Hamming codes

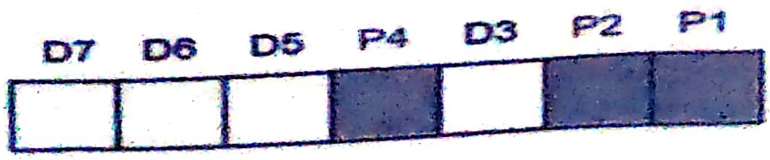
Hamming codes are linear block codes, which is an error-detection and error-correction technique, was proposed by R.W. Hamming.

Consider a message having four data bits which is to be transmitted as a 7-bit codeword by adding three error control bits. This would be called a (7,4) code.

A general method for constructing error-correcting codes by using a minimum distance of three.

Every integer m there is a ($2^m - 1$) bit hamming code which contains m parity check bits and $2^m - 1 - m$ information bits.

The 7-bits Hamming code is as:



D3 D5 D6 D7 = Data bits or Information bits.

If we number the bit positions from 1 to $2^m - 1$, the bit position 2^k, where $0 \leq k \leq m - 1$, are the parity bits, and bits in the remaining positions are information bits. Here is the message sequence to be transmitted.

Calculating the Hamming Code

The key to the Hamming Code is the use of extra parity bits to allow the identification of a single error. Code word created as follows:

1. Mark all bit positions that are powers of two as parity bits (positions 1,2,4,8,16, etc.).
2. All other bit positions are for the data to be encoded (positions 3,5,6,7,9,10,11,12, etc.).
3. Each parity bit calculates the parity for some of the bits in the code word. The position of the parity bit determines the sequence of bits that it alternately checks and skips.

Position 1: Check 1 bit, Skip 1 bit, Check 1 bit, Skip 1 bit, etc.

(1,3,5,7,9,11, 13, ...)

Position 2: Check 2 bits, Skip 2 bit, Check 2 bit, Skip 2 bit, etc.

(2,3,6,7,10,11)

Position 4 : Check 4 bit, Skip 4 bit, Check 4 bit, Skip 4 bit, etc.

(4,5,6,7,12,13,14,15,20, 21, ...)

Position 8: Check 8 bit , Skip 8 bit, Check 8 bit, Skip 8 bit, etc.

(8-15, 24-31, ...)

While checking the parity, if the total number of 1's are odd then write the value of parity bit **P1**(or **P2** etc.) as **1** (which means the error is there) and if it is even then the value of parity bit is **0** (which means no error).

Detection of error.

Example:

A seven bits Hamming code is received as 1110111. What is the correct code?

Solution: Received codeword:

D7	D6	D5	P4	D3	P2	P1
1	1	1	0	1	1	1

<u>Step 1:</u> For checking parity bit **P1**, use **check one and skip one** method, which means, starting from P1 and then skip P2, take D3 then skip P4 then take D5, and then skip D6 and take D7, this way we will have the following bits,

D7 D5 D3 P1 = 1111→number of 1 is even , so we write the value of P1 as 0. This means no error.

<u>Step 2:</u> Check for P2 but while checking for P2, we will use **check two and skip two** method, which will give us the following data bits. But remember since we are checking for P2, so we have to start our count from P2 (P1 should not be considered).

D7 D6 D3 P2 = 1111→number of 1 is even , so we write the value of P2 as 0. This means no error.

<u>Step 3:</u> Check for P4 but while checking for P4, we will use check four and skip four method, which will give us the following data bits. But remember since we are checking for P4, so we have started our count from P4(P1 & P2 should not be considered).

D7 D6 D5 P4= 1110 →number of 1 is odd , so we write the value of P4 as 1. This means error exists.

P4	P2	P1

Now, we write the error word (E) =

1	0	0

i.e.

The decimal equivalent of E is 4.

Hence, the 4th bit in the codeword is incorrect.

3.5 Flow Control