International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

FPGA Prototyping of 8-Bit Trellis Encoder & IC Design using Cadence

Jayesh Argade[1], *Vaishnavi Avhad*[2], *Anisha Bhujbal*[3], *Mr. H. S. Thakar*[4]

[1 2 3] **Department of Electronics and Telecommunication, Pune Institute of Computer Technology**

[4] **Assistant Professor, Department of Electronics and Telecommunication, Pune Institute of Computer Technology**

ABSTRACT-

In this project, we propose an 8-bit trellis encoder that can encode 8-bit data words into 16-bit symbols. The encoder uses a convolutional code with a constraint length of 8 and a generator polynomial of binary form. The encoder is implemented in Verilog and synthesised for a Xilinx 14.7 ISE FPGA device.

The proposed design is then prototyped on a Xilinx FPGA spartan 3E development board. The design is tested for different input data patterns and the output symbols are observed on FPGA board results show that the designed encoder can effectively encode 8-bit data words into 16-bit symbols with a low error rate. The FPGA implementation also demonstrates that the proposed design can be easily integrated into a larger communication system.

Overall, the proposed 8-bit trellis encoder design provides a reliable and efficient method for encoding data in digital communication systems. The FPGA implementation of the design demonstrates its feasibility and applicability in real-world communication systems.

Keywords— Convolution 8 bit Trellis Diagram, VERILOG, FPGA, Cadence, Xilinx 14.7 ISE

INTRODUCTION

FPGA prototyping and IC design are two important aspects of digital circuit design. FPGA stands for Field Programmable Gate Array, which is a type of integrated circuit that can be programmed to perform a specific function. IC design, on the other hand, involves the creation of an integrated circuit from scratch using a specialised software tool called a CAD tool.

The 8-bit TRELLIS encoder is a specific type of encoder that is commonly used in digital communication systems. It uses a specific encoding algorithm to convert an 8-bit input data into a 16-bit output data. This encoding algorithm is based on the TRELLIS coding technique, which is a type of error-correcting code that is commonly used in digital communication systems.FPGA prototyping involves designing and implementing a digital circuit using an FPGA device. This process involves creating a hardware description of the digital circuit using verilog. Once the design is complete, it is synthesised into a bitstream that can be loaded onto the FPGA device. The FPGA device can then be programmed to perform the desired function

IC design using CADENCE involves designing an integrated circuit from scratch using a specialised software tool called a CAD tool. This process involves creating a schematic diagram of the digital circuit and then designing the physical layout of the circuit components on a semiconductor substrate.

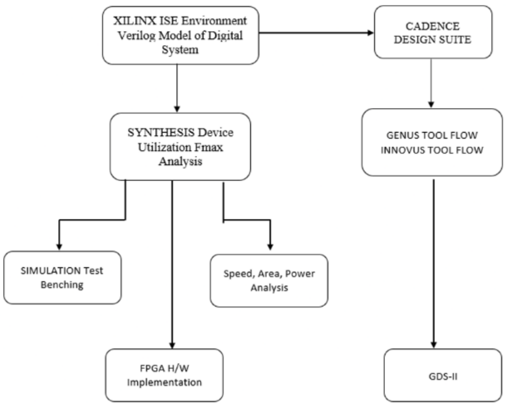
XILINX ISE Environment Verilog Model of Digital System
CADENCE DESIGN SUITE
SYNTHESIS Device Utilization Fmax Analysis
GENUS TOOL FLOW INNOVUS TOOL FLOW
SIMULATION Test Benching
Speed, Area, Power Analysis
FPGA H/W Implementation
GDS-II

Fig a: Block Diagram of 8-Bit Trellis Encoder & IC design Using Cadence

FPGA SIMULATION OF TRELLIS ENCODER

Design Considerations for a Trellis Encoder:

1. Code Rate: The code rate is the ratio of the output bits to the input bits. The code rate of a trellis encoder affects the error-correcting capabilities of the code, as well as its complexity.
2. Constraint Length: The constraint length determines the number of previous input bits that the encoder uses to determine the current output bit. A longer constraint length typically leads to better error-correcting capabilities, but also increases the complexity of the encoder.
3. Trellis Structure: The trellis structure determines the transitions between the encoder states, and can have a significant impact on the performance and complexity of the encoder.
4. Polynomials: The polynomials used in the encoder determine the feedback connections between the shift registers that store the previous input bits. Careful selection of polynomials can improve the error-correcting capabilities of the code.

Implementation of an 8-bit Trellis Encoder using Verilog:

1. Define the inputs and outputs of the encoder, including the data input and clock signal.
2. Implement the shift registers used to store the previous input bits. Implement the trellis structure and state transitions. Define the polynomials used for the feedback connections.
3. Combine the shift registers and trellis structure to generate the encoded output.
4. Simulate the encoder design to verify its functionality.

SIMULATION AND VERIFICATION OF TRELLIS ENCODER

1. Create a testbench to generate input data and clock signals.
2. Simulate the encoder design using the testbench to verify that the encoded output matches the expected results.
3. Verify the error-correcting capabilities of the code by introducing errors into the input data and verifying that the decoder can correct these errors.
4. Use FPGA prototyping to test the encoder design on hardware and verify its functionality in real-world conditions.
5. Use Cadence to design an integrated circuit (IC) for the encoder and verify its functionality through simulation and testing on hardware.

FPGA PROTOTYPING OF TRELLIS ENCODER

FPGA implementation of a trellis encoder involves the use of programmable logic devices to design and implement the trellis encoder. The trellis encoder is a type of error-correcting code that is used to improve the reliability of digital communication systems. The FPGA implementation of the trellis encoder provides a flexible and cost-effective solution for implementing this error-correcting code in various digital communication systems.

The implementation of the 8-bit trellis encoder on an FPGA board involves several steps. First, the trellis encoder algorithm needs to be designed and optimised for FPGA implementation. Next, the hardware architecture of the FPGA needs to be configured to support the trellis encoder. This involves configuring the input and output ports of the FPGA, as well as the internal logic resources such as lookup tables, flip-flops, and multiplexers.

Once the hardware architecture is configured, the trellis encoder algorithm needs to be translated into a hardware description language such as VHDL or Verilog. This involves defining the inputs, outputs, and internal logic resources of the FPGA, as well as specifying the behaviour of the trellis encoder algorithm in terms of hardware operations.

After the hardware description is complete, it needs to be synthesised and implemented on the FPGA board. This involves compiling the hardware description into a bitstream that can be loaded onto the FPGA. The bitstream defines the configuration of the FPGA and the behaviour of the trellis encoder algorithm in terms of hardware operations.

The evaluation of the FPGA implementation of the trellis encoder involves testing the performance and reliability of the encoder under various conditions. This includes testing the encoder with different input data patterns, different noise levels, and different system configurations. The performance of the FPGA implementation can be compared with other implementations of the trellis encoder, such as software implementations and ASIC implementations. The comparison can be based on factors such as performance, cost, power consumption, and flexibility.

IC DESIGN OF TRELLIS ENCODER USING CADENCE

IC (integrated circuit) design involves designing and implementing electronic circuits on a semiconductor chip. Cadence is a software suite used for IC design and verification. The IC design flow using Cadence typically involves the following steps:

1. Design Entry: The design is created using a schematic editor or a hardware description language such as Verilog or VHDL.
2. Logic Synthesis: The design is transformed into a gate-level netlist using logic synthesis tools. Physical Design: The physical layout of the design is created using a layout editor.
3. Place and Route: The placement and routing of the design is done using automated tools.
4. Verification: The design is verified using various verification tools such as static timing analysis fuctio-nal simulation
5. Design for Manufacturing (DFM): The design is analysed for manufacturability using DFM tools.

IC design using cadence has two parts:

Genus

Genus is a Cadence tool used for RTL synthesis, converting a high-level description of a digital design into gate-level representation. It optimizes the design for area, power, and performance, supporting advanced synthesis techniques and hierarchical design.The flow is follow:

- The RTL synthesis tool from Cadence.
- RTL description is transformed into a gate-level representation.
- Focuses on timing, power, and area.
- Creates a gate-level netlist that is optimised.
- Includes capability for design for test (DFT).

Innovus

Innovus, also a Cadence tool, is used for physical implementation, specifically place and route. It maps the synthesized design onto the target technology library and generates the physical layout. Innovus helps achieve timing closure, power optimization, and design manufacturability, featuring support for advanced process nodes, multi-mode and multi-corner optimization, clock tree synthesis, and detailed routing.

- The place-and-route (P&R) tool from Cadence.
- Does actual implementation work.
- Decides where the best physical places are for gates and macros.
- When placing and routing, takes timing, power, and congestion into account.
- Makes links and a physical design layout.
- Synthesises a clock tree to achieve a better clock distribution. •makes sure the design complies with physical specifications and gets it ready for manufacture.

REQUIRED TOOLS

1. Xilinx ISE Design Suite 14.7i
2. Cadence (Genus & Innovus)

RESULT AND DISCUSSION

Xilinx ISE Design Suite 14.7i

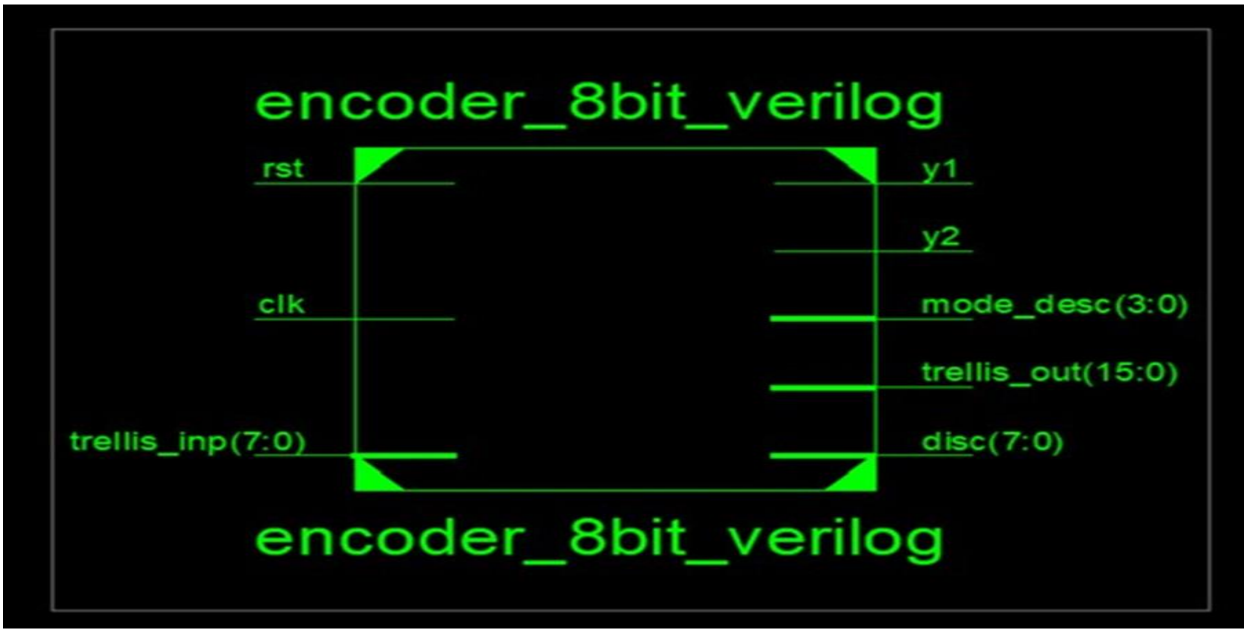

Fig b:RTL Schematic of 8-bit Trellis Encoder

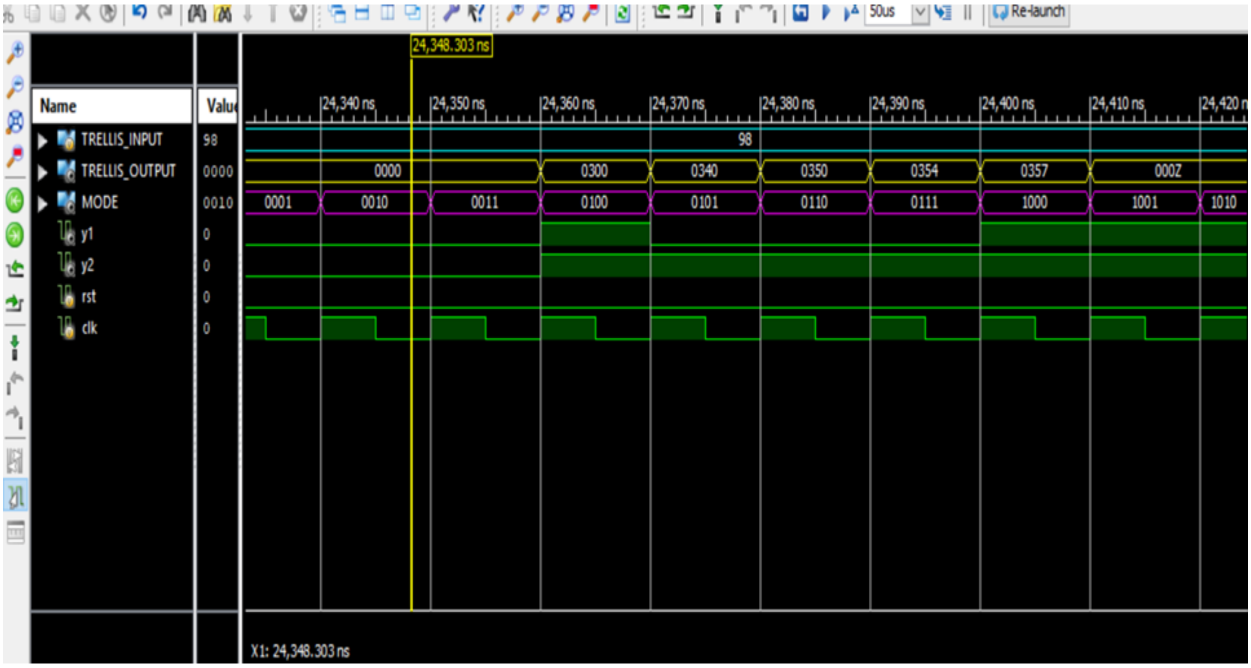

Fig c: Simulation Waveforms of 8-bitTrellis Encoder

Cadence Design Suite

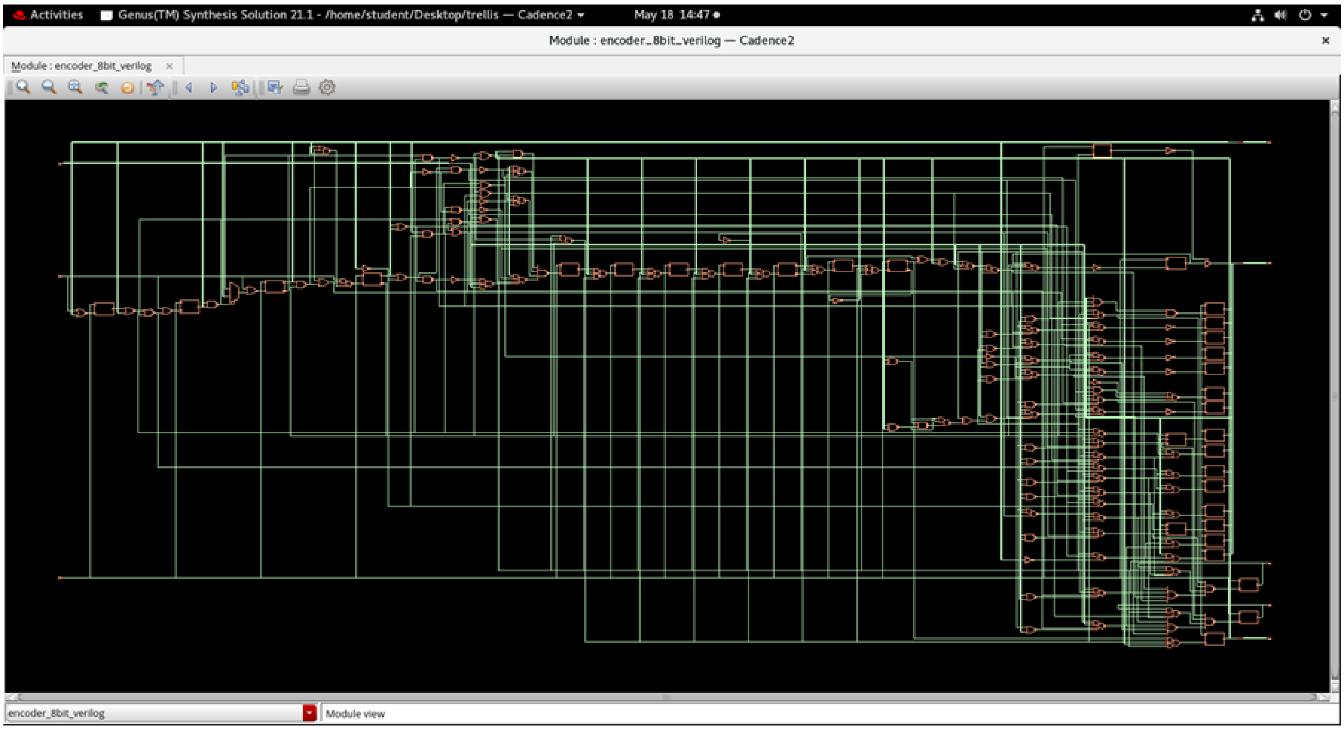
Activities
Genus(TM) Synthesis Solution 21.1 - /home/student/Desktop/trellis — Cadence2
May 18 14:47
Module : encoder_8bit_verilog — Cadence2
Module : encoder_8bit_verilog
encoder_8bit_verilog
Module view

Fig d: Simulation Waveforms of 8-bitTrellis Encoder

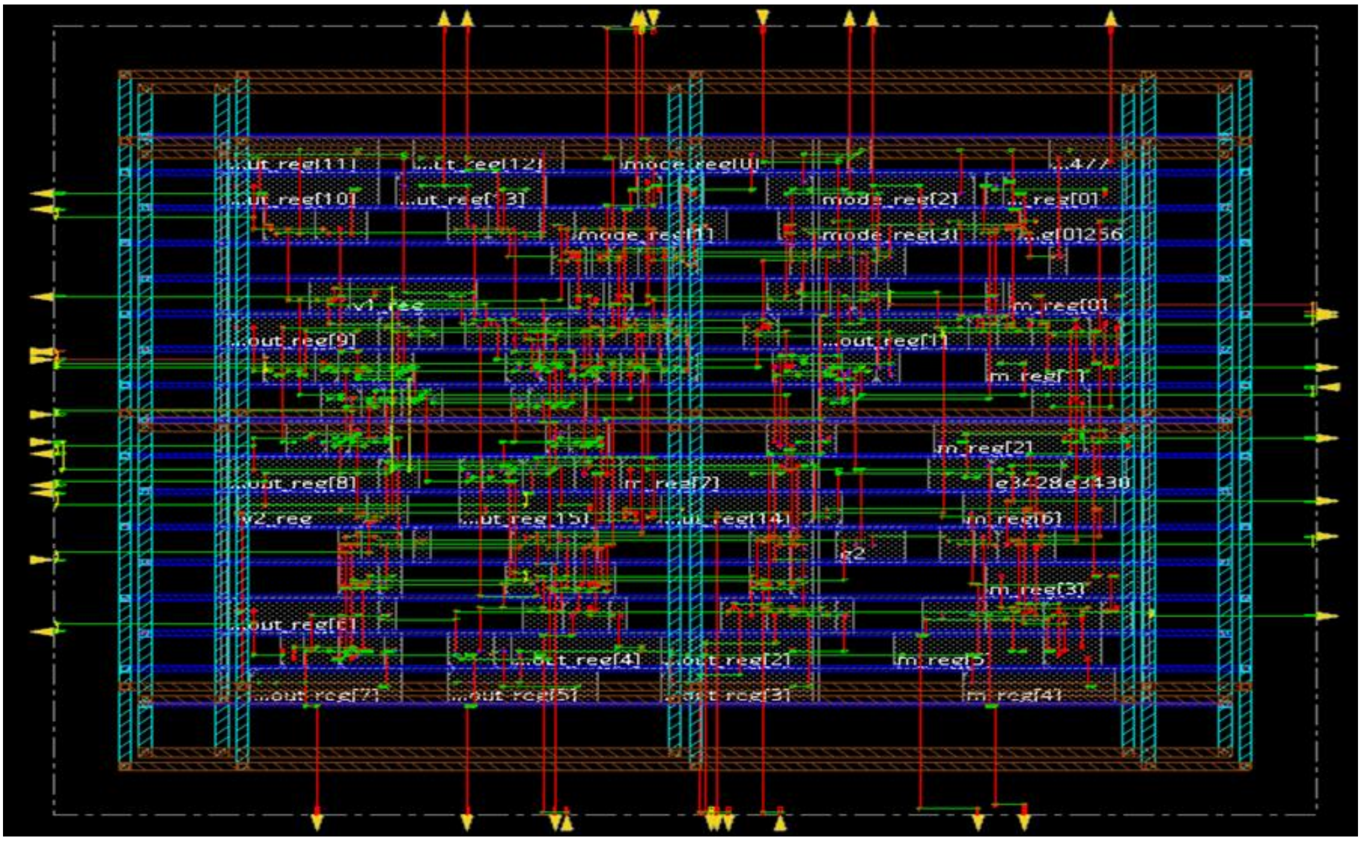

Fig e: Cadence Placement and Routing using Innovus

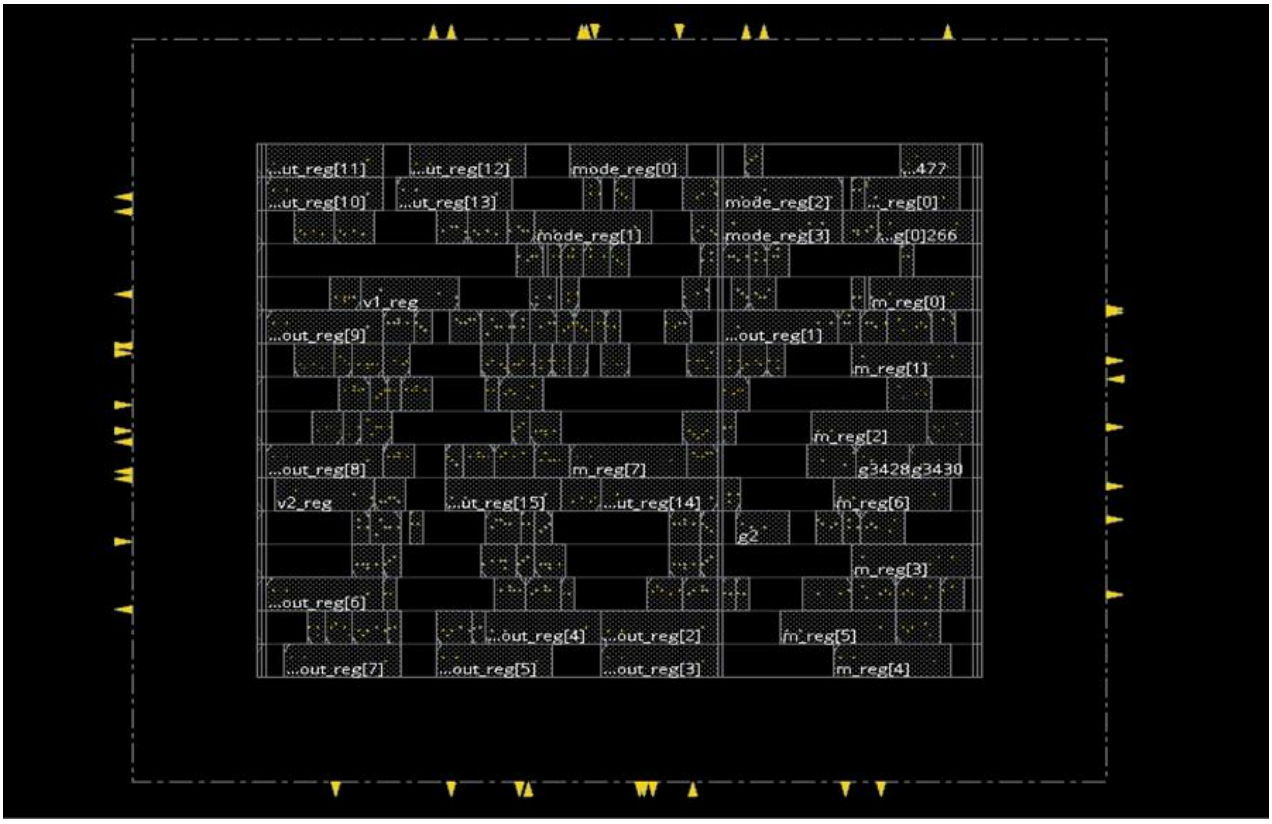

Fig f:Cadence Cell Layout using Innovus

Advantages and limitations of FPGA prototyping and IC design using Cadence:

FPGA prototyping is a quick and easy way to evaluate the performance of a design, especially for designs that require a high degree of flexibility. FPGA prototyping can help identify potential issues with the design and facilitate optimization. However, FPGA implementations may have limitations in terms of performance, especially for designs that require high speed and low power consumption. IC design using Cadence provides a high-performance and low-power solution, but it requires more resources and time to design and manufacture.

IC design can help optimise the performance of a design, especially in terms of speed and power consumption. Cadence provides a comprehensive set of tools for IC design and verification, which can help improve the design's performance and manufacturability.

REFERENCES

1. K. Rajender, K. Bapayya “FPGA Implementation of Efficient Viterbi Decoder for Multi-Carrier Systems” International journal of modern Engineering Research (IJMER), Vol. 4, Iss.9, Sept. 2014, 52
2. D. Chakraborty; P. Raha; A. Bhattacharya; R. Dutta “Speed optimization of a FPGA based modified viterbi decoder” 2013 International Conference on Computer Communication and Informatics
3. Seok-Jun Lee, Savoy, IL (US), Manish Goel, Dallas, TX (US) “Branch Metric unit duplication for high-speed Decoder FPGA Implementation” US 2004/0120427 A1
4. Sanket Kadu, Rahul Bawankule, and Rashmi Puri, “Design and Implementation of Viterbi Encoder and Decoder on FPGA” Department of Electronics & Communication Engineering
5. D. Ciletti Micahel “Advanced digital design with the Verilog HDL ”, Chikago State University(CSU), 1 march 2017