

Computer Engineering

Electronic and Communication Systems

Error correction code

Project Report

Federica Perrone

Summary

[Introduction 3](#_Toc90114071)

[Overview of Hamming Code 3](#_Toc90114072)

[General Algorithm 3](#_Toc90114073)

[Applications 4](#_Toc90114074)

[Possible Architecture 4](#_Toc90114075)

[Architecture description 5](#_Toc90114076)

[VHDL Code 6](#_Toc90114077)

[Test-plan 7](#_Toc90114078)

[Synthesis 8](#_Toc90114079)

[Conclusion 9](#_Toc90114080)

# Introduction

An error correcting code is an algorithm for expressing a sequence of numbers such that any errors, which are introduced, can be detected and corrected (within certain limitations) based on the remaining numbers.

The error correcting codes are used for controlling errors in data over unreliable or noisy communication channels.

The central idea is the sender encodes the message with redundant information in the form of an ECC. The redundancy allows the receiver to detect a limited number of errors that may occur anywhere in the message, and often to correct these errors without retransmission.

The two main categories of ECC codes are block codes and convolutional codes.

Hamming codes are a family of linear error-correcting block codes. Richard Hamming invented Hamming codes in 1950 as a way of automatically correcting errors introduced by punched card readers. The scheme invented by Hamming adds additional parity bits (k) to the information bits (n), and can self-detect and self-correct any single-event effect (SEE) error that occurs during transmission. Once the location of the error is identified and located, the code inverts the bit, returning it to its original form.

Hamming codes form the foundation of other more complex error correction schemes. The original scheme allows **single-error correction single-error detection** (**SECSED**), but with an addition of one parity bit, an extended Hamming version allows **single-error correction and double-error detection** (**SECDED**).

The extended Hamming code is popular in computer memory systems.

## Overview of Hamming Code

### General Algorithm

As mentioned previously, Hamming code uses parity bits to perform error detection and correction. Bit position (code word) is dependent on the amount of data bits protected. Parity bit positions are placed according to, 2 to the power of parity bit:

A general algorithm can be deduced from the following description:

1. Number the bits starting from 1: bit 1, 2, 3, 4, 5, 6, 7, etc.
2. Write the bit numbers in binary: 1, 10, 11, 100, 101, 110, 111, etc.
3. All bit positions that are powers of two (have a single 1 bit in the binary form of their position) are parity bits: 1, 2, 4, 8, etc. (1, 10, 100, 1000)
4. All other bit positions, with two or more 1 bits in the binary form of their position, are data bits.
5. Each data bit is included in a unique set of 2 or more parity bits, as determined by the binary form of its bit position:

* Parity bit 1 (P1) covers all bit positions which have the least significant bit set: bit 1 (the parity bit itself P1), 3, 5, 7, 9, etc. (all the odd numbers);
* Parity bit 2 (P2) covers all bit positions which have the second least significant bit set: bits 2 (P2), 3, 6, 7, 10, 11, etc. (sets of 2);
* Parity bit 4 (P4) covers all bit positions which have the third least significant bit set: bits 4 (P4), 5, 6, 7, 12, 13, 14, 15, etc. (sets of 4);
* Parity bit 8 (P8) covers all bit positions which have the fourth least significant bit set: bits 8 (P8), 9, 10, 11, 12, 13, 14, 15, etc. (sets of 8);
* Parity bit 16 (P16) covers all bit positions which have the fifth least significant bit set: bits 16 (P16), 17, 18, 19, 20, 21, 22, 23, etc. (sets of 16);

In the following table (Table 1), a typical code word layout is shown.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bit position | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Encoded data bits | | **P1** | **P2** | **D1** | **P4** | **D2** | **D3** | **D4** | **P8** | **D5** | **D6** | **D7** | **D8** | **D9** | **D10** | **D11** | **P16** |
| Encoded data coverage | **P1** |  |  | ✓ |  | ✓ |  | ✓ |  | ✓ |  | ✓ |  | ✓ |  | ✓ |  |
| **P2** |  |  | ✓ |  |  | ✓ | ✓ |  |  | ✓ | ✓ |  |  | ✓ | ✓ |  |
| **P4** |  |  |  |  | ✓ | ✓ | ✓ |  |  |  |  |  | ✓ | ✓ | ✓ |  |
| **P8** |  |  |  |  |  |  |  |  | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |  |
| **P16** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 1 - bit layout of Hamming code

The layout makes each column have a unique parity bit combination, for each bit position. This unique parity bit combination in known as the syndrome value. The syndrome allows errors to be located and corrected.

## Applications

The Error Correcting Codes (ECC), and more specifically Hamming codes, are used in many applications, such us:

* Telecommunication industry;
* Computer memory, modems and embedded processors;
* Nano Satellites.

## Possible Architecture

# Architecture description

# VHDL Code

# Test-plan

# Synthesis

# Conclusion