

High Data Rate Pipelined Adaptive Viterbi Decoder Implementation

Ahmed Samy Mohamed¹, Hatem M. Zakaria²
¹(Electrical Engineering Department, Egyptian Armed Forces , Egypt)
²(Benha Faculty of Engineering, Benha University, Egypt)

Abstract: This paper presents a pipelined Adaptive Viterbi algorithm of rate 1/2 convolutional coding with a constraint length $K = 3$ which is designed in a reconfigurable hardware to take full advantage of algorithm parallelism, specialization and the throughput rate. In present work, the hardware implementation of the pipelined Adaptive Viterbi algorithm is performed using FPGA processor (Spartan-3AN starter Field Programmable Gate Array (FPGA) kit), and Model- Sim simulation results are performed to ensure that the implemented scheme satisfy the design specification. On the other hand, processing time, power consumption, and design capacity should be studied well for real time implementation.

Keywords: Convolutional coding, Viterbi Decoder, Pipelining, Adaptive Viterbi algorithm.

I. Introduction

The forward error correction (FEC) system controls the error for data transmission in telecommunication and information theory. As the sender adds the error –correcting code (ECC) to the messages which is a redundancy to allow the receiver to detect and correct a limited number of errors occurring in the message. The receiver gains the ability by FEC to correct errors without utilizing a reverse channel to request data retransmission, [1]. On the other hand this advantage is at the cost of a fixed higher forward channel bandwidth. The process of adding this redundant information is known as channel coding, [1].

Generally, there are two approaches of channel coding; the convolutional coding requires memory and is better than the block coding which doesn't entail memory. This advantage form the ability of a better error correction compared with the block coding, In spite of it requires an extensive memory while decoding ,[2].

Convolution coding is used to act as powerful method, and the most widely implemented type for the FEC coding in wireless communication system, [3]. There are many algorithms for decoding a convolutionally encoded data. The Viterbi algorithm is the preferred decoding method due to its modest complexity and good performance for convolution codes, [4], as it is the most resource-consuming algorithm, and widely applied to decode and estimate the information in communications and signal processing units.

This paper is organized as follows; after the introduction, section 2 gives a survey on the Viterbi algorithm. Section 3 focuses on the Pipelined Adaptive Viterbi decoder. Hardware Implementation of the Pipelined Adaptive Viterbi algorithm using FPGA is presented in section 4. Finally, conclusion comes in section 5.

II. Viterbi Algorithm

The Viterbi Algorithm is a dynamic algorithm that uses certain path metrics to compute the most likely path of a transmitted sequence, [6]. It is based on the maximum likelihood criteria, as it computes the received sequence with every possible code sequence as shown in figure (1). The criterion for deciding between two paths is to select the one having the smallest metric. This method maximizes the probability of a correct decision, [3].

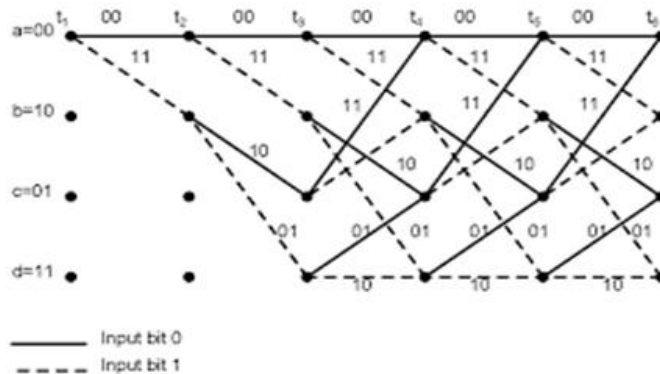


Figure (1): Flow of Viterbi Algorithm.

The Viterbi decoder consists of four main processes; branch metric computation, state metric update, survivor path recording and output decision generation as shown in figure (2), [4].

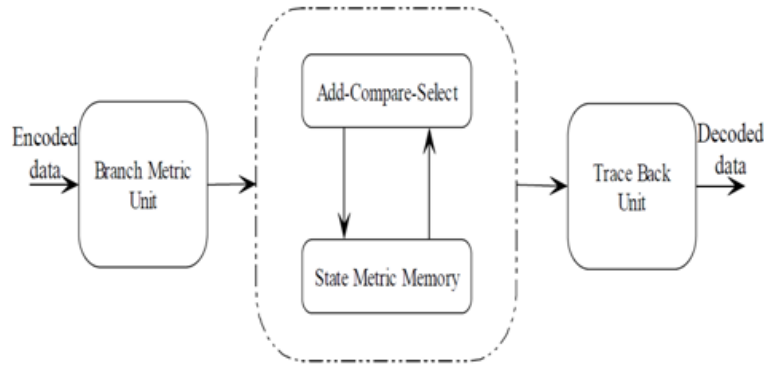


Figure (2): Simplified Viterbi decoder block diagram.

In decoding, the path difference between two states in the trellis diagram is called hamming distance in case of the hard decision decoder, and Euclidean distance in case of the soft decision decoder, [4]. This 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 as shown in figure (3), [5].

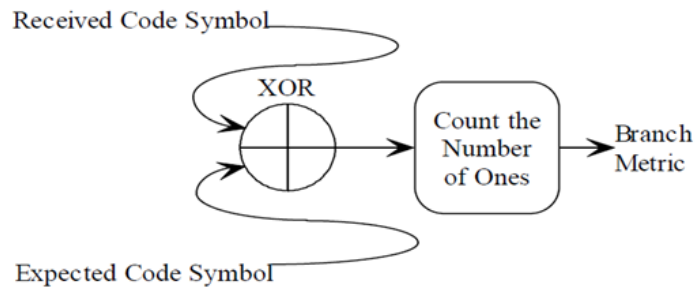


Figure (3): Branch Metric Block diagram.

The decoder will compute Branch metrics. Then the best path metric will be selected on the minimal value of the two possible paths basis. This process will continue till the trellis is completely filled. As depicted in Figure (3). The state metric update is achieved by computing the partial path metrics through accumulating the new values of branch metric to the previous values of state metric. There are two values of state metric due to the two branches enter each node in the trellis. So, the updated value of state metric at each node will take the minimum value. This can be done by using Add Compare Select (ACS) Unit as shown in figure (4), [4].

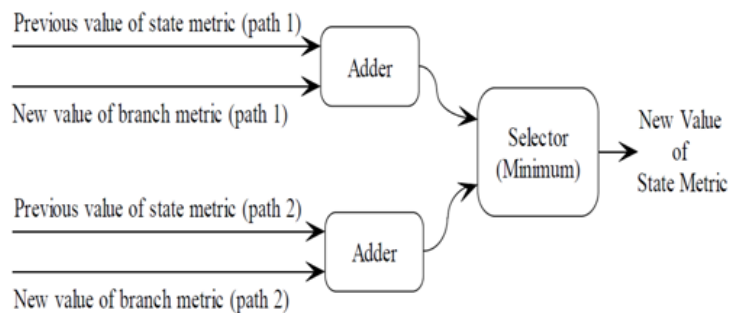


Figure (4): Add-Compare-select Unit.

After updating the state metric for each node in the trellis, the survivor path can be identified by passing through the minimum value of state metric, [4]. Then, the partial path metrics will be updated and survivor information will be stored, [4]. At last, by tracing back in time, all survivor paths merge into a unique path, which is the most likely signal path that we are considering, [6]. The Viterbi algorithm operates well for less-complex codes, indicated by constraint length, K. However the more powerful codes with large constraint lengths, K, the complexity of the algorithm grows exponentially with the code length. To overcome this

problem, the Adaptive Viterbi Decoder is used to provide an acceptable performance with practicable decoding complexity with suitable design, and provides a large coding gain with good performance, [3].

Adaptive Viterbi Decoder is used to reduce the average computation and path storage required for Viterbi algorithm, [7]. So The Adaptive Viterbi Decoder only keeps a number of the most likely state instead of the whole of $2K-1$ states, where K is the constraint length of the convolution encoder which is time conservation. As an alternative of examining all possible paths in trellis graph and determine the most likely one in the conventional decoder. The rest of the other states are discarded based on the most likelihood or metric value of the path. The two decoder types are the hard decision which utilizing the Hamming distance to calculate the metric value and the soft decision decoder is Euclidean distance, [3]. Figure (5) shows, the general block diagram of the Adaptive Viterbi Decoder.

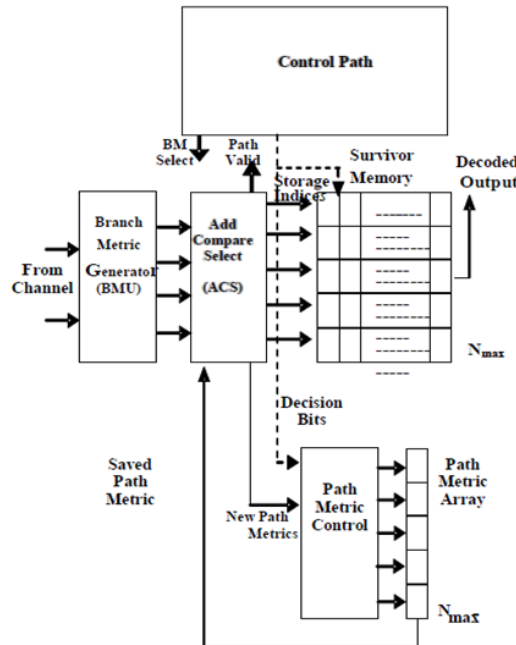


Figure (5): Adaptive Viterbi Decoder block diagram, [7].

III. Pipelined Adaptive Viterbi Algorithm

A pipelined architecture is a technique that implements parallelism formation called instruction-level parallelism within a single processor, which is used for overlapping the execution of several instructions to reduce the execution time of a set of instructions. To achieve the Pipelined, two words must be considered during perform pipelined process; Latency and throughput. Latency is the amount of time that a single operation takes to execute. Throughput is the rate at which operations get executed (generally expressed as operations/second or operations/cycle).

The pipelined Adaptive Viterbi algorithm decreases the average computations number of decoded information bits while achieving comparable bit-error rates (BER) versus Viterbi algorithm implementations, [8]. It is widely used to improve the performance of the algorithm by splitting the processing or the operation instruction into a series of independent steps, with storage at the end of each step, [9], as shown in figure (6), [6].

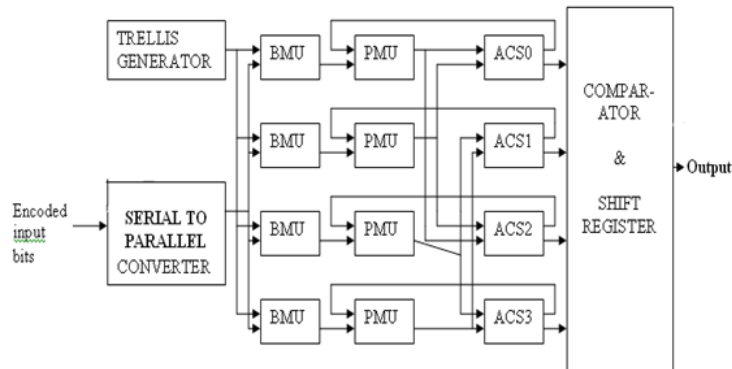


Figure (6) Pipelined Adaptive Viterbi Decoder block diagram, [6].

By breaking the logic into smaller pieces and inserting flip flops between the pieces of logic, the time delay of the logic to give valid outputs is reduced. By increasing the pipeline stages, the path delays of each stage are decreased and consequently the overall performance of the circuit is improved and increases their instruction throughput. This speedup process is possible as a result of increasing the number of pipeline stages, the more operations the processor can work on simultaneously and the more operations it can complete in a given period of time. And some combinational circuits such as adders or multipliers can be made faster by adding more circuitry. But, the operation latency in a non-pipelined processor is slightly lower than in a pipelined equivalent. This is because extra flip-flops must be added to the data path of a pipelined processor, [9].

IV. Pipelined Adaptive Viterbi Algorithm Hardware Implementation

In this section, an overview of FPGA structures and design steps shall be presented. Design and implementation of the Pipelined Adaptive Viterbi Decoder is introduced and illustrated by block diagrams explaining each sub-module supported with experimental results. The presented scheme is implemented on a Xilinx Spartan-3AN (XC3S700AN in FGG484 package) FPGA. Figure (7) shows the schematic diagram of the path metric unit calculation, which is generated by the Xilinx package ISE13.1 program.

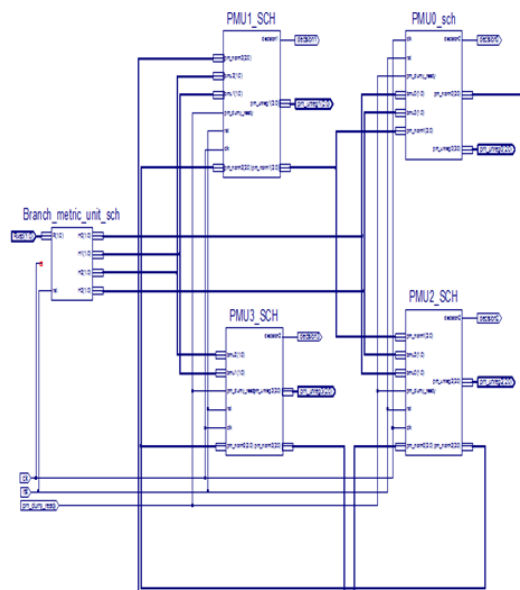


Figure (7): Schematic diagram of the path metric unit calculation.

Figure (8) shows the schematics diagram of the Pipelined Adaptive Viterbi Decoder minimum path block diagram, which is generated by the Xilinx package ISE13.1 program.

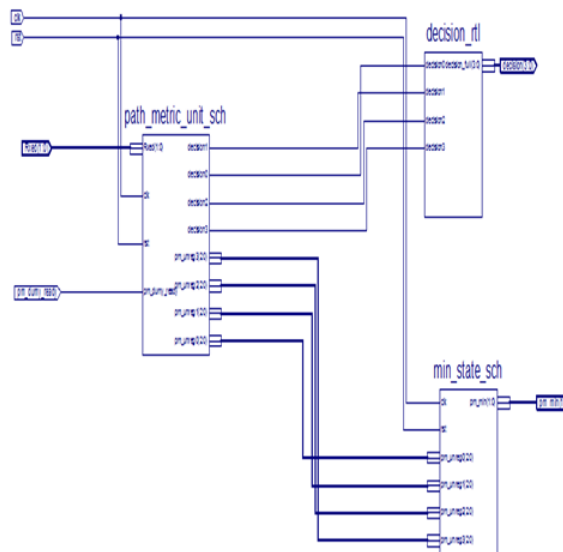


Figure (8): Schematic diagram of the minimum path.

Figure (9) shows the schematic diagram of the Pipelined Adaptive Viterbi Decoder which involves an add-compare-select unit with the normalizer unit. Every unit is connected with the other according to the pre-designed Pipelined Adaptive Viterbi Decoder, which is generated by the Xilinx package ISE13.1 program.

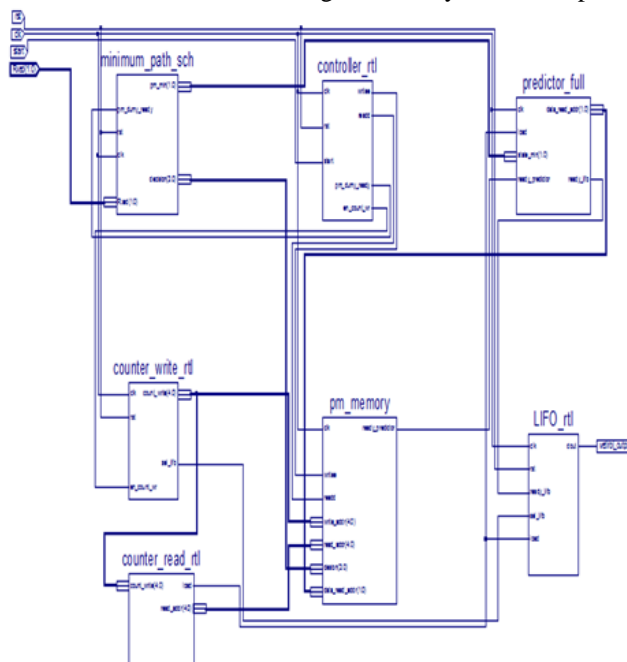


Figure (9): schematic diagram of the Pipelined Adaptive Viterbi Decoder.

Figure (10) shows the schematics diagram of the branch metric unit calculation, which is generated by the Xilinx package ISE13.1 program.

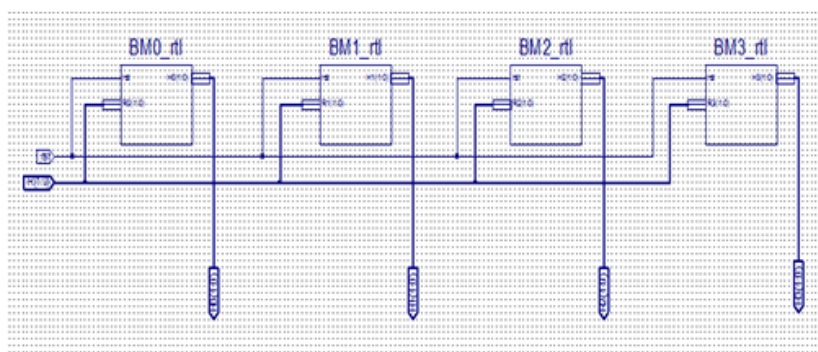


Figure (10): Schematic diagram of the branch metric unit calculation.

Model-Sim is a tool that integrates with Xilinx ISE to perform simulation and testing. Simulation is used to make sure that the logic of a design is correct and make sure that the design will behave as expected when it is downloaded onto the FPGA chip. Figures (11) to (14) shows the Model-sim simulation results for the Pipelined Adaptive Viterbi Decoder, which is generated by the Xilinx package ISE13.1 program and simulated using Model-sim 6.5.

The output of the encoder is "11111011011100011010001001001111100001011001111101".

The input to the decoder according to the multiplexer are:

- ❖ No Transmission (selector-00)
- ❖ 11111011011100011010001001001111100001011001111101 (selector -01) --no error
- ❖ 11111000011100011010001001001111100001011001111101 (selector -10) --error of 2 bits
- ❖ 11111000101100011010001001001111100001011001111101 (selector -11) --error of 4 bits

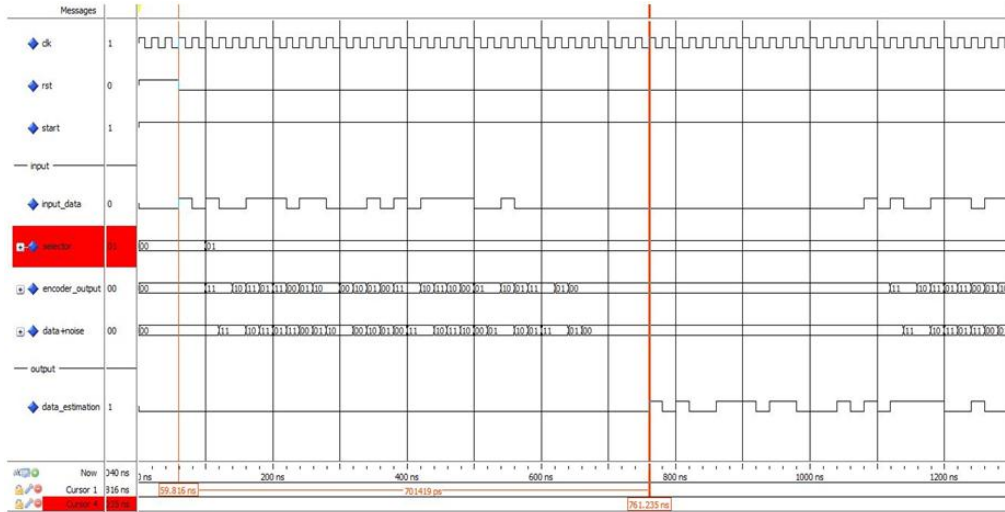


Figure (11): simulation results of Pipelined Adaptive Viterbi Decoder in case of transmission link with no bits errors.

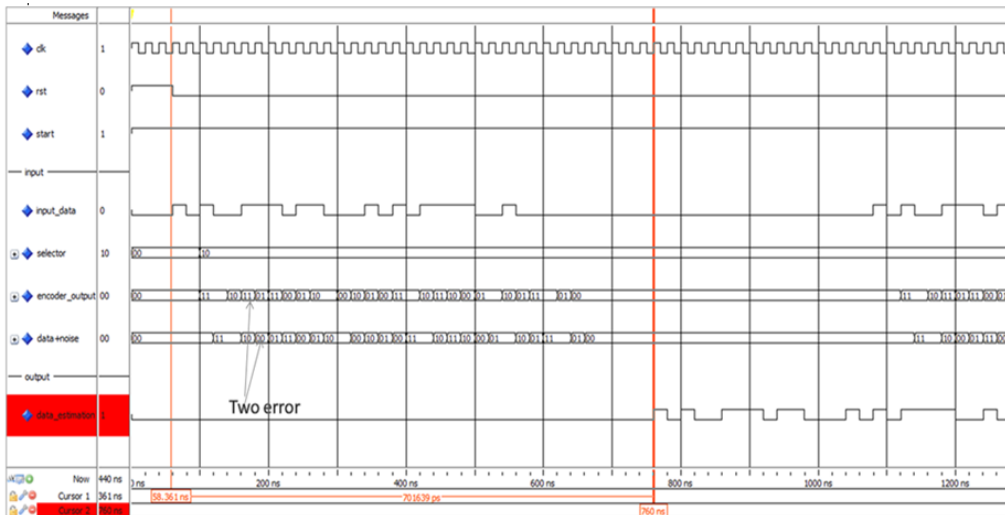


Figure (12): simulation results of Pipelined Adaptive Viterbi Decoder in case of transmission link with two bits errors.

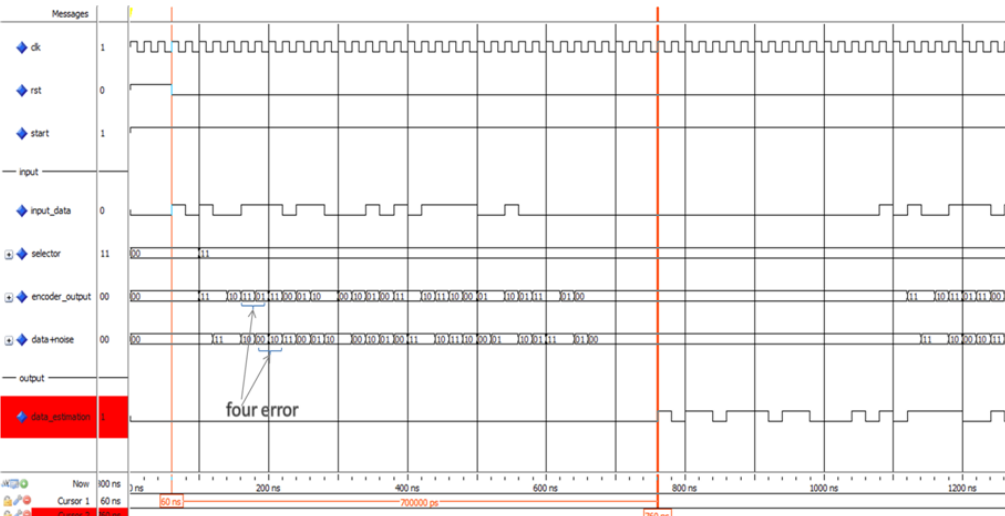


Figure (13): simulation results of Pipelined Adaptive Viterbi Decoder in case of transmission link with four bits errors.

As shown in the figures (11) to (13), the input to the Pipelined Adaptive Viterbi Decoder is the output of the convolution encoder which its input data is "101001110110001010111001". The output from the decoder is the same data that is encoded by the convolution encoder. Constraint length of this error corrector is 3 bits. We can apply any data with any length as an input data for the encoder. In order to check the working of this error corrector, four possible cases for the Multiplexer were investigated as shown in table (1).

Table (1): The Four Possible Encoded Transmission States.

Multiplexer Case	Selector bits	Data status
Case-1	"00"	Data will not be transmitted by the transmitter.
Case-2	"01"	Data will be transmitted without any error, as shown in fig (11).
Case-3	"10"	Data will be transmitted with error of 2 bits. The decoder will be able to detect this error of 2 bits and it will correct it, as shown in fig (12).
Case-4	"11"	Data will be transmitted with error of 4 bits. Since error is above correction capabilities, now this detector will not correct this error of four bits, as shown in fig (13).

V. Conclusion

The use of error-correcting codes has proven to be an effective way to overcome data corruption in digital communication channels. Because of the need for high-speed, low power for Viterbi decoding especially in wireless communication are always required.

In this paper A Pipelined Adaptive Viterbi Decoder is designed to reduce the processing execution time and increase the speed which is suitable for high speed applications like satellite communications and 3G applications when using the Viterbi algorithm as an error correcting code in the decoding of convolution codes. Using the Pipelined Adaptive Viterbi Decoder reduces the complexity of the decoder to be in the order of the real-time domain to avoid any losing in the input data. The design is coded in VHDL and implemented on a SPARTAN3 and I have been simulated it by using Model Sim. This approach has shown significant speedup versus both software implementations on previous FPGA-based implementations, the processing time is 700 ns (35 clock cycle) with operating clock 20 ns (50 MHz), as shown in the figures (11) to (13). The sequential algorithm processing time is 1100 ns (55 clock cycles) with operating clock 20 ns (50 MHz). Through experimentation, it was shown that Pipelined mechanism can be used effectively to improve the speed by 36.4% with respect to the sequential algorithm.

References

- [1]. D.RahulVarma , A. Uday Kumar, B. VijayaBhaskar, "Reconfigurable Viterbi Decoder" ,Department of ECE, St.Theresa College of Engg. and Tech., Andhra Pradesh, India, International Journal of Engineering Research and Applications (IJERA) ,Vol. 2, Issue 5, pp.124-127, September- October 2012.
- [2]. I. Kang, A. N. W. Jr, "Low-Power Viterbi Decoder for CDMA Mobile Terminals", IEEE J. Solid-State Circuits, vol. 33, no. 3, Mar. 1998.
- [3]. P. Subhashini, D. R. Mahesh Varma, Y. David Solomon Raju, "Implementation Analysis of adaptive Viterbi Decoder for High Speed Applications ",International Journal of Computer Applications (0975 – 8887) ,Vol. 31– No.2, October 2011.
- [4]. Dr. Dhafir A. Alneema, YahyaTaherQassim, "FPGA Based Implementation of Convolutional Encoder- Viterbi Decoder Using Multiple Booting Technique", Computer Engineering Dept. University of Mosul, Mosul, Iraq, Al-Rafidain Engineering Vol.18 No.6 December 2010.
- [5]. Saleem, S. et.al," Design and Tradeoff Analysis of an Area Efficient Viterbi Decoder", IEEE international conference, ISSN: 0-7803-9429-1 Vol.1, Issue.1, pp-1-5, march- 2007.
- [6]. Nitin S. Sonar, Dr. Faris S. Al-Naimy, Dr. R.R. Mudholkar, "An Improved Dynamically Reconfigurable Pipelined Adaptive Viterbi Decoder for High Throughput" International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 7, January 2013.
- [7]. Prof. Siddeeq Y. Ameen, Mohammed H. Al-Jammas, Ahmed S. Alenezi, "FPGA Implementation of Modified Architecture for Adaptive Viterbi Decoder", Electronics, Communications and Photonics Conference (SIEPC), 2011 Saudi International ,IEEE, Print ISBN: 978-1-4577-0068-2, INSPEC Accession Number: 12063553, April 2011
- [8]. SriramSwaminathan, Russell Tessier, Dennis Goeckel, and Wayne Burleson," A Dynamically Reconfigurable Adaptive Viterbi Decoder", ACM New York, NY, USA, ISBN: 1-58113-452-5 Order Number: 480020 ,24-2-2002.
- [9]. Master Of Technology In VLSI Design and CAD. Submitted By ,Suneha Gupta, "Design And Implementation Of An Optimized Viterbi Decoder" Roll No.601061025.Thapar University, Patiala ,July 2012.