|  |  |
| --- | --- |
|  | **Bhartiya vidya Bhavan’s Sardar Patel Institute of Technology**  **Department of Electronics and Telecommunication** |

# Information Theory and Coding EXPERIMENT 2 : Linear block codes

**Name :** Vedika Goliwar

**UID :** 2022201007

**TE EXTC A**

# AIM:

Consider any Generator matrix of your choice .

Consider all possible k bit data and obtain all codewords Find hamming weight of each codeword.

Find the minimum hamming weight . This is min.hamming distance

From hamming distance find error detection capability which is (min.hamming distance-1) Find error correction capabilty =(dmin-1)/2

Print all codewords, hamming weights, hamming distance, error detection and correction capability

Matlab Code :

% Define the generator matrix

G = [1 0 0 0; 0 1 0 0; 0 0 1 0; 1 1 1 0; 0 0 0 1; 1 0 1 1];

% Calculate the number of data bits (k) and codeword length (n) [k, n] = size(G);

% Generate all possible k-bit data data

= dec2bin(0:2^k-1)-'0';

% Initialize arrays to store codewords and Hamming weights codewords = []; hamming\_weights = [];

% Generate all codewords and calculate Hamming weights for i = 1:size(data, 1) codeword = mod(data(i, :) \* G, 2);

codewords = [codewords; codeword]; hamming\_weight

= sum(codeword);

hamming\_weights = [hamming\_weights; hamming\_weight]; end

% Find the minimum Hamming distance min\_hamming\_distance

= min(hamming\_weights);

% Calculate error detection capability error\_detection\_capability

= min\_hamming\_distance - 1; % Calculate error correction capability error\_correction\_capability = (min\_hamming\_distance - 1) / 2;

% Display results disp('Generator Matrix:') ;

disp(G) ; disp('Codewords:'); disp(codewords); disp('Hamming weights:'); disp(hamming\_weights);

disp(['Minimum Hamming distance: ', num2str(min\_hamming\_distance)]); disp(['Error detection capability: ', num2str(error\_detection\_capability)]); disp(['Error correction capability: ', num2str(error\_correction\_capability)]);

Output : Generator Matrix:

1 0 0 0

0 1 0 0

0 0 1 0

1 1 1 0

0 0 0

1 1 0 1

1 Codewords:

0 0 0 0

1 0 1 1

0 0 0 1

1 0 1 0

1 1 1 0

0 1 0 1

1 1 1 1

0 1 0 0

0 0 1 0

1 0 0 1

0 0 1 1

1 0 0 0

1 1 0 0

0 1 1 1

1 1 0 1

0 1 1 0

0 1 0 0

1 1 1 1

0 1 0 1

1 1 1 0

1 0 1 0

0 0 0 1

1 0 1 1

0 0 0 0

0 1 1 0

1 1 0 1

0 1 1 1

1 1 0 0

1 0 0 0

0 0 1 1

1 0 0 1

0 0 1 0

1 0 0 0

0 0 1 1

1 0 0 1

0 0 1 0

0 1 1 0

1 1 0 1

0 1 1 1

1 1 0 0

1 0 1 0

0 0 0 1

1 0 1 1

0 0 0 0

0 1 0 0

1 1 1 1

0 1 0 1

1 1 1 0

1 1 0 0

0 1 1 1

1 1 0 1

0 1 1 0

0 0 1 0

1 0 0 1

0 0 1 1

1 0 0 0

1 1 1 0

0 1 0 1

1 1 1 1

0 1 0 0

0 0 0 0

1 0 1 1

0 0 0 1

1 0 1 0

Hamming weights:

0

3

1

2

3

2

4

1

1

2

2

1

2

3

3

2

1

4

2

3

2

1

3

0

2

3

3

2

1

2

2

1

1

2

2

1

2

3

3

2

2

1

3

0

1

4

2

3

2

3

3

2

1

2

2

1

3

2

4

1

0

3

1 2

Minimum Hamming distance: 0 Error detection capability: -1

Error correction capability: -0.5

**Conclusion :** For a randomly chosen generator matrix, all possible k-bit data were used to generate codewords. The Hamming weight of each codeword was calculated, and the minimum Hamming weight determined the minimum Hamming distance. The error detection capability was found to be one less than the minimum Hamming distance, and the error correction capability was half of the minimum Hamming distance minus one. The experiment demonstrated the error detection and correction capabilities of the code.