ISSN 2277-2685
IJESR/Jan-Mar. 2025/ Vol-15/Issue-1s/318-327
S. Bhavani *et. al.*, / International Journal of Engineering & Science Research

Low Power Viterbi Decoder Design Based On Reversible Logic Gates

[1]N Sony, [2]S. Bhavani, [3]Esha Jain, [4]B. Hema Sri
[1]Assistant professor, Electronics and Communication Engineering, BRECW
[2,3,4]B.Tech Students, Department of Electronics and Communication Engineering, BRECW

ABSTRACT

In 5G mobile terminals the Viterbi decoder consumes approximately one third of the power consumption of a basic band mobile transceiver. Viterbi decoders employed in digital wireless communications are complex and dissipate large power. In this project, to reduce the power consumption, and to increase the speed, an asynchronous technique that is delay insensitive null convention logic (NCL) for Viterbi decoder using dual rail signal is proposed. NCL reduces the dynamic power consumption in terms of reducing the switching activity and it also reduces the glitch power significantly, thereby achieving the lower power.

The Viterbi decoder consists of branch metric unit, add compare, and select unit and the survivor path memory unit. Viterbi algorithm is an effective implementation of a discrete-time finite state Markov process perceived in memoryless noise and optimality can be achieved by following the maximum-likelihood criteria.

1- INTRODUCTION

VITERBI ALGORITHM

The Viterbi algorithm was introduced in 1967 as an efficient method for decoding convolutional codes widely used in communication systems. This algorithm is utilized for decoding the codes used in various applications including satellite communication, cellular, and radio relay. It has proven to be an effective solution for a lot of problems related to digital estimation. Moreover, the Viterbi decoder has practical use in implementations of high-speed (5 to 10 Gb/s) serializer- deserializers (SERDESs) which have critical latency constraints. SERDESs can be further used in local area and synchronous optical networks of 10 Gb/s. Furthermore, they are used in magnetic or optical storage systems such as hard disk drive or digital video disk.

The Viterbi algorithm process is similar to finding the most-likely sequence of states, resulting in sequence of observed events and, thus, boasts of high efficiency as it consists of finite number of possible states. It is an effective implementation of a discrete-time finite state Markov process perceived in memoryless noise and optimality can be achieved by following the maximum- likelihood criteria. It helps in tracking the stochastic process state using an optimum recursive method which helps in the analysis and implementation.

A top-level architecture for Viterbi decoders is shown in Figure. 1.1. As seen in this Figure, Viterbi decoders are composed of three major components: branch metric unit (BMU), add- compare-select (ACS) unit, and survivor path memory unit (SMU). BMU generates the metrics corresponding to the binary trellis depending on the received signal, which is given as input to ACS which, then, updates the path metrics. The survival path is updated for all the states and is stored in the additional memory. SMU is responsible for managing the survival paths and giving out the decoded data as output

BMU and SMU units happen to be purely forward logic. ACS recursion consists of feedback loops;

hence, its speed is limited by the iteration bound. Hence, the ACS unit becomes the speed bottleneck for the system. M-step look-ahead technique can be used to break the iteration bound of the Viterbi decoder of constraint length K . A lookahead technique can combine several trellis steps into one trellis step, and if M > K, then throughput can be increased by pipelining the ACS architecture, which helps in combines binary trellis of multiple-steps into a single complex trellis of one-step. BMP dominates the overall latency and complexity for deep look-ahead architectures. Before the saturation of the trellis, only add operation is needed. After the saturation of the trellis, add operation is followed by compare operation where the parallel paths consisting of less metrics are discarded as they are considered unnecessary.

2- Literature Survey

The FPGA industry sprouted from programmable read-only memory (PROM) and programmable logic devices (PLDs). PROMs and PLDs both had the option of being programmed in batches in a factory or in the field (field-programmable). However, programmable logic was hard-wired between logic gates. In the late 1980s, the Naval Surface Warfare Center funded an experiment proposed by Steve Casselman to develop a computer that would implement 600,000 reprogrammable gates. Casselman was successful and a patent related to the system was issued in 1992. Some of the industry's foundational concepts and technologies for programmable logic arrays, gates, and logic blocks are founded in patents awarded to David W. Page and Luverne R. Peterson in 1985.

Altera was founded in 1983 and delivered the industry's first reprogrammable logic device in solving the problem of iteration bound, and is frequently used in high-speed communications. Branch metric precomputation (BMP) which is in the front end of ACS is resulted due to the look-ahead technique and it dominates the overall complexity and latency for deep look- ahead architectures. BMP consists of pipelined registers between every two consecutive steps and

1984 – the EP300 – which featured a quartz window in the package that allowed users to shine an ultra-violet lamp on the die to erase the EPROM cells that held the device configure ration. As of December 2015, it was acquired by Intel. Xilinx co-founders Ross Freeman and Bernard Vonderschmidt invented the first commercially viable field-programmable gate array in 1985 – the XC2064. The XC2064 had programmable gates and programmable interconnects between gates, the beginnings of a new technology and market. The XC2064 had 64 conFigureurable logic blocks (CLBs), with two three-input lookup tables (LUTs). More than 20 years later, Freeman was entered into the National Inventors Hall of Fame for his invention.

Altera and Xilinx continued unchallenged and quickly grew from 1985 to the mid-1990s, when competitors sprouted up, eroding significant market share. By 1993, Actel (now Microsemi) was serving about 18 percent of the market. By 2013, Altera (31 percent), Actel (10 percent) and Xilinx (36 percent) together represented approximately 77 percent of the FPGA market. The 1990s were a period of rapid growth for FPGAs, both in circuit sophistication and the volume of production. In the early 1990s, FPGAs were primarily used in telecommunications and networking. By the end of the decade, FPGAs found their way into consumer, automotive, and industrial

applications.

BINARY GROUPING (BBG) APPROACH: This section focuses only on branch metric computation, leaving aside the operations of compare-and-discard. An optimal approach of BBG metrics computation is said to be carried out sequentially for a conventional Viterbi decoder. When two consecutive binary-trellis steps are combined, for each state, there are two incoming and two outgoing branches, and the computational complexity is 4 × N. As the results do not depend on the order of the trellis combination, the way the trellis steps are grouped and combined helps in determining the computational complexity. The combination in a backward nested procedure can be ex-plained as follows. The main M-step trellises are divided into two groups consisting of m0 and m1 trellis steps. The binary decomposition on each subgroup goes on till it becomes a single trellis step. The decomposition helps in removing maximum possible redundancy and, thus, helps achieve minimum delay and complexity. Finally, it can be verified that the complexities involved in the BBG approach are less as compared to the ones in the intuitive approach.

LOOK-AHEAD-BASED LOW-LATENCY ARCHITECTURES: This approach is a highly-efficient design approach based on the BBG scheme for a general M which provides less or equal latency, and also has much less complexity compared to other existing architectures [3]. For constraint length K and M-step lookahead, the execution of BMP is done in a layered manner. An M-step trellis is a bigger group consisting of M K sub-groups with a trellis of K-step. Thus, the total numbers of P1 processors needed are M K and each P1 is responsible for computing K- step trellises. is taken into consideration in order to remove all redundancies which are usually responsible for longer delay and extra complexity, since various paths share common computations. Branch

Accordingly, we have the complexities and latencies of P1 and P2 as Comp.P1 = N (∑k i=2 2 i) + N2, Comp.P2 = N2 (N − 1) + N3, and Lat.P1, P2 = K, where N = 2k−1 is the number of trellis states. For P1 processors, the complexity of add operation is N ∑k= 2i and that of the “compare” operation is N2. Similarly, for P2 i2processors, the complexity of add operation is N2 (N −1) and that of the compare operation is N3. For both P1 and P2 processors, the latency is same, i.e., K; however, the complexity of P2 is larger than that of P1. As the BBG approach is very efficient in computing the branch metrics, more operations of trellis combination can be allotted into BBG-based P1 processors in order to reduce the number of P2 processors as they are expensive in terms of complexity. The trellis Steps L, which is computed in the P1 processors, has the constraint of being less than 2 ×K in order to make sure that the latency feature is not lost. The number of groups Ng can be determined by Ng = 2 ⌊ log2(MK)⌋ . The overall layered structure of the Viterbi algorithm is shown in Figure. 2.1 (in this Figure, i, j ∈ [1, N] and l ∈ [1, K]). As seen in this Figure, within two layers (shown by Layer 1 and Layer 2 in Figure. 2.1), we have Ng steps, going through P1 and P2 processors. In each L-level P1 processor, the initial step combination is performed using the BBG approach, followed by concatenated add–compare operations executed one step at a time for the remaining L−K-step phase-II computation. In Layer 2, the outputs of P1 processors are combined for computing the final equivalent complex trellis.

This Figure also shows the P1 processor

Layer 1, although P1 leads to longer latency, as the depth of Layer 2 is reduced as well, latency penalty in not incurred.

ALTERNATIVE TO LOOK-AHEAD APPROACH As the state nodes are connected pairwise, there are a total of N2 connections, consisting of 2(M−K+1) parallel paths. The number of parallel paths increases exponentially with respect to M, thereby, increasing the complexity. Generally, the exponential increase of parallel paths is avoided by a compare operation performed in each binary-trellis steps combination, thus, the parallel paths with less metrics are always discarded. Nonetheless, each of such add-compare operations results in a substantial amount of latency. The complexity efficiency of look-ahead depends on constraint length of Viterbi decoder. For larger constraint lengths, latency reduction is achieved at the expense of prohibitive computational complexity which limits the application of look-ahead- based architectures.

3-SOFTWARE REQUIREMENTS

XILINX ISE OVERVIEW: The Integrated Software Environment (ISE™) is the Xilinx® design software suite that allows you to take your design from design entry through Xilinx device programming. The ISE Project Navigator manages and processes your design through the following steps in the ISE design flow.

DESIGN ENTRY: Design entry is the first step in the ISE design flow. During design entry, you create your source files based on your design objectives. You can create your top-level design architecture based on the BBG algorithm. In file using a Hardware Description Language (HDL), such as VHDL, Verilog, or ABEL, or using a schematic. You can use multiple formats for the lower-level source files in your design.

SYNTHESIS: After design entry and optional simulation, you run synthesis. During this step, VHDL, Verilog, or mixed language designs become netlist files that are accepted as input to the implementation step.

IMPLEMENTATION: After synthesis, you run design implementation, which converts the logical design into a physical file format that can be downloaded to the selected target device. From Project Navigator, you can run the implementation process in one step, or you can run each of the implementation processes separately. Implementation processes vary depending on whether you are targeting a Field Programmable Gate Array (FPGA) or a Complex Programmable Logic Device (CPLD).

4-LOW POWER VITERBI DECODER BASED ON REVERSIBLE LOGIC GATES

Existing System

The existing system of low power Viterbi decoders is designed using a combination of logic gates, which are fundamental building blocks of digital circuits. These logic gates, such as AND, OR, and NOT gates, perform logical operations on binary inputs to produce a desired output. In the context of low power Viterbi decoders, the goal is to minimize power consumption while maintaining efficient video or audio signal decoding. This is achieved through careful design and optimization of the circuitry. One approach to reducing power

consumption is by using low- power variants of logic gates, which are specifically designed to operate with lower power requirements. These gates are designed to minimize leakage currents, reduce switching power, and optimize transistor sizing to achieve better power efficiency. Another technique used in low power Viterbi decoders is power gating, where certain parts of the circuitry are selectively powered off when not in use. This helps to conserve power by reducing unnecessary power consumption in idle or unused components. Furthermore, advanced power management techniques, such as voltage scaling and clock gating, are employed to dynamically adjust the voltage and clock frequency of the circuitry based on the processing requirements. This allows for power savings during periods of low computational demand. Overall, the design of low power Viterbi decoders involves a careful balance between performance, power consumption, and efficiency. By utilizing optimized logic gates, power gating,

and intelligent power management techniques, these decoders can achieve efficient signal decoding while minimizing power usage. It's a fascinating field of study that aims to make our digital devices more energy-efficient.

Proposed System

It is well-known that in different variants of concurrent error detection, either redundancy in hardware, i.e., increase in area/power/energy consumption, e.g., through error detection codes such as hamming codes, or redundancy in time, adding negligible area overhead at the expense of higher total time (throughput and latency), is performed. In this thesis, we utilize recomputing with encoded operands, where, the operations are redone for different operands for detecting errors. During the first step, operands are applied normally. In the recomputed step, the operands are encoded and applied and after decoding, the correct results can be generated. Moreover, through signature-based schemes, we propose schemes through which both transient and permanent errors can be detected.

Block Diagram

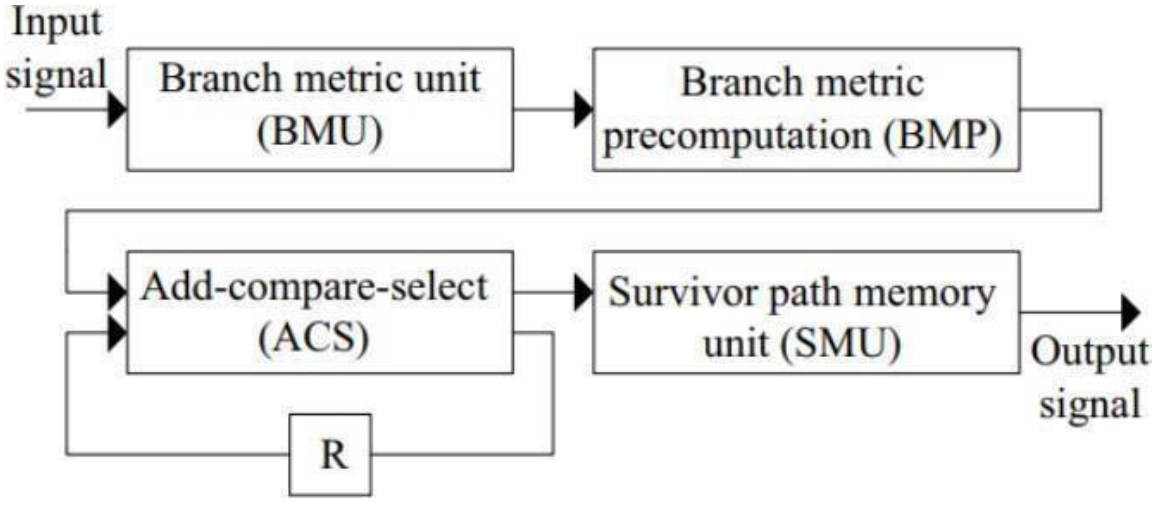

Figure 3.4: Viterbi decoder block diagram.

A top-level architecture for Viterbi decoders is shown in Figure. 1.1. As seen in this Figure, Viterbi decoders are composed of three major components: branch metric unit (BMU), add- compare-select (ACS) unit, and survivor path memory unit (SMU). BMU generates the metrics corresponding to the

binary trellis depending on the received signal, which is given as input to ACS which, then, updates the path metrics. The survival path is updated for all the states and is stored in the additional memory. SMU is responsible for managing the survival paths and giving out the decoded data as output. BMU and SMU units happen to be purely forward logic. ACS

5-ADVANTAGES, DISADVANTAGES AND APPLICATIONS

Advantages

1. **Error Correction:** Viterbi decoders are widely used in communication systems to correct errors introduced during transmission.
2. **Efficiency:** They offer high coding efficiency, making them suitable for various error- correcting codes, such as convolutional codes.
3. **Low Bit Error Rates**: Viterbi decoding helps achieve low bit error rates, crucial for reliable data transmission.

Disadvantages

Design and Implementation Disadvantages

1. Increased design complexity: Reversible logic gates require more complex design and layout.
2. Higher gate count: More gates are needed to implement reversible logic.
3. Difficulty in optimizing reversible logic gates: Optimizing reversible logic gates for power and performance is challenging.

Applications

- **Wireless Communication:** Viterbi decoders are extensively used in wireless communication systems, including cellular networks, to correct errors introduced during signal transmission.
- **Satellite Communication:** Applied in satellite communication systems to ensure reliable data transmission in the presence of noise and interference.
- **Digital Broadcasting:** Utilized in digital broadcasting standards like DVB (Digital Video Broadcasting) for error correction in TV and radio signals.
- **Mobile Networks:** Viterbi decoding plays a crucial role in mobile networks, enhancing the reliability of data transmission in technologies like GSM, CDMA, and LTE.

6-RESULTS

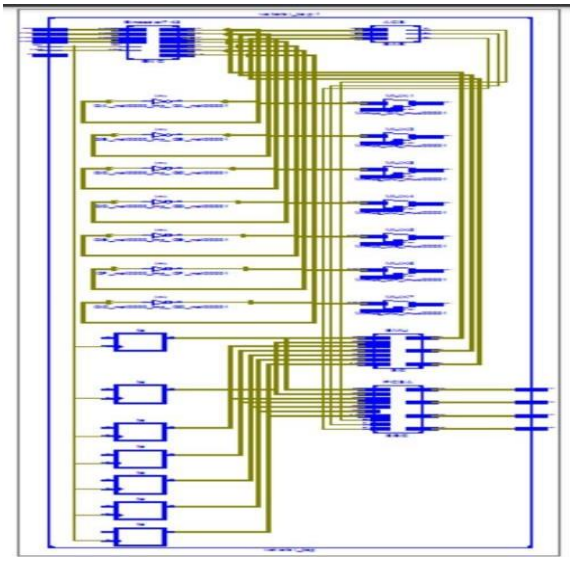

Figure 5.3.1: RTL Schematic

Table: 5.3.2 Design summary

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	91	16640	0%
Number of Slice Flip Flops	112	33280	0%
Number of 4 input LUTs	150	33280	0%
Number of bonded IOBs	69	309	22%
Number of GCLKs	1	24	4%

```
=========================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'Clock'
  Total number of paths / destination ports: 6272 / 32
-------------------------------------------------------------------------
Offset:              12.868ns (Levels of Logic = 6)
  Source:            OB_0 (FF)
  Destination:       YC1<7> (PAD)
  Source Clock:      Clock rising

  Data Path: OB_0 to YC1<7>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FD:C->Q               5   0.591   0.776  OB_0 (OB_0)
     LUT2:I0->O            1   0.648   0.452  SC/Mxor_S0_0_xo<0>21 (N111)
     LUT4:I2->O            1   0.648   0.563  EAB/e026 (EAB/e026)
     LUT4:I0->O           36   0.648   1.406  EAB/e0206 (e0)
     LUT4:I0->O            8   0.648   0.900  SEC/Mmux_YC31211 (SEC/N2)
     LUT4:I0->O            1   0.648   0.420  SEC/Mmux_YC39 (YC3_2_OBUF)
     OBUF:I->O                 4.520          YC3_2_OBUF (YC3<2>)
    ----------------------------------------
    Total                     12.868ns (8.351ns logic, 4.517ns route)
                                       (64.9% logic, 35.1% route)

=========================================================================
```

Figure 5.3.3: Time Summary

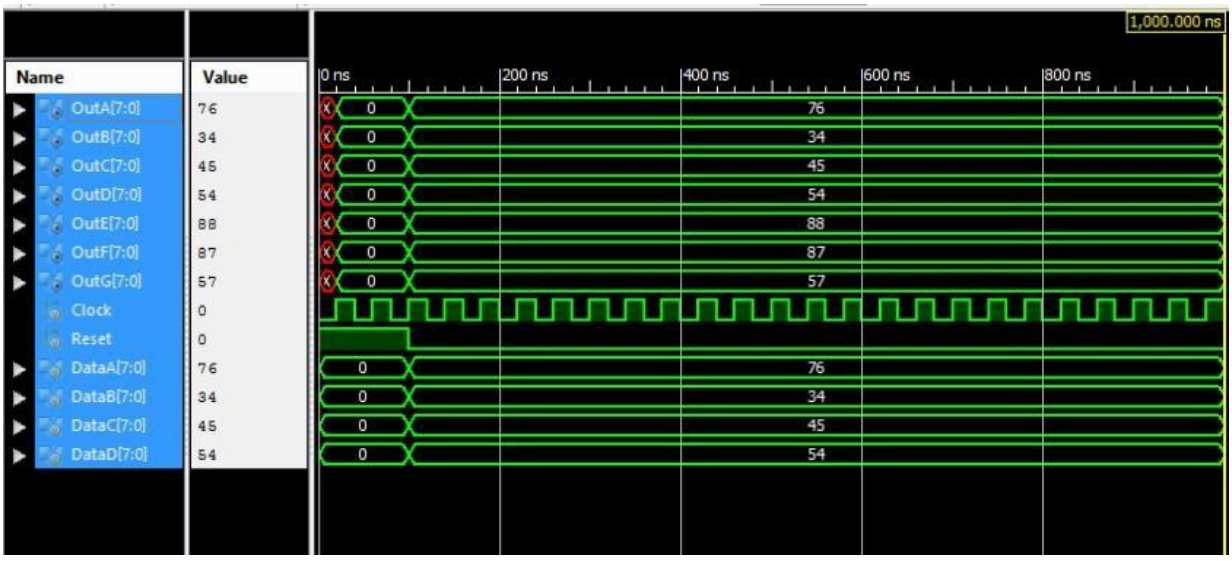

Figure 5.3.4: Encoder Output

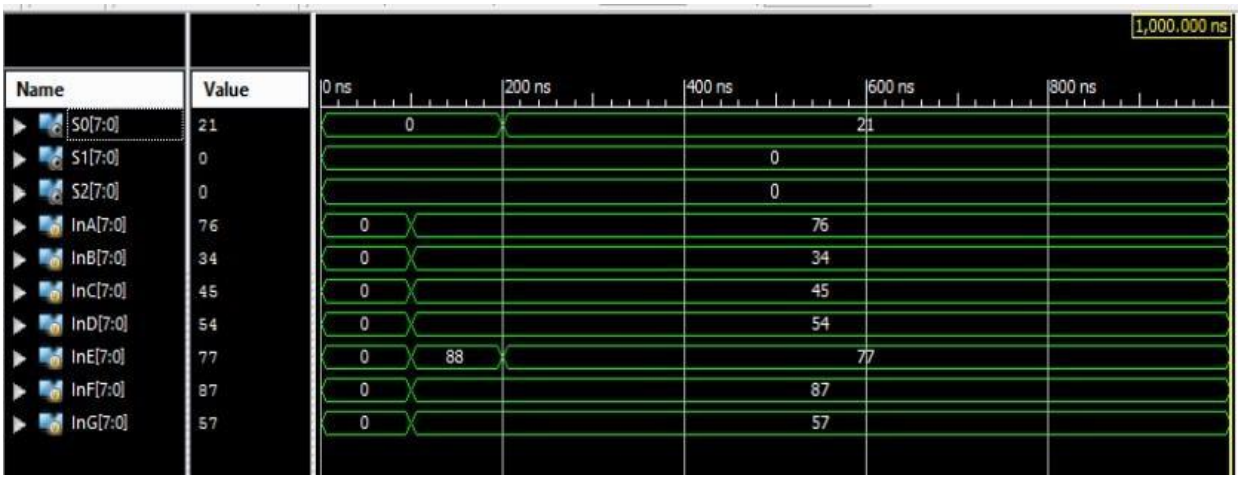

Figure 5.3.5: BMU Output

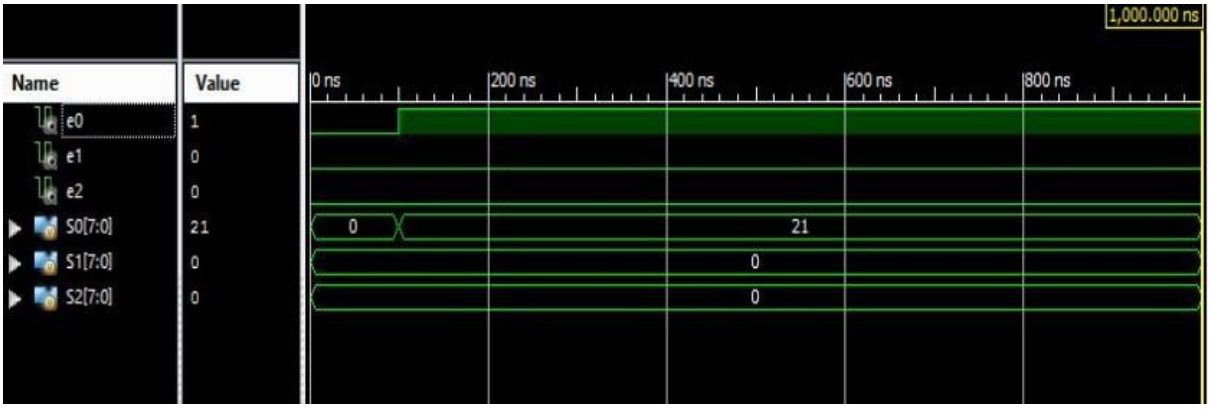

Figure 5.3.6: ACS Output

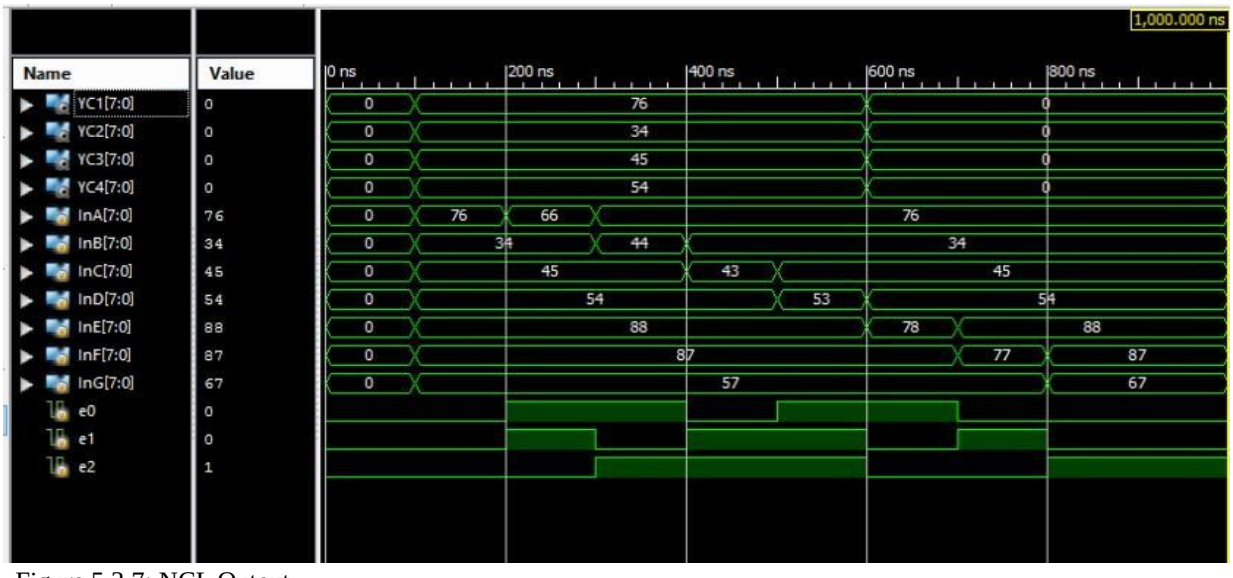

Figure 5.3.7: NCL Output

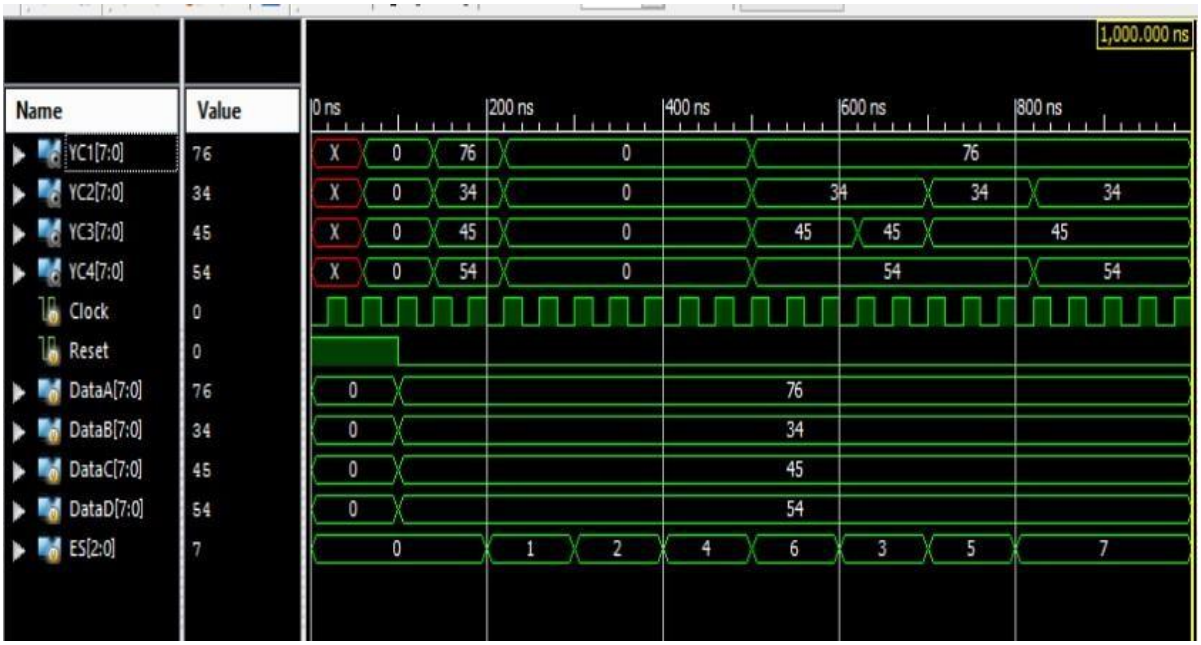

Figure 5.3.8: VITERBI Output

7-CONCLUSION

In this thesis, we presented fault diagnosis models for the CSA and NCL units of low complexity and low-latency Viterbi decoder. The simulation results for the proposed methods of RESO, RERO, modified RESO, parity and self-checking adder-based designs for both CSA and NCL units show very high fault coverage (almost 100 percent) for the randomly distributed injected faults. The proposed architectures have been successfully implemented on Xilinx Virtex-6 Family and also by using the 32nm library using Synopsys Design Compiler for the ASIC implementation. Also, the ASIC and FPGA implementation results show that overheads obtained are acceptable. Thus, the proposed models are reliable and efficient.

REFERENCES

[1] Massoud Pedram, "Power minimization in ic design: Principles and applications," ACM Trans. Des. Autom. Electron. Syst., vol. 1, no. 1, pp. 3–56, Jan. 1996.

[2]Qing Wu, M. Pedram, and Xunwei Wu, "Clock-gating and its application to low power design of sequential circuits," Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on, vol. 47, no. 3, pp. 415–420, Mar 2000.

[3]G.E. Tellez, A. Farrahi, and M. Sarrafzadeh, "Activity-driven clock design for low power circuits," in Computer-Aided Design, 1995. ICCAD-95. Digest of Technical Papers., 1995 IEEE/ACM International Conference on, Nov 1995, pp. 62–65.

[4]E. Lee and A. Sangiovanni-Vincentelli, "Comparing models of computation," in Proceedings of the 1996 IEEE/ACM international

conference on Computer-aided design. IEEE Computer Society, 1997, pp. 234–241.

[5]Gilles Kahn, "The Semantics of Simple Language for Parallel Programming," in IFIP Congress, 1974, pp. 471–475.

[6]Edward A. Lee and David G. Messerschmitt, "Static scheduling of synchronous data flow programs for digital signal processing," IEEE Trans. Comput., vol. 36, no. 1, pp. 24–35, 1987.

[7]E.A. Lee and T.M. Parks, "Dataflow process networks," Proceedings of the IEEE, vol. 83, no. 5, pp. 773 –801, may 1995.

[8]Syed Suhaib, Deepak Mathaikutty, and Sandeep Shukla, "Dataflow architectures for GALS," Electronic Notes in Theoretical Computer Science, vol. 200, no. 1, pp. 33– 50, 2008.

[9]Tzyh-Yung Wuu and Sarma B. K. Vrudhula, "Synthesis of asynchronous systems from data flow specification," Research Report ISI/RR-93-366, University of Southern California, Information Sciences Institute, Dec 1993.

[10] Behnam Ghavami and Hossein Pedram, "High performance asynchronous design flow using a novel static performance analysis method," Comput. Electr. Eng., vol. 35, no. 6, pp. 920–941, Nov. 2009.