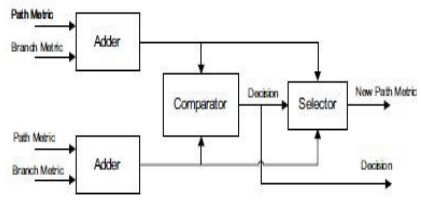

Figure 4: Block Diagram of Add Compare Select unit

2.3.3 Trace back unit

The final unit is trace back unit where the survivor path and output data are identified. The Viterbi decoding flowchart is given in Fig. 5.

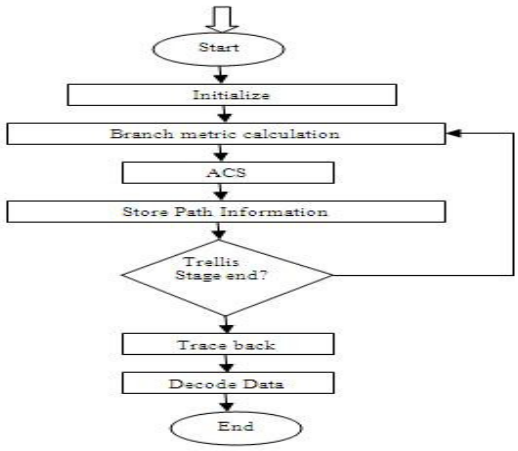

Figure 5: Viterbi decoding Flow Chart

2.4 Types of Viterbi Decoding

2.4.1 Hard decision Viterbi deocding

Demodulator output configured by variety of ways [4]: In which output of demodulator is quantized into two levels, zeros and one and fed into decoder (1- bit is used to describe each code symbol). Decoder operates on hard-decisions made by demodulator, decoding is called Hard-decision decoding. In which path through trellis is determined using hamming distance measure.

2.4.2 Soft decision Viterbi deocding

In which output of demodulator is quantized into greater than two levels [4]. If output of demodulator is quantized into 3-bit result in 8- level output then 3-bits is used to describe each code symbol. In which Euclidian distance as a distance is measured instead of hamming distance. The advantage of using soft- decision decoding is to provide decoder with more information, which decoder then use for recovering the message sequences. It provides better error performance than hard- decision type Viterbi decoding also Performance improvement of approximately 2 dB in required S/N ratio compared to two level quantization for a Gaussian Channel. Disadvantage of using soft decision decoding is increase in required memory size at decoder and reduce speed.

2.5 Viterbi Decoding Techniques

There are mainly two types of decoding techniques available in order to decode the data at the receiver end.

2.5.1 Register Exchange Method

In this method, a register assigned to each state contains information bits for the survivor path from the initial state to current state. In fact, register keeps decoded output sequences along the path. This method requires copy of all registers at each stage. The need to trace back is eliminated since the register of final state contains decoded output sequence. This approach results in complex hardware due to need to copy contents of all register in a stage to next stage. Since the RE method does not need tracing back, it is faster.

2.5.2 *Traceback Method*

Trace back is memory organization method to store survivor paths and retrieve the decoded data. Direct implementation of this method is not practical because of an infinite storage size is needed; therefore in practice semiconductor infinite memory locations are reused periodically. The Trace Back Unit performs three operations: write new Data, Trace Back Read and Decode Read. Memory is organized as a two dimensional structure where row are assigned to states and columns to time steps. Three memory blocks are used in operation: write block is used to store ACS decision vectors, Decode block where the decoded bit sequences is read in backward order and Trace Back Block which is used to find the starting point of next trance back sequences. Traceback Depth (D) is a predefined parameter that defines the size of each memory block [13]. To guarantee the convergence a traceback depth of D = 5K is sufficient and the memory block size will be 2^{K-1} x 5K. Traceback method is area efficient and better than RE method. . Register exchange method requires complex hardware compare to the Traceback method for larger constraint length though it will give faster speed.

In this project I had implemented a hard decision and trace back method for viterbi decoding.

3. PROGRAMMABLE DEVICES

Programmable devices are those devices which can be programmed by the user. Various programmable devices are PLDs, CPLDs, ASICs and FPGAs.

3.1 Field Programmable Gate Arrays

Field-Programmable Gate Arrays (FPGAs) are pre-fabricated silicon devices that can be electrically programmed to become almost any kind of digital circuit or system. FPGAs contain programmable logic components called "logic blocks", and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together"—somewhat like a one-chip programmable breadboard. Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.

They have many advantages over Application Specific Integrated Circuits (ASIC). ASICs are designed for specific application using CAD tools and fabricated at a foundry. Developing an ASIC takes very much time and is expensive. Furthermore, it is not possible to correct errors after fabrication. In contrast to ASICs, FPGAs are configured after fabrication and they also can be reconfigured. This is done with a hardware description language (HDL) which is compiled to a bit stream and downloaded to the FPGA.

The advantages of the FPGA approach to CPLD implementation include highest amount of logic density, the most features, and the highest performance. CPLDs, by contrast, offer much smaller amounts of logic - up to about 10,000 gates. But CPLDs offer very predictable timing characteristics and are therefore ideal for critical control applications.

The advantages of the FPGA approach to DSP implementation include higher sampling rates than are available from traditional DSP chips, lower costs than an ASIC. The FPGA also adds design flexibility and adaptability with optimal device utilization conserving both board space and system power that is often not the case with DSP chips. Due to the increase of transistor density FPGA were getting more powerful over the years. Therefore, FPGAs are increasingly applied to high performance embedded systems.

3.2 SPARTAN XC3S400A FPGA

The Spartan®-3A 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 offersdensities ranging from 50,000 to 1.4 million system gates. The Spartan-3A FPGAs are part of the Extended Spartan-3A family, which also include the non-volatile Spartan-3AN and the higher density Spartan-3A DSP FPGAs. The Spartan-3A family builds on the success of the earlier Spartan-3E and Spartan-3 FPGA families. New features improve system performance and reduce the cost of configuration. These Spartan-3A family enhancements, combined with proven 90 nm 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-3A FPGAs are ideally suited to a wide range of consumer electronics applications, including broadband access, home networking, display/projection, and digital television equipment. The Spartan-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.

4.SOFTWARE USED

Xilinx ISE (**I**ntegrated **S**oftware **E**nvironment) 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 and synthesized using Xilinx 10.1 ISE.

4.1 Designing FPGA Devices using VHDL

VHDL stands for VHSIC Hardware Description Language. VHSIC is itself an abbreviation for Very High Speed Integrated Circuits. VHDL is hardware description language. It describes behaviour of an electronic system, from which the physical Layer or system can then be implemented. It is intended for circuit synthesis as well as circuit simulation.

The two main applications immediate of VHDL are in the field of Programmable logic devices and in the field of ASICs. Once the VHDL code has been written, it can be used either to implement the circuit in programmable device or can be submitted to a foundry for fabrication of a ASICs chip.

5.SIMULATION AND SYNTHESIS RESULTS

Synthesis is a process of constructing a gate level netlist 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.

5.1 Simulation Waveforms of Viterbi Decoder

The Simulation Waveform of Viterbi Decoder is shown in Fig. 6. To observe the speed and resource utilization, RTL is generated, verified and synthesized using Xilinx Synthesis Tool (XST).

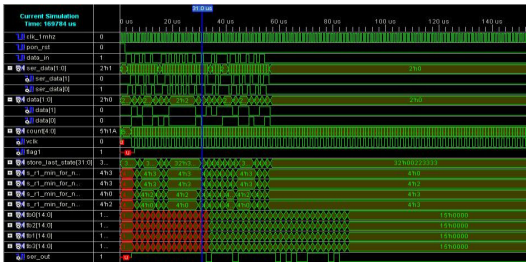

Figure 6: Simulation Waveform of Viterbi Decoder

5.2 RTL Schematic of Viterbi Decoder

Below Shown is the RTL Schematic of the Viterbi Decoder.

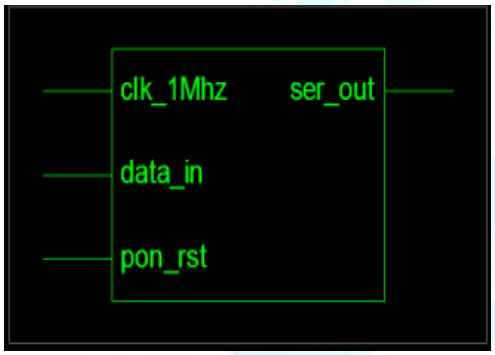

Figure 7: RTL Schematic of Viterbi Decoder

5.3 Device Utilization Report

This synthesis report is generated after the compilation of Design for the targeted Xilinx SPARTAN 3A based Xc3s400a FPGA Device. Here, The Design unit is not implemented on targeted FPGA Device. This report contains about component used.

Table 1. Device utilization Summary

Device Utilization Summary		
Logic Utilization	Used/Available	Utilization
Number of Slices	104/3584	2 %
Number of Slice FFs	95/7168	1 %
Number of 4 input LUTs	146/7168	2 %
Number of Bonded IOBs	4/495	2 %
Number of GCLKs	2/24	8 %

5.4 Timing and Power Summary

After the synthesis report, the timing diagram generated according to the given input. With the help of timing diagram speed grade, Minimum period, Maximum Frequency, Maximum output required time after clock is calculated.

Timing Summary

- Speed Grade: -4
- Minimum period: 30.190ns
- Maximum Frequency: 33.124MHz
- Minimum input arrival time before clock: 2.993ns
- Maximum output required time after clock: 5.531ns
- **Power** summary

  Total estimated power consumption: P (mw): 49 mw

5.5 Comparative Analysis between Various FPGA Devices

Different FPGA family of SPARTAN are used to measure the performance of proposed Viterbi Decoder Design.

5.5.1 Performance Comparison of proposed Viterbi Decoder Design in Various SPARTAN FPGA Devices

Table 2. Comparison between various SPARTAN FPGA Devices

Family	Device	No. of Slices	No. of Slice FFs.	Total No. Of 4 i/p LUTs	Number of Bonded IOBs	Max. Freq.
SPARTAN2	xc3s100 -6fg256	103/1200 (8 %)	95/2400 (3 %)	150/2400 (6 %)	4/176 (2 %)	22.867 MHz
SPARTAN 2E	xc2s200e -7ft256	104/2352 (4 %)	95/4704 (2 %)	150/4704 (3 %)	4/178 (2 %)	26.867 MHz
SPARTAN 3E	xc3s500e -4fg320	104/4656 (2 %)	95/9312 (1 %)	143/9312 (1 %)	4/232 (1 %)	31.512 MHz
SPARTAN 3A	xc3s400a -4ft256	104/3584 (2 %)	95/7168 (1 %)	146/7168 (2 %)	2/195 (2 %)	33.124 MHz

5.5.2 Performance Comparison of proposed Viterbi Decoder Design in Various VIRTEX FPGA Devices

Different FPGA family of VIRTEX are used to measure the performance of proposed Viterbi Decoder Design.

Table 3. Comparison between various VIRTEX FPGA Devices

Family	Device	No. of Slices	No. of Slice FFs.	Total No. Of 4 i/p LUTs	Number of Bonded IOBs	Max. Freq.
Virtex 2	xc2v500 -6fg256	104/3072 (3 %)	95/6144 (1 %)	142/6144 (2 %)	4/172 (2 %)	40.888 MHz
Virtex 4	xc4vlx100 -12ff1148	105/49152 (0 %)	95/98304 (0 %)	145/98304 (3 %)	4/768 (0 %)	67.057 MHz
Virtex 5	xc5vlx110 -3ff676	95/69120 (1 %)	No. of Fully used LUT-FF Pair 23/190 (12 %)	No. of Slice LUT 118/69120 (1 %)	4/440 (0 %)	113.104 MHz
Virtex E	Xcv400e -8fg676	103/4800 (2 %)	95/9600 (0 %)	150/9600 (1 %)	4/404 (0 %)	29.933 MHz

6. CONCLUSION

In this Paper Resource optimized Viterbi Decoder has been proposed. The proposed Viterbi Decoder has been designed with VHDL using traceback method. The designed Viterbi Decoder has been simulated using Xilinx ISE simulator and synthesized with XST. The simulated and synthesized results show that proposed design can work at an estimated frequency of 33.124 MHz by using considerable less resources of target FPGA device SPARTAN 3A. This Paper also contains comparative analysis between various FPGA devices for the same Design. The result shows that proposed design can work at Max. Frequency 113.104 MHz for targeted FPGA Device VIRTEX 5 among all FPGA Devices.

So, VRTEX 5 FPGA Device can give Max. Frequency for proposed Design among all FPGA Devices.

REFERENCES

[1] Mahe Jabeen and Salma Khan, Design of Convolution Encoder and Reconfigurable Viterbi Decoder, *International Journal of Engineering and Science, Vol. 1, No.3*, Sep 2012.

[2] P. Subhashini, D. R. Mahesh Varma and Y. David Solomon Raju, Implementation Analysis of adaptive Viterbi Decoder for High Speed Applications, International *Journal of Computer Applications (0975-– 8887), Volume 31– No.2*, October 2011

[3] S.V.Viraktamath and G.V.Attimarad, Impact of constraint length on performance of Convolutional Codec in AWGN channel for image application, *International Journal of Engineering Science and Technology, Vol. 2(9)* , 2010, 4696-4700.

[4] Bernard Sklar, *Digital Communications Fundamentals and Application*, Published by Pearson education, Year 2003

[5] B.P.Lathi, *Modern Digital and Analog Communication Systems*, Third Edition.

[6] J.G. Proakis, *Digital Communications*, McGraw Hill.

[7] J. Bhaskar, *A VHDL Primer*, Third Edition

[8] Volnei A. Pedroni, *Circuit Design with VHDL*

[9] Christian Baumann, ―Field Programmable Gate Arrray (FPGA), Summary paper for the seminar Embedded System Architecture, University of Innsbruck, January 13, 2010.

[10] Matthias Kamuf, Member, IEEE, Viktor Öwall, Member, IEEE, and John B. Anderson, Fellow, IEEE, Optimization and Implementation of a Viterbi Decoder under Flexibility Constraints, *IEEE Transactions on Circuits and Systems—I: Regular Papers,Vol. 55, No. 8*, September 2008.

[11] C. Arun, V. Rajamani, Design and VLSI implementation of a Low Probability of Error Viterbi decoder, *First International Conference on Emerging Trends in Engineering and Technology,* IEEE 2008.

[12] Miloš Pilipovic, Marija Tadic, FPGAImplementation of Soft Input Viterbi Decoder for CDMA2000 System, *16Th Telecommunication forum TELEFOR*, 2008.

[13] Behzad Mohmad Heravi and Bahram Honary, Multi-rate Parameterized Viterbi Decoding for Partial Reconfiguration, *PGNet* 2006.

[14] Sriram Swaminathan, Russel Tessier, Dennis Goeckel and Wayne Burleson, A Dynamically Reconfigurable Adaptive Viterbi Decoder, FPGA 02, Feb 2002, pages 24-26.

[15] T. K. Truong, Senior Member IEEE, Ming- Tang Shih, Irving S. Reed, Fellow IEEE, and E.H. Sartorius, Member IEEE, A VLSI Design for Trace- back Viterbi Decoder, *IEEE Transaction on Communications, Vol. 40, No. 3*, March 1992.

[16] A. J. Viterbi, Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm, *IEEE Trans. Inform Theory, vol. IT-13*, pp. 260-269, Apr. 1967.