Hamming Code

- interaction of electrical signals may cause errors in storing and retrieving binary information
- the reliability of memory unit may be improved by employing error detecting and error correcting codes.
- parity bit is used for error detection.
- A parity bit is generated and stored along with the data word.
- The data word is accepted if the parity of the bits read out is correct.
- once error is detected, it can be corrected by error correcting mechanism.
- An error correcting code generates multiple check bits that are stored with data word in memory.
- If the check bits read are correct no error has occurred
- If the check bits donot match out then the bit which has changed value is identified and complemented to rectify the error.
- One of the most common error-correcting codes used in RAM is HAMMING CODE.

→ In Hamming code, k-parity bits are added to n-bit data word forming a new word of n+k bits.

→ The bits position are numbered in sequence from 1 to n+k.

→ The bit positions numbered as power of 2 are reserved for parity bits.

→ Remaining bits are data bits.

Example with a data word of 8 bits.

consider 8-bit data word 11000 0100

we include 4-bits as parity so we get 12-bits

P_1, P_2, P_3, P_4 → parity bits (4-bit)

Bit Position	1	2	3	4	5	6	7	8	9	10	11	12
12-bit word	P_1	P_2	1	P_3	1	0	0	P_4	0	1	0	0

P_1 = XOR of bits (3, 5, 7, 9, 11)

P_2 = XOR of bits (3, 6, 7, 10, 11)

P_3 = XOR of bits (5, 6, 7, 12)

P_4 = XOR of bits (9, 10, 11, 12).

XOR → odd function [equal to 1 for odd number of 1's).

for given data word,

$P_1 = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 0$

$P_2 = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$

$P_3 = 1 \oplus 0 \oplus 0 \oplus 0 = 1$

$P_4 = 0 \oplus 1 \oplus 0 \oplus 0 = 1$

Sub. the values,

12-bit word 0 0 1 1 1 0 0 1 0 1 0 0

→ 12 bits read from memory are checked for errors

→ parity bits are checked for same combination of bits, including parity bit.

→ 4-bit check bits are, C_1 C_2 C_4 C_8

$$C_1 = \text{XOR of bits}(1, 3, 5, 7, 9, 11)$$

$$C_2 = \text{XOR of bits}(2, 3, 6, 7, 10, 11)$$

$$C_4 = \text{XOR of bits}(4, 5, 6, 7, 12)$$

$$C_8 = \text{XOR of bits}(8, 9, 10, 11, 12)$$

The resultant check bit, $C = C_8 C_4 C_2 C_1$

if $C = 0$ → No error

$C \neq 0$ → error has occured.

→ the 4-bit binary number formed $C = C_8 C_4 C_2 C_1$ gives the position of erroneous bit.

Bit position	1	2	3	4	5	6	7	8	9	10	11	12
C=0000, no error ⇐	0	0	1	1	1	0	0	1	0	1	0	0
C=0001, Error in bit 1 ⇐	1	0	1	1	1	0	0	1	0	1	0	0
C=0101, Error in bit 5 ⇐	0	0	1	1	0	0	0	1	0	1	0	0

→ The error is corrected by complementing the bit in that position.