IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 4, April 2023

DOI: 10.17148/IJARCCE.2023.124174

DESIGN AND IMPLEMENTATION OF CONVOLUTION ENCODER AND VITERBI DECODER

Neha K R[1], Nesara S Naik[2], Shreya Bijjur[3], Sinchana M Jagatap[4], Mrs Sumathi K[5]

Student, Dept. of Electronics and Communication Engineering, JNNCE, Shivamogga, India[1-4]

Assistant Professor, Dept. of Electronics and Communication Engineering, JNNCE, Shivamogga, India[5]

Abstract: Data transmissions over wireless channels are affected by attenuation, distortion, interference and noise, which affect the receiver's ability to receive correct information. Convolution encoding with Viterbi decoding is a powerful method for forward error correction. Convolution encoders and Viterbi decoders play an important role in digital communication especially, when channel is noisy and introduces errors in transmitted signal. The use of re-transmission methods is not efficient and has large latency measure up to the rising speed and data rates of communication links, the need of new techniques arise here to be compatible with those systems. Convolution encoding with forward error correction Viterbi decoding is designed. 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. The maximum likelihood detection of a digital stream is possible by Viterbi algorithm. In this paper, we present a Convolutional encoder and Viterbi decoder with a constraint length of 7 and code rate of 1/2. Implementation parameters for the decoder have been determined through simulation and the decoder should be implemented on a Xilinx FPGA Kit. Verilog HDL language is used as a design entry.

Keywords: Convolutional encoder, veterbi algorithm, decoder, FEC

I. INTRODUCTION

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

Viterbi decoding was developed by Andrew J. Viterbi in 1967 . Viterbi decoder with the throughput at the order of Giga-bit per second, without using off-chip processor(s) or memory can be realized using Advanced field programmable Array(FPGA) and well developed electronic design automatic (EDA) tools . The main purpose of this is to use VHDL, Synopsys synthesis and simulation tools to realize a Viterbi decoder having constraint length 7 aiming Xilinx FPGA technology In information theory and telecommunication, forward error correction (FEC) (also called channel coding) is a process of error detection and correction for data transmission, here transmitter adds redundant data to its messages, also known as an errorcorrecting code (ECC) these bits which are added up takes care of error detection at receiver end. FEC gives the receiver an ability to correct errors without needing a reverse channel to request retransmission of data.FEC is therefore applied in situations where retransmissions are relatively costly, or impossible such as when broadcasting to multiple receivers.There are three alternative methods that are often used to describe the convolutional code.These are the tree diagram, state diagram and trellis diagram.

There exist four basic convolutional codes decoding techniques: sequential, threshold, maximal likelihood and the Viterbi algorithm. The sequential algorithm can provide very strong correcting capabilities while it needs relatively large memory, which strongly depends on communication channel error density. The threshold algorithm is extensively good for channels with mid to good signal to noise ratios (SNR). The Viterbi algorithm is an optimum decoding technique. It is optimum as it results in the minimum probability of error.Viterbi algorithm is a maximum likelihood algorithm and performs decoding by searching the minimum cost path in a weighted oriented graph, named trellis. The basic building

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 4, April 2023

DOI: 10.17148/IJARCCE.2023.124174

blocks of Viterbi decoder are Branch Metric Unit (BMU), Path Metric Unit (PMU), Add Compare and Select Unit (ACSU) and Survivor Memory Management Unit (SMU). Convolution coding with Viterbi decoding is a FEC technique is used to that channel which is corrupted by additive white Gaussian noise (AWGN). Convolutional code definition parameters are the following: code rate (r), generating polynomial g (n), and number of input bits (k), number of output bits (n) and constraint length (K). We shall assume a 8-bit and give as an input to the convolutional encoder rate 1/2 code (two output bits for every input bit). This was 2x8 output matrix, with the extra bits allowing for the correction. This 16 bit output is given as the input to the viterbi decoder which decodes the convolutional codes into original data

II. METHODOLOGY

2.1 Block Daigram

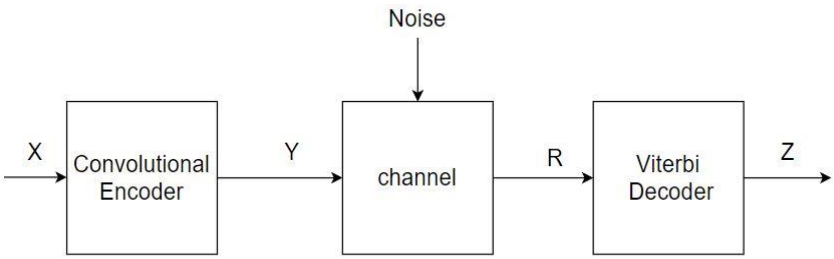

Figure 1: Block diagram of convolutional. Encoder and viterbi decoder.

2.1 Convolutional Encoder

A convolutional code generates a codeword of n symbols from some consecutive message blocks of k symbols.Generally, a convolutional encoder consists of k m-stage shift registers containing message symbols and circuits that perform some linear function to generate the codeword, as shown in Fig. 1.6.1. The message symbols are fed into the shift registers k symbols at a time. The contents of the shift registers are shifted each time a ksymbol block is fed in. The linear function generator generates an n-symbol block from both the contents of the shift registers and the k input symbols each time an input block is fed in.Initially the contents of the shift registers are assumed to be all zeros. The number of message symbols from which an output block is generated (= (m + 1)k) is called the constraint length of the code. The ratio R = k/n is called the code rate or coding rate, which is consistent with the definition of the code rate for a block code.

.

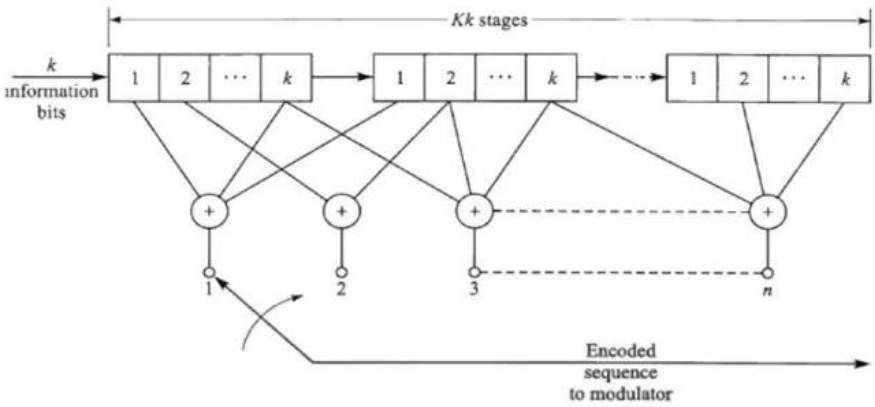

Figure 2:General block diagram of Convolutional Encoder.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 4, April 2023

DOI: 10.17148/IJARCCE.2023.124174

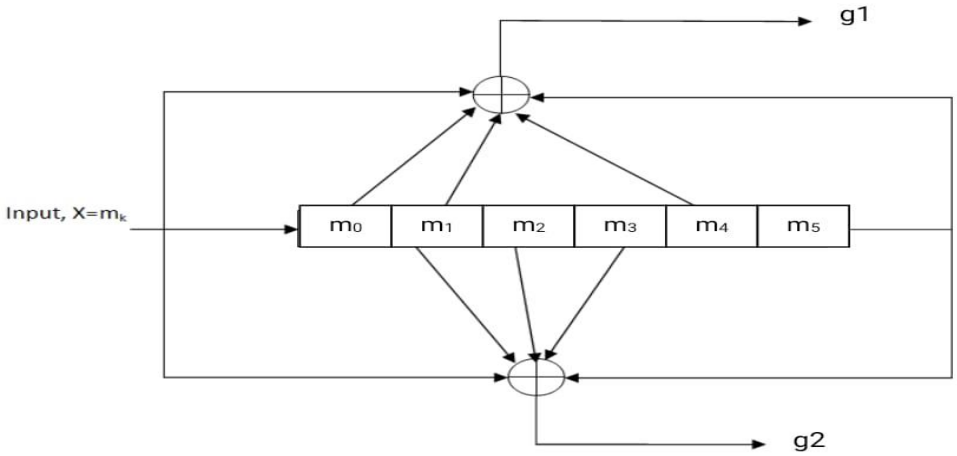

Figure 3: Convolution encoder of constraint length k=7

2.1.1: Ciruit description:

In Convolutional encoder, the following notations are used.

- N= Number of output bits produced for each k-bit sequence
- k = Number of input bits fed into the encoder at a time.
- m = number of stages of shift register.
- L = number of bits in a message sequence.
- j = number of modulo 2 adders.

2.1.2: Convolution encoder algorithm:

1. Initialize the Memory Registers with zeros on reset.
2. Store the incoming bit in memory register m − in. m − in = data − in .
3. After the input bit has arrived and datain is valid the operation starts and the output is calculated according to the generator polynomial.
4. Perform shifting operation. m1 = m − in and m(n) = m(n − 1) .
5. Steps 2, 3 and 4 are repeated for the length of input data bits.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 4, April 2023

DOI: 10.17148/IJARCCE.2023.124174

2.1.3: Flowchart:

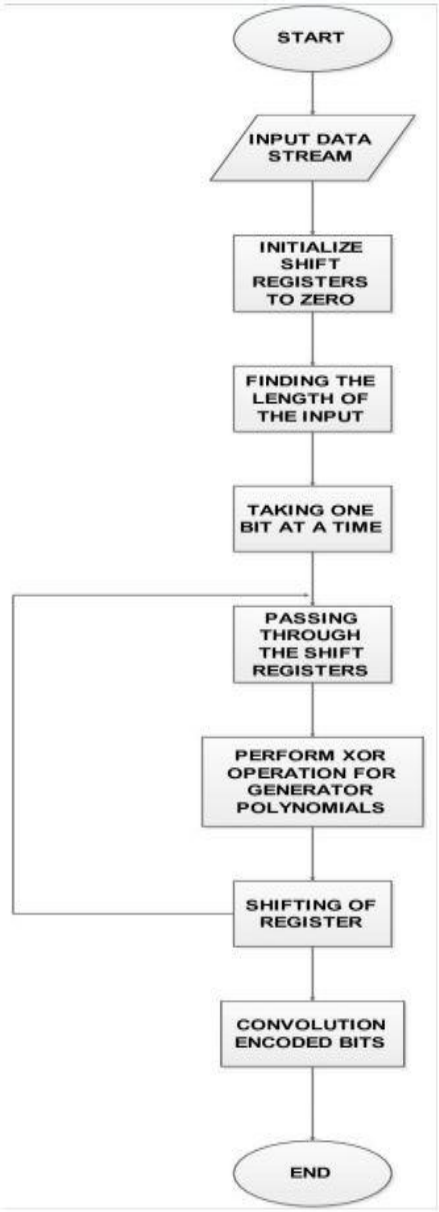

Figure 4: Flowchart of convolutional encoder

2.2: Trellis

The trellis diagram of the convolutional encoder gives us the information that "how each possible input to the encoder influences both the output and the state transitions of the encoder". From each trellis node, corresponding to a value of the state Sk and noted by a point in Figure, leave two branches associated to the presence of a "1" symbol (dotted) and of a "0" symbol (solid) at the encoder input respectively.

The succession of branches constitutes a path in the trellis diagram; each path is associated to particular sequence transmitted by the encoder (here, sequence indicates the sequence of coded symbols transmitted by the encoder, placed in a series).

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 4, April 2023

DOI: 10.17148/IJARCCE.2023.124174

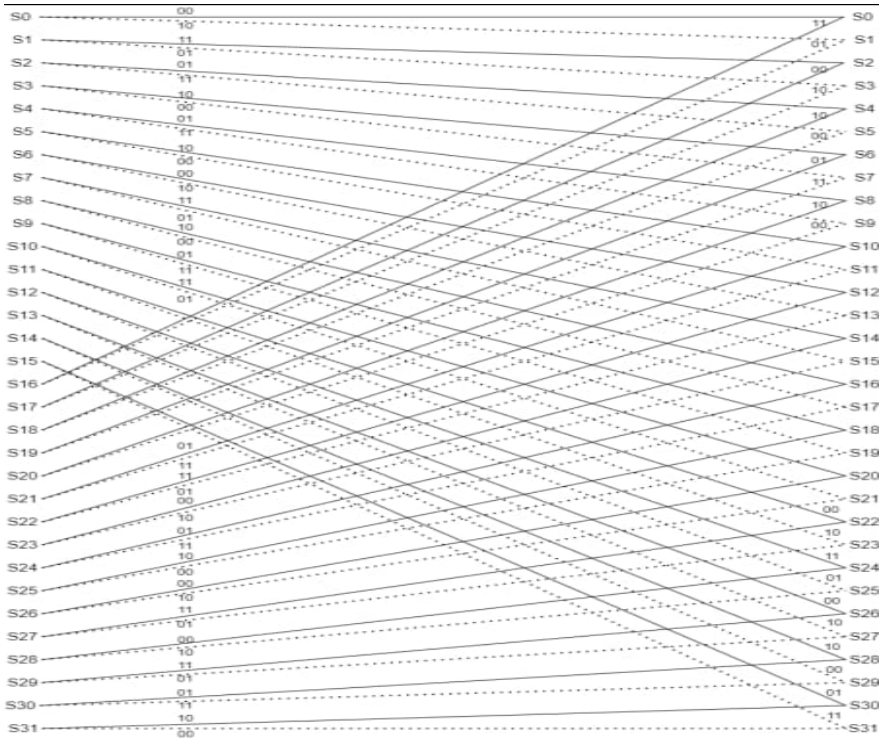

Figure 5: Trellis of constraint length 7

2.3 Viterbi Decoder

Viterbi decoding was developed by Andrew J. Viterbi. A Viterbi decoder uses the Viterbi algorithm for decoding a bit stream that has been encoded using a Convolution code or trellis code. There are other algorithms for decoding a convolution encoded stream. The Viterbi algorithm is the most resource-consuming, but it does the maximum likelihood decoding. It is most often used for decoding convolution codes with constraint lengths k<=3, but values up to k=15 are used in practice.There are both hardware (in modems) and software implementations of a Viterbi decoder. Viterbi decoding is used in the iterative Viterbi decoding algorithm.

In communication systems, forward error correction is an effective error control method for data transmission. It is used to improve the capacity of channel by adding redundant data in the controlled manner to original messages. The receiver uses redundant data to correct errors without asking the sender to re-transmit more additional data .The Viterbi algorithm belongs to the maximum likelihood decoding category. It drops the least likely trellis path at each transmission stage. In this way, it decreases decoding complexity with early rejection of unlike paths and gets efficiency via concentrating on survival paths of the trellis .The receiver receives the serial encoded data and converts it into 3-bit parallel data. Trellis generator generates the predefined sequence. It is used by the branch metric unit to calculate survival shortest path. Viterbi decoder used at the receiver side decodes the data.

The Viterbi decoder calculates a semi-brute-force estimate of the likelihood for each path through the trellis Once the estimates for all states in a step/iteration of the trellis have been calculated, the probabilities for all previous steps/iterations can be discarded; only the most likely entry to a state must be remembered. It comprises four basic steps:

- Calculate the trellis.

a) Weight the trellis branches by calculating branch metrics.

b) Compute the minimum weight path to time n+1 in terms of the minimum weight path to time n. Retain step decisions. Uses add-compare-select(ACS) algorithm.

- Find the last state of the minimum weight path.
- Find the entire minimum weight path. Also called survivor path decode or trace back.
- Reorder bits into correct forward ordering.

2.3.1: Block diagram:

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 4, April 2023

DOI: 10.17148/IJARCCE.2023.124174

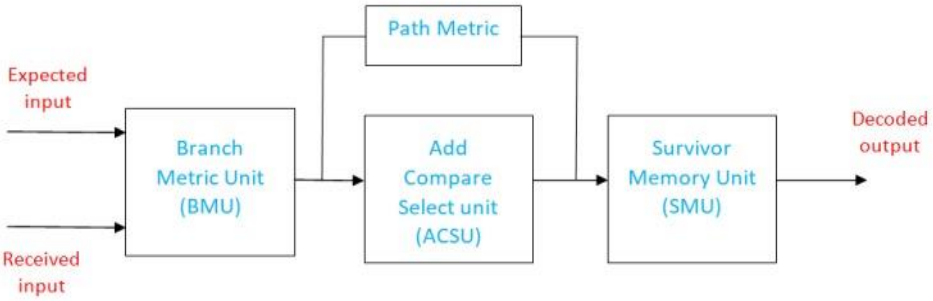

Figure 6: Block diagram of Viterbi decoder

A) Branch Metric Unit

- A branch metric unit's function is to calculate branch metrics, which are normed distances between every possible symbol and the received symbol.
- The comparison between received code symbol and expected code symbol is done by branch metric unit. • It also counts the number of differing bits.It is the smallest unit in the Viterbi decoder.
- The measured value of the BMU can be the Hamming distance in case of the hard input decoding or the Euclidean distance in case of the soft input decoding.

B) Path Metric Unit

- The partial path metric of each branch is computed by the use of two adders.
- The partial path metric is compared by the comparator and an appropriate branch is selected by the selector.
- The selector selects the smaller value and stored hat value as the new path metric for each state.
- The corresponding bit decision is transferred to the SMU.

C) Survivor Memory Unit

- The Survivor Memory Unit receives the bit decision from the PMU. This will produces the decoded sequence.
- After finishing the SMU, it is important to perform the trace back module.
- When the trellis diagram is finished, the trace-back module will search the maximum likelihood path from the final state which is state0 to the beginning state which is state0.
- Each time, the trace-back block just left shifts one bit of the binary state number and add one bit from the survivor path metric to compute the previous state
- It stores the path information in the form of an array of recursive pointers. By doing this, the most likelihood path is found.

2.3.2: Viterbi Decoding Algorithm:

- Initialization: Set the all-zero state of the trellis to zero.
- Step 1:Start the computation at some time-unit j and determine the metric for the path that enters each state of the trellis. Hence, identify the survivor and store the metric for each one of the states.
- Step 2: For the next time-unit j+1 , determine the metrics for all 2 L−1 paths that enters a state where L is a constraint length of the convolutional encoder: hence do the following:

1. Add the metrics entering the state to the metric of the survivor at the preceding timeunit j.
2. Compare the metrics of all 2 L−1 paths entering the state.
3. Select the survivor with the largest mertic, store it along with its metric, and discard all other paths in the trellis.

- Step 3: Repeat step 2 for time-unit j< N+N' , where N is the length of the message sequence and N' is the length of the termination sequence.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 4, April 2023

DOI: 10.17148/IJARCCE.2023.124174

2.3.3: Flowchart:

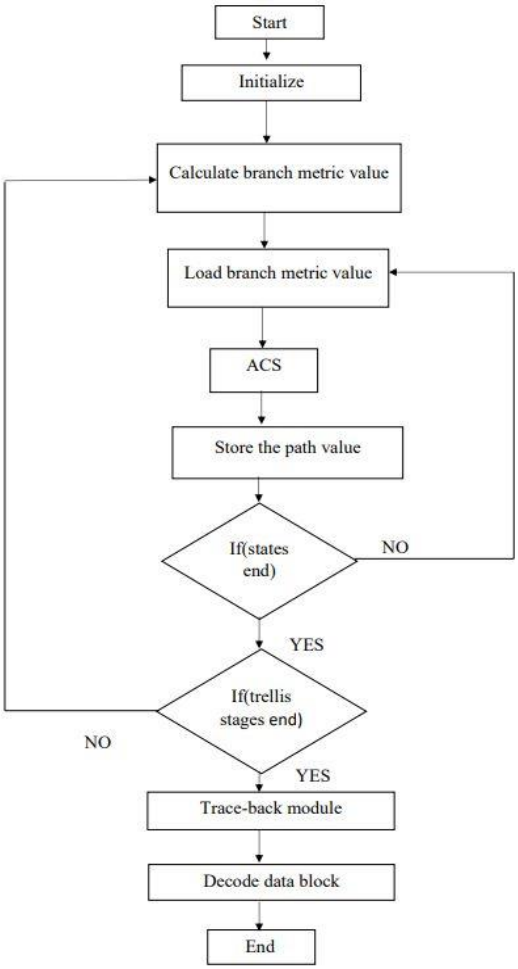

Figure 7: Flowchart of viterbi decoder

2.4 Hardware Used:

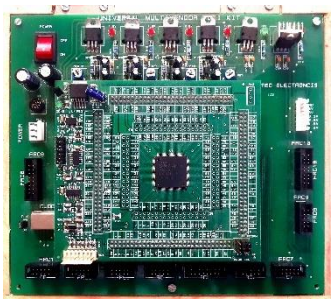

Figure 8: Spartan6 FPGA

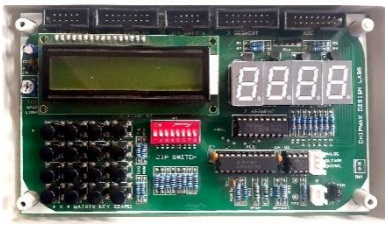

Figure 9: External Interface Board

Figure 10: USB Programmer FPGA

FPGAs were invented by Xilinx in 1984. They are structured like the gate arrays form of Application Specific Integrated Circuits (ASIC). However, the architecture of the FPGAs architecture consists of a gate-array-like architecture, with configurable logic blocks, configurable I/o blocks, and programmable interconnect. Additional logic resources such as ALUs, memory, and decoders may also be available. The three basic types of programmable elements for an FPGA are static RAM, anti-fuses, and flash EPROM. Segments of metal interconnect can be linked in an arbitrary manner by programmable switches to form the desired signal nets between the cells.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 4, April 2023

DOI: 10.17148/IJARCCE.2023.124174

FPGAs can be used in virtually any digital logic system and provide the benefits of high integration levels without the risks of expenses of semicustom and custom IC development.FPGAs give us the advantage of custom functionality like the Application Specific Integrated Circuits while avoiding the high development costs and the inability to make design modifications after production. The FPGAs also add design flexibility and adaptability with optimal device utilization while conserving both board space and system power. The gate arrays offer greater device speed and greater device density. Taking these into consideration, FPGAs are especially suited for rapid prototyping of circuits, and for production purposes where the total volume is low.

Spartan™ 6 devices offer industry-leading connectivity features such as high logic-to-pin ratios, small form-factor packaging, MicroBlaze™ soft processor, and a diverse number of supported I/O protocols. Ideally suited for a range of advanced bridging applications found in consumer, automotive infotainment, and industrial automation.

III. RESULTS AND DISCUSSIONS

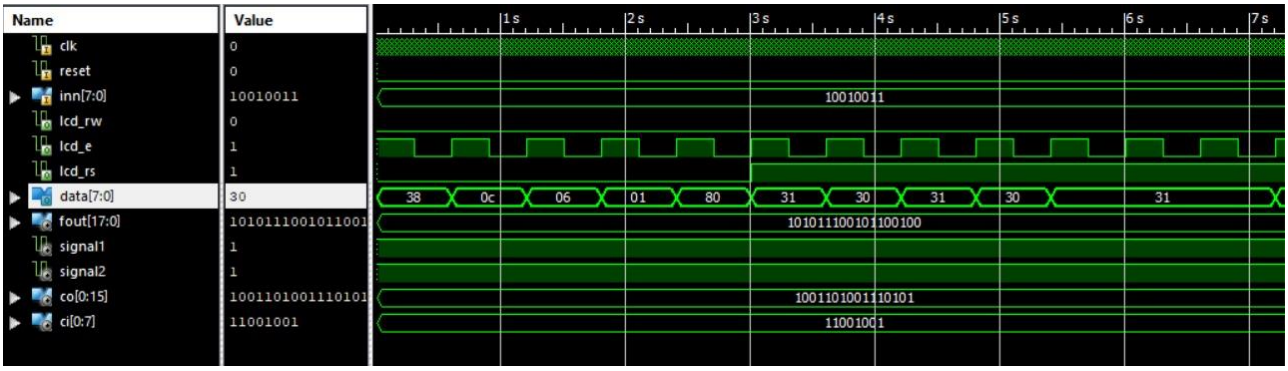

Figure 11: Output Waveform of convolutional encoder and Viterbi decoder

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	2996	126800	2%
Number of Slice LUTs	31101	63400	49%
Number of fully used LUT-FF pairs	1938	32159	6%
Number of bonded IOBs	21	210	10%
Number of BUFG/BUFGCTRL/BUFHCEs	1	128	0%

Figure 12: Device Utilization

IV. CONCLUSION

The high growth of the semiconductor industry over the past two decades has put Very Large-Scale Integration in demand all over the world. Digital Signal Processing has played a great role in expanding VLSI device area. The FFT is one of the most widely used algorithms for calculating the Discrete Fourier Transform (DFT) owing to its efficiency in reducing computation time. The work chosen for this project is to Implement FFT in FPGA using Verilog HDL. The reason for selecting this project is Fast Fourier Transform (FFT) has been used in a wide range of applications, such as wide- band mobile digital communication system based onorthogonal Frequency Division Multiplexing (OFDM) principle. The objective of this project is to implement a FFT Architecture using Verilog HDL. The Architecture that is going to be implemented is Combined SDC-SDF. Standard FPGA Flow is adapted to implement this project. i.e. right from specification to bit file generation, which is going to be programmed on Spartan6 FPGA. Simulations and Synthesis will be done using Xilinx ISE.

We propose a combined SDC-SDF pipelined FFT architecture which produces the output data in the normal order. The proposed SDC PE mainly reduces 50% complex multipliers, compared with the other radix-2 FFT designs. Therefore, the proposed FFT architecture is very attractive for the single-path pipelined radix-2 FFT processors with the input and output sequences in normal order.

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Peer-reviewed / Refereed journal ⁜ Vol. 12, Issue 4, April 2023
DOI: 10.17148/IJARCCE.2023.124174

REFERENCES

[1]. Z. Wang, X. Liu, B. He, and F. Yu, "A combined sdc-sdf architecture for normal i/o pipelined radix-2 fft," IEEE Transactions on very large scale integration (VLSI) systems, vol. 23, no. 5, pp. 973–977, 2014
[2]. M. Garrido, R. Andersson, F. Qureshi, and O. Gustafsson, "Multiplierless unity-gain sdfffts," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 9, pp. 3003–3007, 2016
[3]. X.-Y. Shih, Y.-Q. Liu, and H.-R. Chou, "48-mode reconfigurable design of sdf fft hardware architecture using radix-3 2 and radix-2 3 design approaches," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 64, no. 6, pp. 1456–1467, 2017
[4]. C. Ingemarsson, P. Källström, F. Qureshi, and O. Gustafsson, "Efficient fpga mapping of pipeline sdf fft cores," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 9, pp. 2486–2497, 2017.
[5]. Y. Chandu, M. Maradi, A. Manjunath, and P. Agarwal, "Optimized high speed radix-8 fft algorithm implementation on fpga," in 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, 2018, pp. 430–435