Single Parity Bit - Error Detection Study Notes

Overview

Single Parity Bit is a simple and cost-effective method for error detection in data communication. It uses the **m+1** formula where:

- **m** = number of message bits
- **+1** = one additional redundant bit (parity bit)

Key Characteristics

Advantages

- **Least expensive method** - adds only 1 extra bit
- **Simple implementation**
- Suitable for applications that can tolerate minor errors (e.g., audio transmission)

Comparison with Other Methods

- Double Parity, CRC, and Checksum require more redundant bits
- Single parity sends fewer extra bits, making it cost-effective

Types of Parity

Even Parity (Most Common)

- **Rule**: Total number of 1s in the codeword should be even
- **Implementation**: Add parity bit to make total 1s even

Examples:

- Data: `1010` → Number of 1s = 2 (even) → Parity bit = `0` → Codeword: `10100`
- Data: `1110` → Number of 1s = 3 (odd) → Parity bit = `1` → Codeword: `11101`

Odd Parity

- **Rule**: Total number of 1s in the codeword should be odd
- Less commonly used than even parity

Data Word Length Convention

- Data words are typically sent in powers of 2: 4, 8, 16, 32, 64 bits
- This follows binary system conventions (base 2)

Error Detection Capabilities

Can Detect:

1. **All single-bit errors**
2. **All odd-number bit errors** (1, 3, 5, 7, ... bits)

Detection Process:

1. Receiver counts 1s in received data word
2. Calculates expected parity
3. Compares with received parity bit
4. Mismatch indicates error

Examples of Detection:

Single Bit Error:

- Sent: `11101`
- Received: `01101` (first bit changed)
- Data bits: `0110` → 1s count = 2 → Expected parity = 0
- Received parity = 1 → **Error detected**

Three Bit Error:

- Sent: `11101`
- Received: `00001` (first 3 bits changed)
- Data bits: `0000` → 1s count = 0 → Expected parity = 0
- Received parity = 1 → **Error detected**

Cannot Detect:

- **Even-number bit errors** (2, 4, 6, ... bits)

Example of Undetected Error:

- Sent: `11101`
- Received: `00101` (first 2 bits changed)
- Data bits: `0010` → 1s count = 1 → Expected parity = 1
- Received parity = 1 → **No error detected** (false negative)

Limitations

- **Detection only** - cannot correct errors

- **Cannot identify error location**
- **Cannot detect even-number bit errors**

Hamming Distance Concept

Definition

- **Hamming Distance**: Number of bit positions where two codewords differ
- Calculated by performing XOR operation between two codewords and counting 1s

Calculation Steps:

1. Take any two valid codewords
2. Perform XOR operation
3. Count number of 1s in result

Example:

- Codeword 1: `0000`
- Codeword 2: `1111`
- XOR result: `1111`
- Hamming Distance = 4

Error Detection Formula

If minimum Hamming Distance = **d**, then:

- **Can detect up to (d-1) bit errors**

For single parity bit systems:

- Minimum Hamming Distance = 2
- Can detect (2-1) = 1 bit error

Why d-1?

- If d bits change, the corrupted data might match another valid codeword
- Receiver cannot distinguish between valid transmission and error
- Only (d-1) bit changes guarantee detection

Valid Codewords Example (4-bit data with even parity)

Data	Parity	Codeword
0000	0	00000
0001	1	00011
0010	1	00101
0011	0	00110
0100	1	01001
0101	0	01010
0110	0	01100
0111	1	01111
1000	1	10001
1001	0	10010
1010	0	10100
1011	1	10111
1100	0	11000
1101	1	11011
1110	1	11101
1111	0	11110

Key Formulas & Rules

1. **Codeword length**: m + 1 (m data bits + 1 parity bit)
2. **Even parity**: Total 1s in codeword = even number
3. **Error detection**: Can detect odd-number bit errors only
4. **Hamming distance**: d → can detect (d-1) bit errors
5. **Minimum Hamming distance for single parity**: 2

Applications

- Systems where occasional errors are acceptable
- Cost-sensitive applications
- Simple communication protocols
- Audio/video transmission where minor errors don't significantly impact quality

Exam Tips

- Remember the m+1 formula
- Understand why even-bit errors cannot be detected
- Practice Hamming distance calculations

- Know the relationship between minimum Hamming distance and error detection capability