Report: Hamming Code Error Detection and Correction in Cadence

1. Introduction

In digital communication and data storage systems, errors caused by noise, interference, or faults can lead to incorrect data being received or stored. **Hamming codes**, introduced by Richard W. Hamming in 1950, are a simple and effective method to detect and correct single-bit errors in transmitted data. These codes are widely used in applications where lightweight error correction is sufficient.

The **Hamming (7,4) code** encodes 4 data bits into a 7-bit codeword by adding 3 parity bits. These parity bits are calculated using combinations of the data bits, enabling detection and correction of a single-bit error. If an error occurs, a process called **syndrome decoding** analyzes the received codeword to identify the erroneous bit, which is then corrected.

This project implements a **Hamming (7,4) Error Detection and Correction System** using **Cadence Virtuoso**, a leading platform for circuit design and simulation. The project involves:

1. Encoding a 4-bit message into a 7-bit Hamming code.
2. Simulating errors in the received codeword.
3. Detecting errors using a syndrome generator.
4. Correcting the erroneous bit to recover the original data.

2. Objectives

The main goals of the project are:

1. To design a Hamming (7,4) encoder for generating 7-bit codewords from 4-bit messages.
2. To implement a syndrome generator to detect single-bit errors.
3. To design logic for error correction and data recovery.
4. To validate the design through simulation using Cadence.

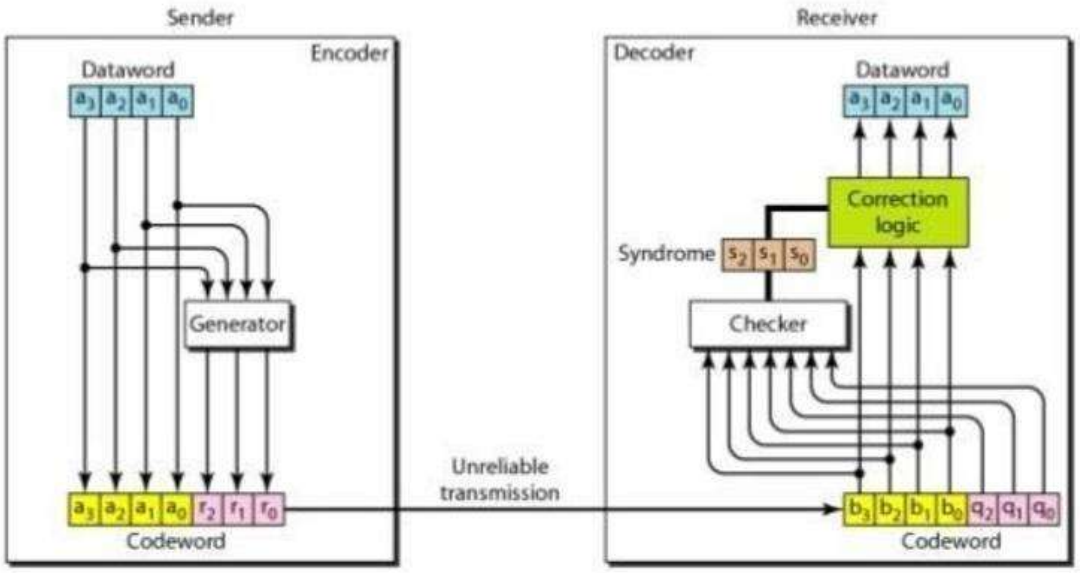
3. System Overview

This section describes the overall architecture and flow of the Hamming code system.

Subsections:

1. **Components**:
    - Encoder: Generates the Hamming code with parity bits.
    - Syndrome Generator: Detects errors by recomputing parity.
    - Error Locator and Correction: Identifies and corrects erroneous bits.
    - Data Extraction: Recovers the original data.

2. **Block Diagram**:

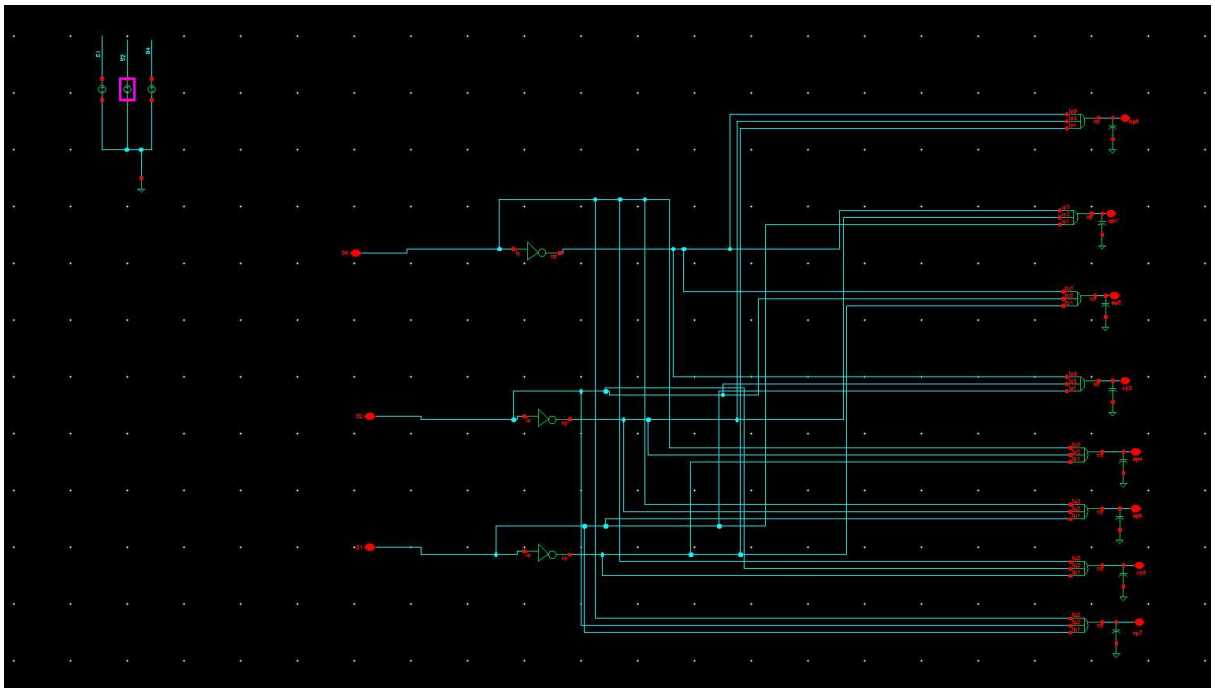


3.1 Components Used

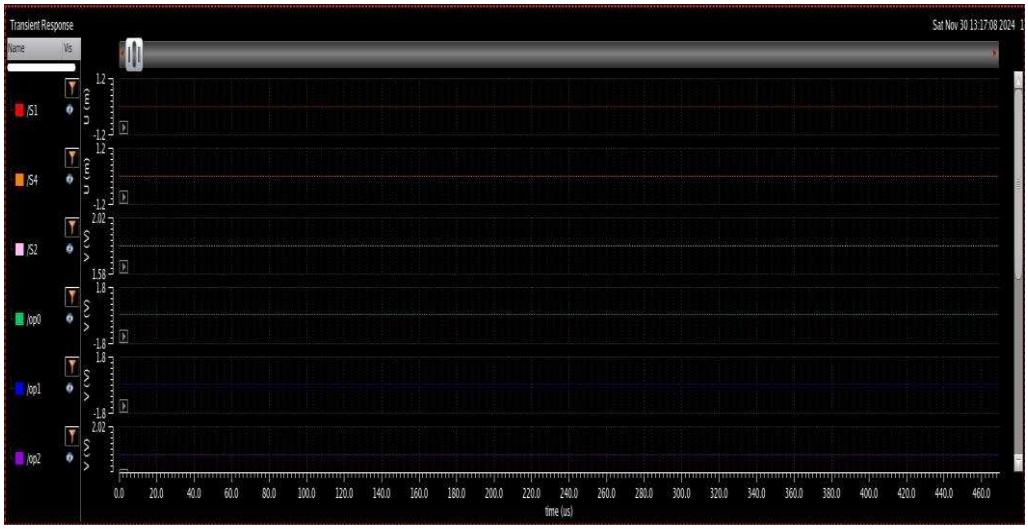
The following gates and components were designed and simulated:

- **3x8 Decoder**:
    - Converts a 3-bit syndrome vector (S1,S2,S4S_1, S_2, S_4S1,S2,S4) into a one-hot output, identifying the error location.
- 

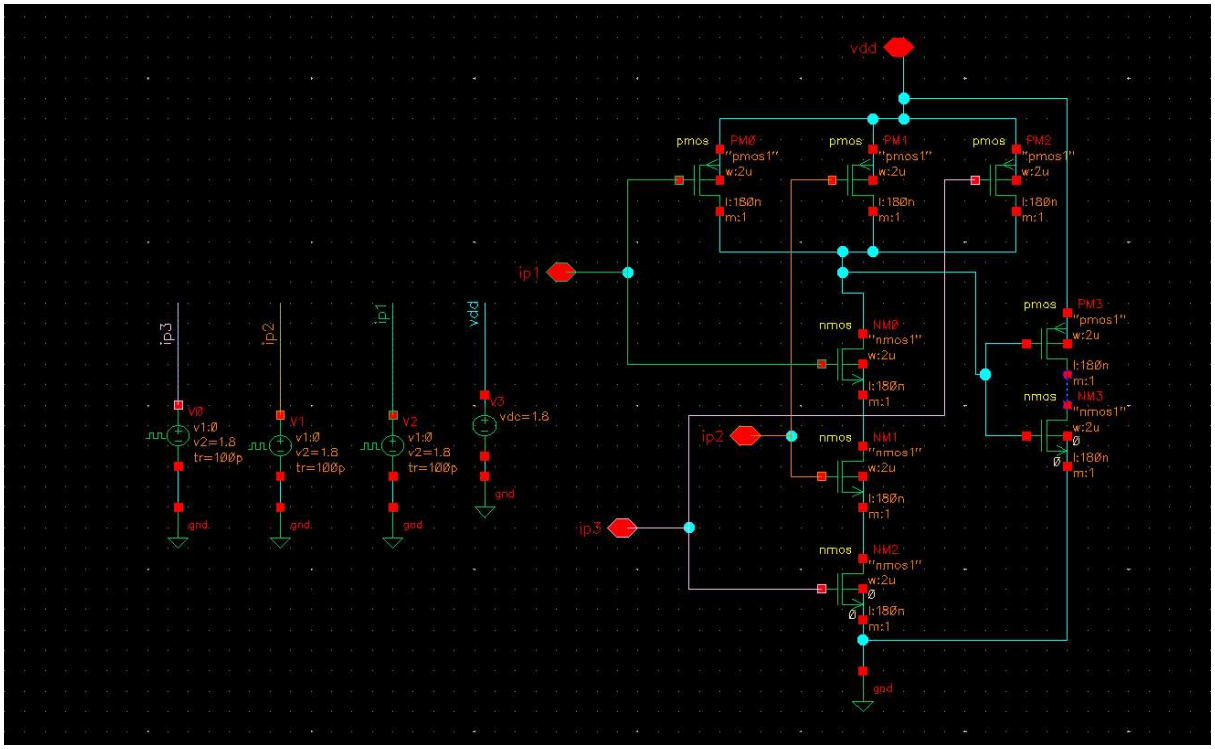
Internal circuits consists of and gates

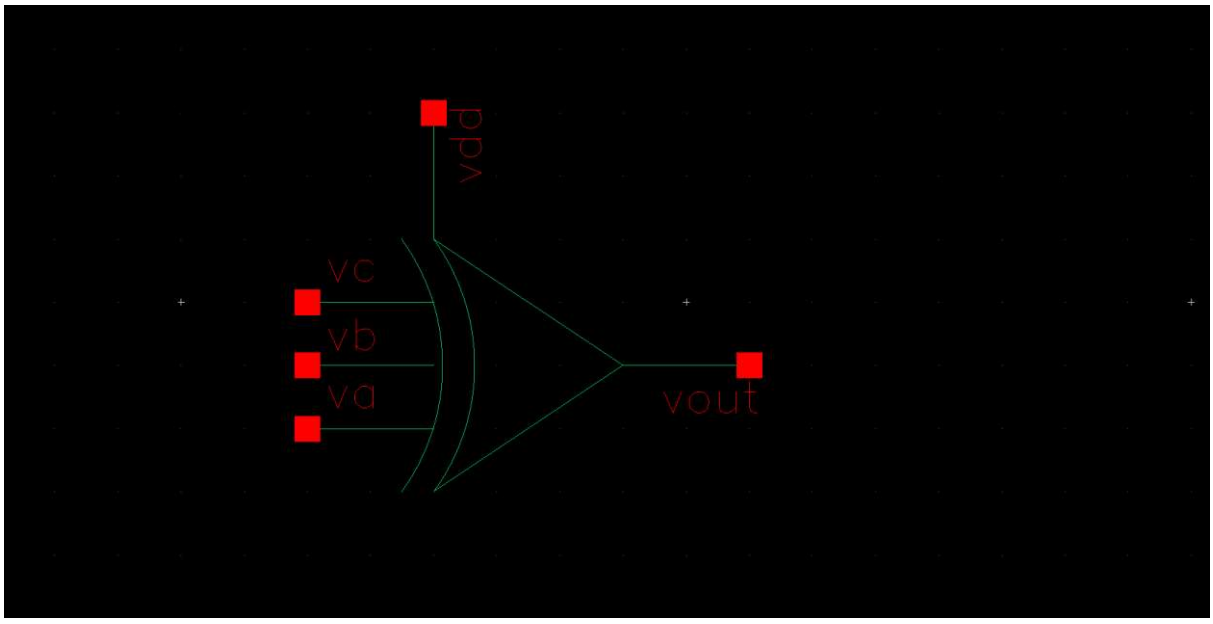
- 

Waveform of its test bench

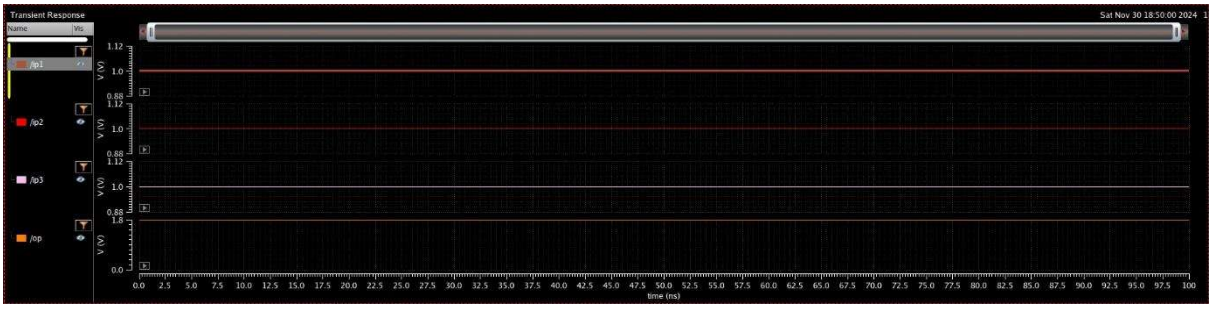


- **3-Input AND Gate**:
    - Combines signals for error detection logic. In decoder



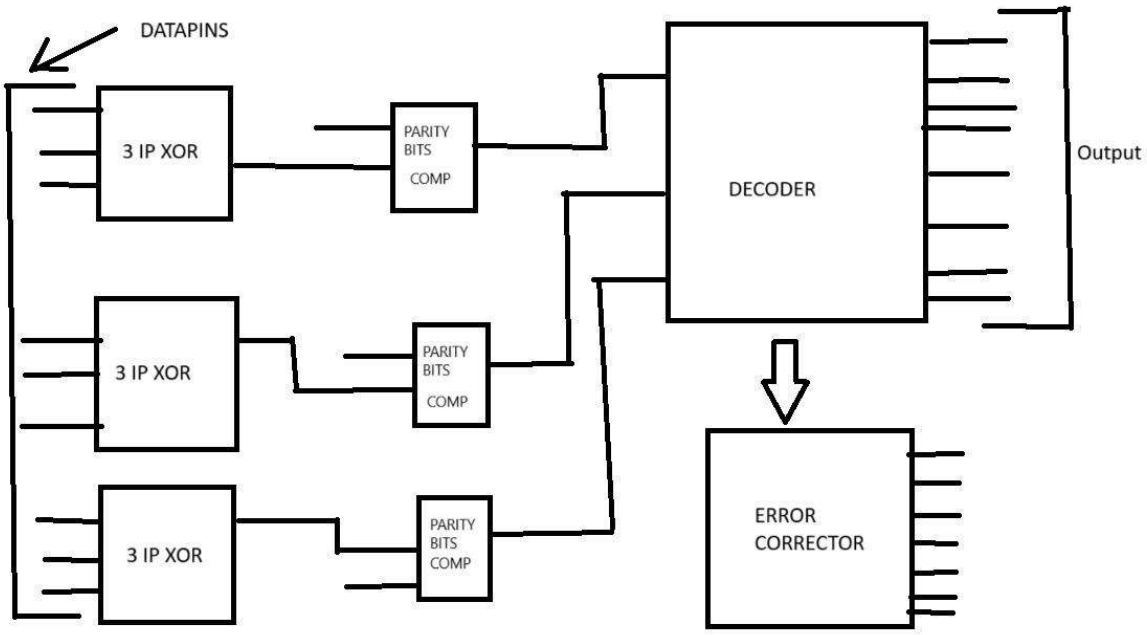
- **3-Input XOR Gate**:
  - Used in syndrome generation and parity calculations.
  - 

Waveform



- **2-Input XOR Gate**:
    - Performs bit-flipping in the error correction process.
    - Where its output goes to decoder with not gate included

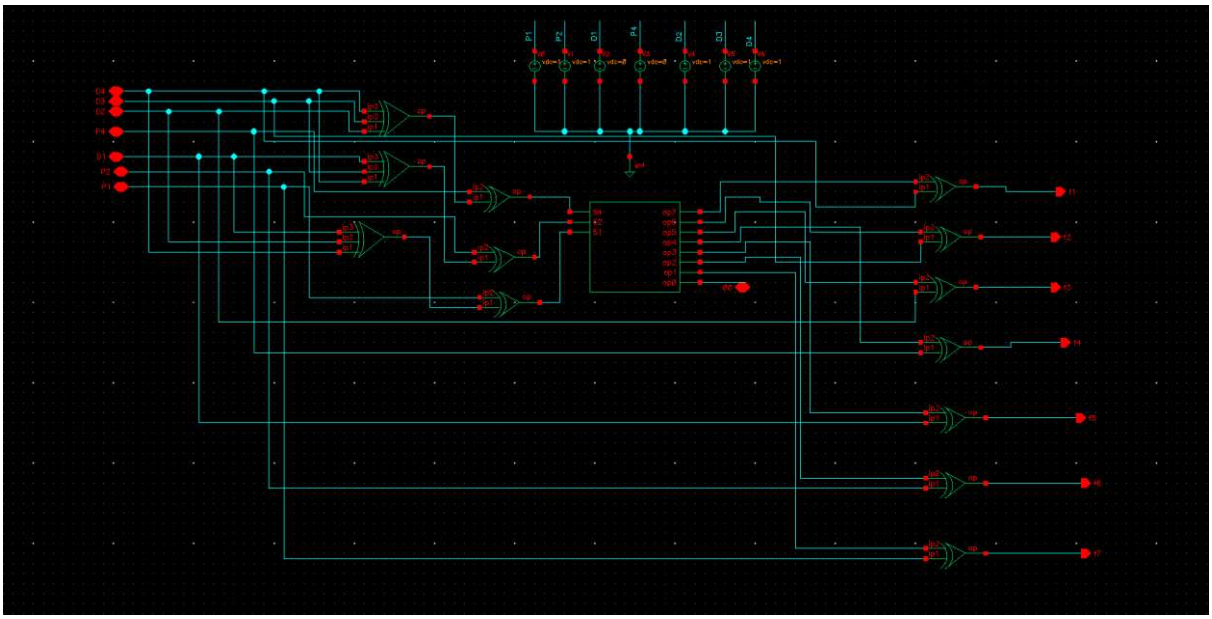
4.Real flow diagram and its interconnection of block



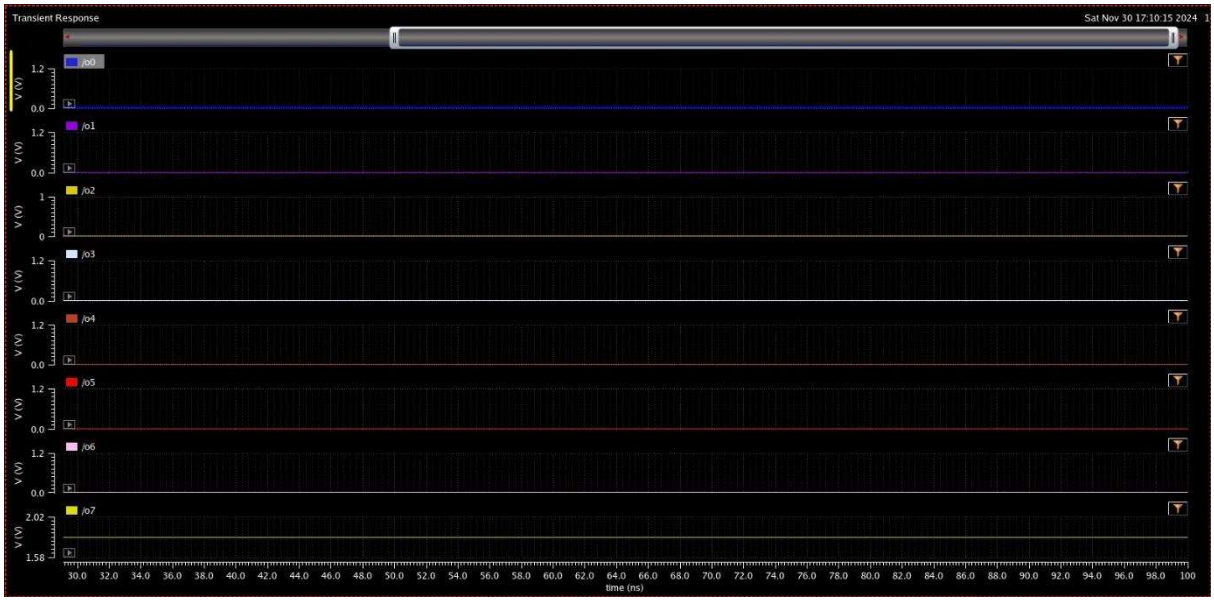
Process Overview

1. **Input Setup**:
    - The 7-bit data stream consists of:
        - **4 Message Bits** (D1,D2,D3,D4).
        - **3 Parity Bits** (P1,P2,P4).
2. **Parity Generation**:
    - Parity bits are calculated from the message bits using **3-input XOR gates** as follows:
        - $P1=D1 \oplus D2 \oplus D4$
        - $P2=D1 \oplus D3 \oplus D4$
        - $P4=D2 \oplus D3 \oplus D4$
    - These parity bits ensure error detection and correction during data transmission.
3. **Parity Comparison**:
    - During transmission, the received data includes 4 message bits and 3 parity bits.
    - Using **2-input XOR gates**, the system compares the received parity bits with recalculated parity bits to generate the **syndrome vector** (S1,S2,S4S_1, S_2, S_4S1,S2,S4).
4. **Error Detection with Decoder**:
    - The **3x8 Decoder** takes the syndrome vector as input and outputs a **one-hot signal** to indicate the error location in the 7-bit stream.
5. **Error Correction**:
    - Using XOR gates, the system flips the erroneous bit based on the one-hot signal from the decoder.
    - The corrected 7-bit stream is then used to extract the original 4-bit message.
6. **Output Recovery**:
    - The corrected 7-bit stream is split into 4 message bits p1p2d1p3d2d3d4

Final hamming code block that corrects data and detect too



Output waveform of detector is



Here d4 I have change and in these 07 gets 1 while all others are 0 indicating d4 as eroor last 7 bit

Corrector waveform

Defect in hamming code

Only defect is that it able to correct and trace an single bit

Conclusion

The implementation of the Hamming (7,4) Error Detection and Correction System successfully demonstrates the ability to identify and correct single-bit errors in transmitted data. Using Cadence Virtuoso, the design was validated through simulation, ensuring robust performance of the encoding, error detection, and correction mechanisms.

Key takeaways include:

1. The use of XOR gates for parity generation and comparison effectively ensures error detection.
2. The syndrome vector and 3x8 decoder accurately pinpoint the error location within the 7-bit stream.
3. The error correction circuit reliably flips the erroneous bit, recovering the original 4-bit message.

This project showcases the practical application of error-correcting codes in digital communication, ensuring data integrity and reliability. The modular design allows for easy

integration into larger systems and highlights the importance of error correction in modern electronics.