UNIT 5 *Binary Codes* Overhead Slides

Overhead Slides

5.1 Codes (Appendix)
5.2 Hamming Distance
5.3 Error Detection and Correction

OS 5.1 *Codes (Appendix)*

Code 1

0 0 0 0
0 0 1 1
0 1 0 1
0 1 1 0
1 0 0 1
1 0 1 0
1 1 0 0
1 1 1 1

Code 2

0 0 0 0 0 0
0 0 1 1 1 0
0 1 0 1 0 1
0 1 1 0 1 1
1 0 0 0 1 1
1 0 1 1 0 1
1 1 0 1 1 0
1 1 1 0 0 0

Code 3

0 0 0 0
0 1 0 1
1 0 1 0
1 1 1 1

Code 4

0 0 0 0
1 1 0 0
0 0 1 1
1 1 1 1

Code 5

0 0 0 0 0 0 0
0 0 1 1 1 0 1
0 1 0 1 0 1 1
0 1 1 0 1 1 0
1 0 0 0 1 1 1
1 0 1 1 0 1 0
1 1 0 1 1 0 0
1 1 1 0 0 0 1

Code 6

0 0 0 0 0 0 0
1 1 0 1 0 0 1
0 1 0 1 0 1 0
1 0 0 0 0 1 1
1 0 0 1 1 0 0
0 1 0 0 1 0 1
1 1 0 0 1 1 0
0 0 0 1 1 1 1
1 1 1 0 0 0 0
0 0 1 1 0 0 1
1 0 1 1 0 1 0
0 1 1 0 0 1 1
0 1 1 1 1 0 0
1 0 1 0 1 0 1
0 0 1 0 1 1 0
1 1 1 1 1 1 1

Code 7

0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1
0 0 1 1 0 0 1 1
0 1 0 1 0 1 0 1
0 1 1 0 0 1 1 0
0 1 0 1 1 0 1 0
0 0 1 1 1 1 0 0
0 1 1 0 1 0 0 1
1 1 1 1 1 1 1 1
1 1 1 1 0 0 0 0
1 1 0 0 1 1 0 0
1 0 1 0 1 0 1 0
1 0 0 1 1 0 0 1
1 0 1 0 0 1 0 1
1 1 0 0 0 0 1 1
1 0 0 1 0 1 1 0

Code 8

0 0 0 0 0 0 0
0 0 1 0 1 1 1
0 1 0 0 1 1 0
0 1 1 0 0 0 1
1 0 0 0 1 0 1
1 0 1 0 0 1 0
1 1 0 0 0 1 1
1 1 1 0 1 0 0
0 0 0 1 0 1 1
0 0 1 1 1 0 0
0 1 0 1 1 0 1
0 1 1 1 0 1 0
1 0 0 1 1 1 0
1 0 1 1 0 0 1
1 1 0 1 0 0 0
1 1 1 1 1 1 1

Code 9

0 1 0 1 0 1 0
1 0 0 1 1 0 0
0 0 1 1 0 0 1
1 1 1 0 0 0 0
0 1 0 0 1 0 1
1 0 0 0 0 1 1
0 0 1 0 1 1 0

OS 5.2 *Hamming Distance*

Find the Hamming distance for the code with codewords

(1 1 0 0 0) , (0 0 1 0 1) , (1 0 1 0 1) , (1 1 1 1 1)

First find the distance between each pair of codewords.

$d((1\,1\,0\,0\,0), (0\,0\,1\,0\,1)) = 4$

$d((1\,1\,0\,0\,0), (1\,0\,1\,0\,1)) =$ ☐

$d((1\,0\,1\,0\,1), (1\,1\,1\,1\,1)) =$ ☐

$d((0\,0\,1\,0\,1), (1\,0\,1\,0\,1)) =$ ☐

$d((0\,0\,1\,0\,1), (1\,1\,1\,1\,1)) =$ ☐

$d((1\,0\,1\,0\,1), (1\,1\,1\,1\,1)) =$ ☐

Hamming distance = minimum distance between any two codewords

= ☐

OS 5.3 *Error Detection and Correction*

Complete this table.

Code	*Hamming distance*	*Errors corrected*	*Errors detected*
1	2	0	1
2	3	1	1
3	...	...	...
4	...	...	...
5	...	...	...

- If δ is ***odd***, the code can correct and detect up to $\square$ errors.

- If δ is ***even***, the code can correct up to $\square$ errors and detect up to $\square$ errors.