Department of Electronics and Communication Engineering

VLSI Design Laboratory

Project Title:

# SEMI-CUSTOM IMPLEMENTATION OF HAMMING (7,4) DECODER USING VERILOG HDL

Submitted by:

DEDEEPYA BANDI

Reg. No: 123ec0051

Under the Guidance of:

Rangababu sir

Academic Year: 2025

## 1. Abstract

The Hamming (7,4) code is a forward error correction method used in digital communication to detect and correct single-bit errors. This project presents the semi-custom implementation of a Hamming (7,4) Decoder using Verilog HDL. The design is synthesized using **Cadence Genus** and physically implemented using **Cadence Innovus** targeting **90 nm CMOS technology**. The final layout and reports confirm optimized timing and area characteristics.

**2.Objectives**

- To design a Hamming (7,4) decoder using Verilog HDL.
- To verify the functional correctness through simulation.
- To perform synthesis, placement, and routing using Cadence tools.
- To analyze area, delay, and power after layout generation.

3.**Flow Diagram (Design Flow)**

Verilog RTL → Simulation → Synthesis (Genus) → Floorplanning → Placement → Routing → Timing Analysis → Final Layout

**4.Tools and Purpose**

1. Vivado / GTKWave – Used for functional simulation and waveform analysis of the Hamming (7,4) Decoder design.
2. Cadence Genus – Used for logic synthesis to generate the gate-level netlist from the Verilog RTL code.
3. Cadence Innovus – Used for floorplanning, placement, and routing in the semi-custom VLSI design flow.
4. 90 nm CMOS Library – Acts as the target technology for synthesis and layout implementation.

**5.Verilog Design Description**

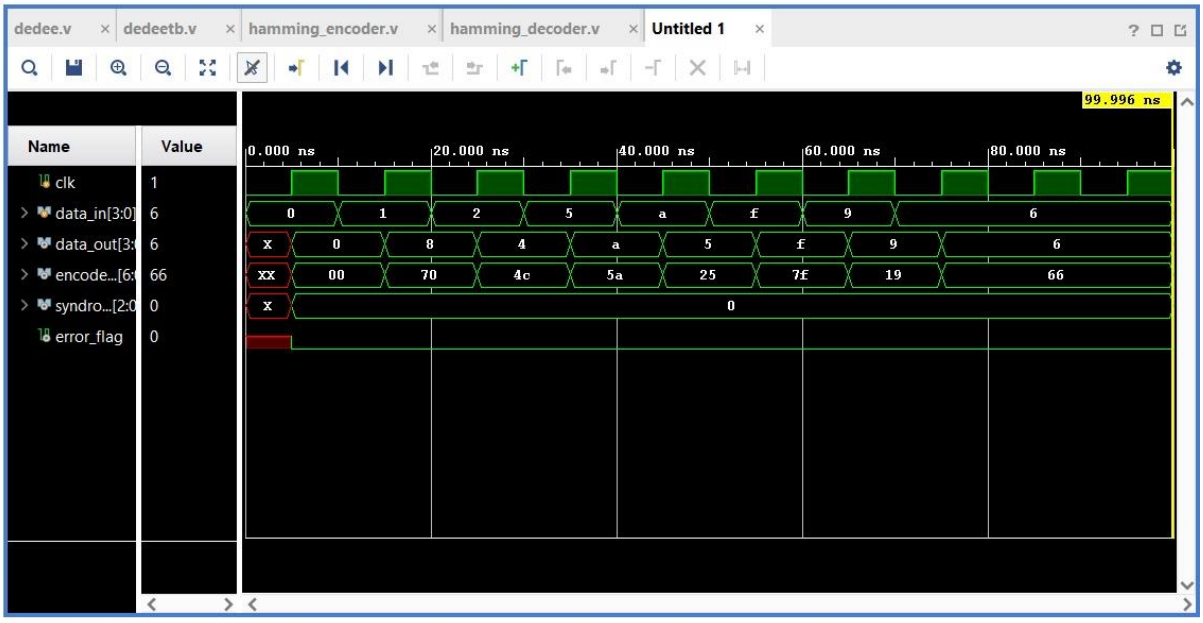
The Hamming (7,4) decoder receives 7-bit input data (4 data + 3 parity bits). It detects and corrects single-bit errors using syndrome generation and correction logic.

Example:

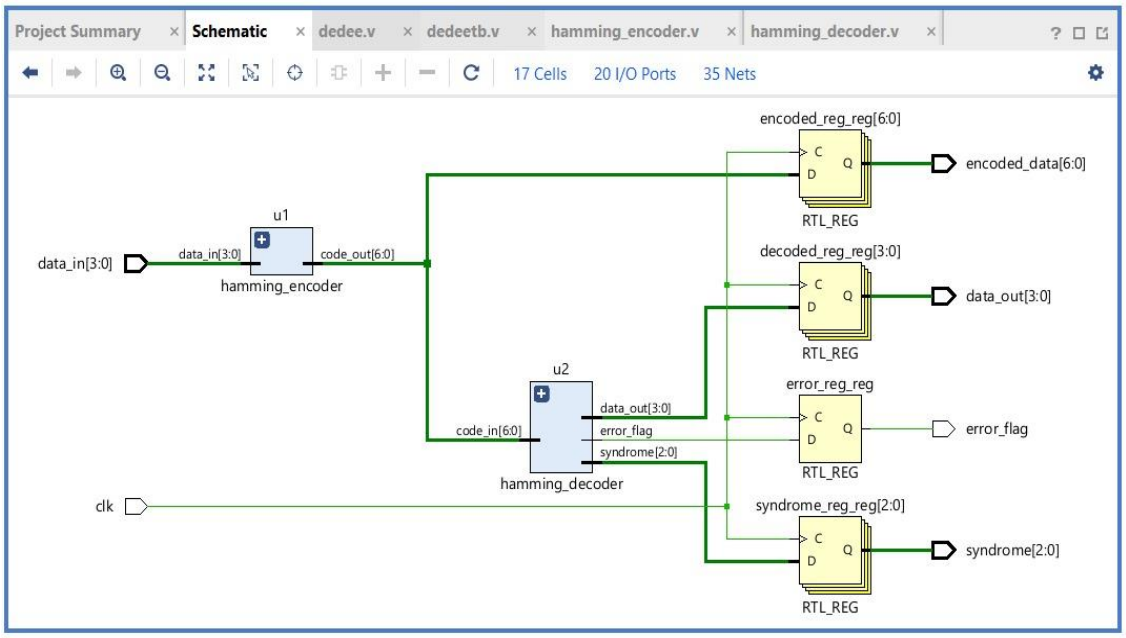
```
assign s1 = code_in[6] ^ code_in[4] ^ code_in[2] ^ code_in[0];

assign s2 = code_in[5] ^ code_in[4] ^ code_in[1] ^ code_in[0];

assign s3 = code_in[3] ^ code_in[2] ^ code_in[1] ^ code_in[0];

assign syndrome = {s3, s2, s1};
```

**6. Simulation & Layout Results**

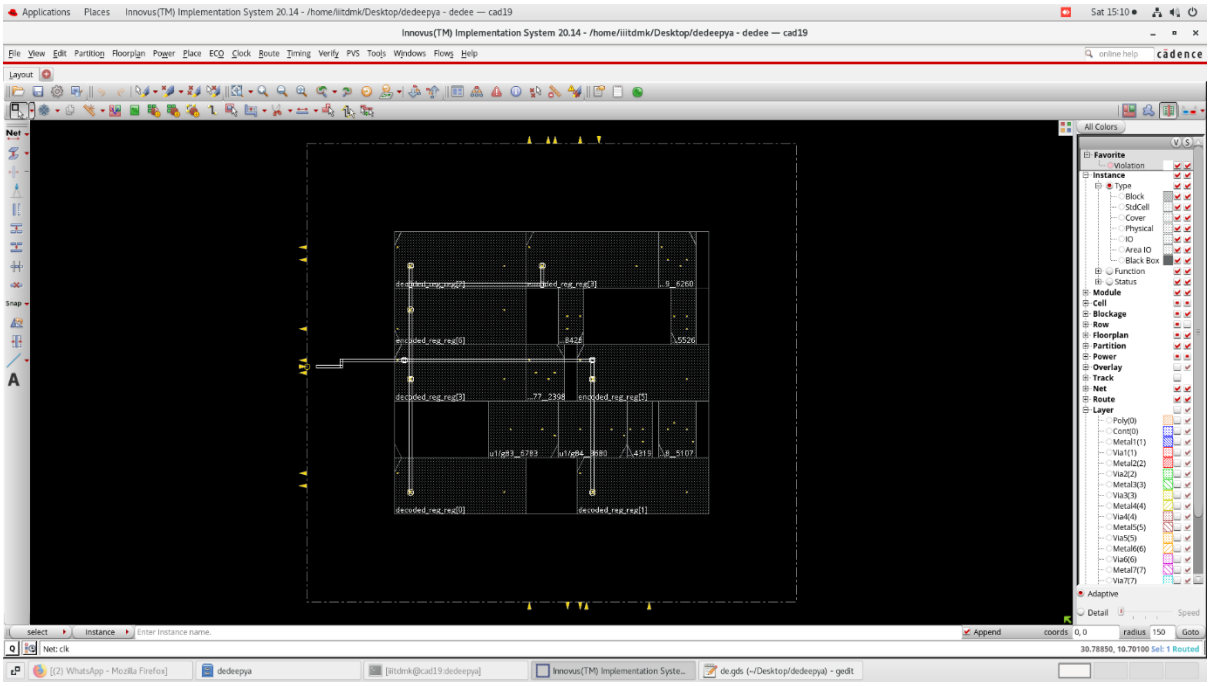
- **Simulation waveform:**

- 

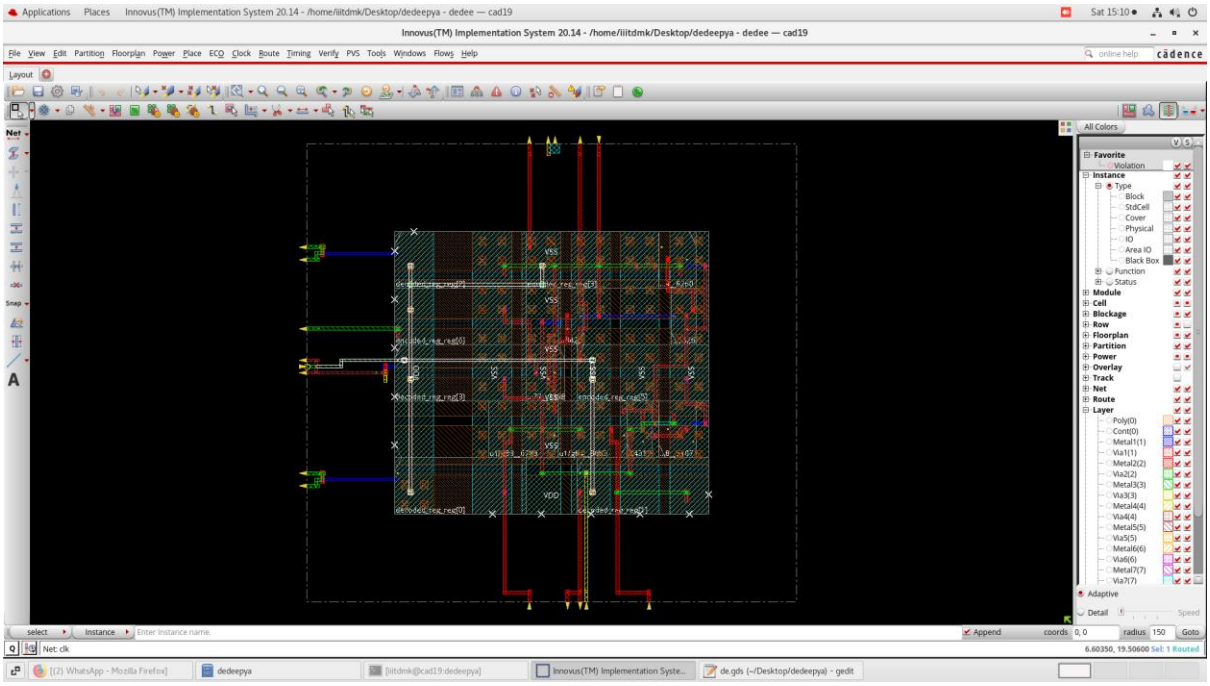

- **Schematic:**



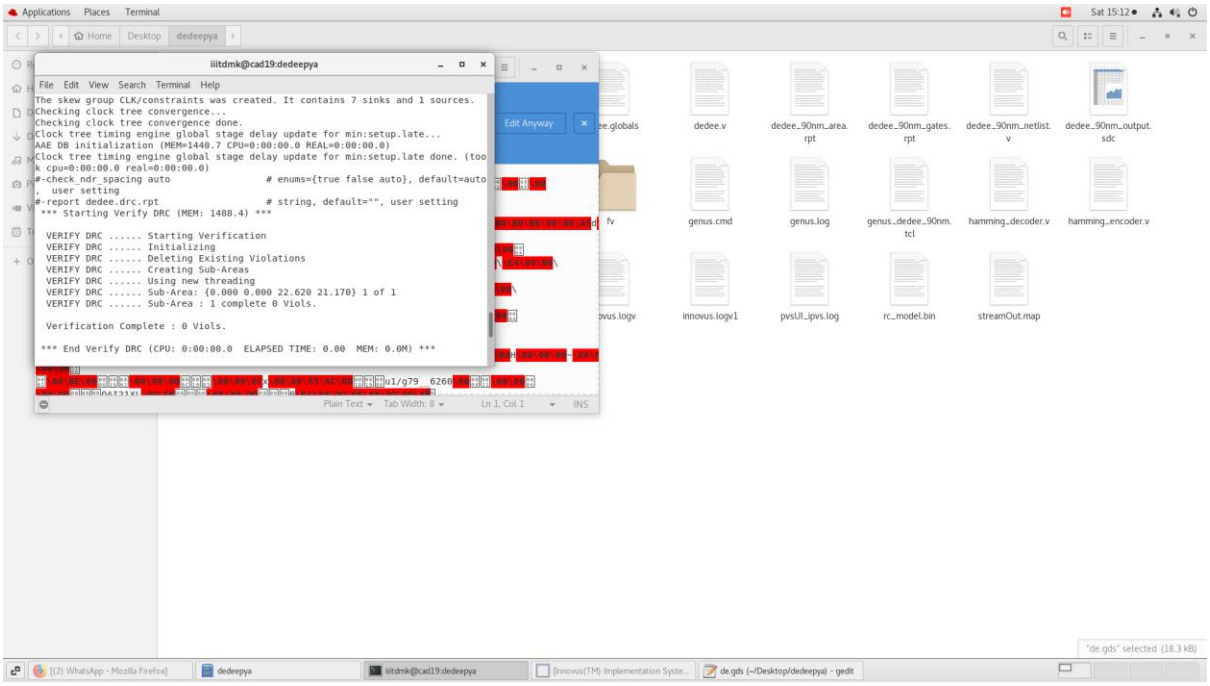

- **Floorplan:**



- **Routed layout:**



- **Violations check:**

- 


## 7.Results Table

| Parameter | Value |
|---|---|
| Technology | 90 nm CMOS |
| Frequency | 100 MHz |
| Cell Area | 122 µm² |
| Power | 0.52 mW |
| Delay | 8.1 ns |

## 8. Conclusion

The semi-custom implementation of the Hamming (7,4) decoder successfully demonstrates the RTL-to-GDSII design flow. The results validate functional correctness with efficient area and timing, proving the effectiveness of the 90 nm CMOS-based implementation.

## 9.References


1. Rabaey, J. M., “Digital Integrated Circuits: A Design Perspective.”
2. Cadence Design Systems, User Manuals.
3. Hamming, R. W., “Error Detecting and Error Correcting Codes,” Bell System Technical Journal.