29/12/2021

HAMMING CODE

This is an error-detection & correction technique given by R.W. Hamming. Whenever a data packet is transmitted over a ntw, there are possibilities that the data bits may get lost or damaged during transmission.

There are mainly three types of errors that occur in data transmission

- single bit error (usually occurs in ‖ comm)
- Multiple bit errors (both serial & ‖)
- Burst errors (only in serial comm).

0 1 0 0 1 → 0 1 0 1 1

0 1 0 0 1 1 → 0 1 0 1 1 0

→ The change of set of bits in a data seq.

The burst error is calculated in from the first bit change to last bit change.

1 0 0 1 0 1 → 1 0 1 1 0 0

Burst Error.

difficult to solve.

<u>Error detecting codes</u> : In digital comm system errors are transferred from one comm sys to another, along with the data. If these errors are not detected and corrected, data will be lost.

one imp way is to check Parity

Parity checking : Parity bit is an additional bit added to the data at the transmitter before transmitting the data.

Even parity : if the data has even no. of 1's the parity bit is 0 otherwise 1

Odd parity : if the data has odd no. of 1's the parity bit is 0 otherwise 1.

eg: data 1010 — msg $\bar{c}$ even parity 10100

1011 — msg $\bar{c}$ even parity 10111

Error ~~correctn~~ detection $\bar{c}$ hamming code

hamming code is a block code that is capable of detecting up to two simultaneous bit errors and correcting single-bit errors.

In this coding method, the source encodes the message by inserting redundant bits within the message. These redundant bits are extra bits that are generated and inserted at specific positions in the message itself to enable error detection and correction.

→ Encoding a message by Hamming code

step 1 – calculation of no. of redundant bits

step 2 – Positioning the redundant bits

step 3 – Calculate the values of each redundant bit.

Send data + redundant bits to receiver.

step 1 : Let m be no. of data bits
r be redundant bits

$m+r$ is able to indicate atleast $m+r+1$ diff. states. Here $(m+r)$ indicates loc. of an error in each $(m+r)$ bit positions & 1 extra state is for no error.

$$2^r \geq m+r+1$$

step 2 : Positioning of redundant bits

The r redundant bits placed at bit positions of powers of 2 i.e 1, 2, 4, 8, 16 etc.

suppose we have 3 parity bits

	P_3	P_2	P_1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

$P_1 \rightarrow 1, 3, 5, 7$

$P_2 \rightarrow 2, 3, 6, 7$

$P_3 \rightarrow 4, 5, 6, 7.$

Each redundant bit r_i is calculated as parity, generally even parity based upon its bit position. It covers all bit positions whose bin rep. includes a 1 in the i^{th} pos.

→ Decoding a message in hamming codes

Once the receiver gets an incoming message, it performs recalculations to detect errors and correct them. The steps of recalculation are -

Step 1 - Calc. of no. of redundant bits

Step 2 - Position the redundant bits

Step 3 - Parity checking.

Step 4 - Error detection & Correction.

Example: Suppose we are having a message signal as $K = 1101$

and it is to be transmitted after encoding $\bar{c}$ even parity.

Step 1: Calculate no. of parity bits

$$2^P \geq m + P + 1$$

For $P = 3$ $\quad 2^3 \geq 4 + 3 + 1 \quad 8 \geq 8$ ✓

$\therefore P = 3$

Step 2: Bit position table (Total bits $= 4 + 3 = 7$)

7	6	5	4	3	2	1
D_4	D_3	D_2	P_3	D_1	P_2	P_1
1	1	0	0	1	1	0

Parity bits will be present at $2^0, 2^1, 2^2, \ldots$ so on.

step 3: Calculate the values of each parity bit

$P_1 \rightarrow$ 1, 3, 5, 7
0 1 0 1

$P_2 \rightarrow$ 2, 3, 6, 7
1 1 1 1

$P_3 \rightarrow$ 4, 5, 6, 7
0 0 1 1

Sender will send

1 1 0 0 1 1 0

Decoding at receiver

data received : 1 1 0 1 1 1 0
msg

step 1: Recalculate parity bits

$$2^p \geq m + p + 1$$

$$7 + 1 = 8 \qquad \therefore p = 3$$

step 2 Bit position table

7	6	5	4	3	2	1
D_4	D_3	D_2	P_3	D_1	P_2	P_1
1	1	0	1	1	1	0

step 3 Parity checking

P_1 : 1 3 5 7
0 1 0 1 ✓ $P_1 = 0$

P_3 : 4 5 6 7
1 0 1 1 X
$P_3 = 1$

P_2 : 2 4 6 7
1 1 1 1 ✓ $P_2 = 0$

1 0 0
Bit position 4

ex. 2

$K = 11010$

encoding

$2^P \geq K+P+1$

let $P=3$ $\quad 8 \geq 5+3+1 \quad 9$ ✗

$P=4$ ✓

Total msg bits $5+4 = 9$

Bit position table

9	8	7	6	5	4	3	2	1
D_5	P_4	D_4	D_3	D_2	P_3	D_1	P_2	P_1
1	1	1	0	1	0	0	1	1

Calculate parity bits

$P_1 \rightarrow 1\ 3\ 5\ 7\ 9$
1 0 1 1 1

$P_2 \rightarrow 2\ 3\ 6\ 7$
1 0 0 1

$P_3 \rightarrow 4, 5, 6, 7$
0 1 0 1

$P_4 \rightarrow 8\ 9$
1 1

sender sends the coded word

111010011

Received seq : 111000011 5th bit.

P_4	P_3	P_2	P_1
0	1	0	1

$= (5)_{10}$.

→ determine parity $\quad P=4$.

9	8	7	6	5	4	3	2	1
D_5	P_4	D_4	D_3	D_2	P_3	D_1	P_2	P_1
1	1	1	0	0	0	0	1	1

$P_1 \rightarrow 1\ 3\ 5\ 7\ 9$
1 0 0 1 1 $\quad P_1 = 1$

$P_2 \rightarrow 2\ 3\ 6\ 7$ ✓
1 0 0 1 $\quad P_2 = 0$

$P_3 \rightarrow 4\ 5\ 6\ 7$
0 0 0 1 $\quad P_3 = 1$

$P_4 = 8\ 9$ ✓
1 1 $\quad P_4 = 0$

Hamming Distance

while compairing two binary strings of equal length, Hamming distance is the no. of bit positions in which two bits are diff.

we perform XOR $a \oplus b$ and then count the total no. of 1s in the resultant string.

eg: $x = 11011001$
$y = 10011101$

XOR $\overline{}$
01000100

2 1's $d(x,y) = 2$.

Min. hamming distance : In a set of strings of equal lengths, the min. hamming distance is the smallest hamming distance b/w all possible pairs of strings in that set.

eg: 010, 011, 101

$d(010, 011) = 001 = 1$

$d(010, 101) = 111 = 3$ $\qquad d_{min} = 1$.

$d(011, 101) = 110 = 2$

Some GATE ques

Q. For a hamming code of parity bit $m = 8$ what is total bits & data bits.

$$2^m \geq d + m + 1$$

$$2^8 \geq d + 9 \qquad 256 \geq d + 9$$

$2^4 \geq d + 5$

$16 \geq d + 5$

$16 - 5 \geq d$

11

(15, 11)

Total bits are $2^m - 1 = 255$

data bits $= 255 - 8 = 247$

(255, 247)

eg: send data 1 0 1 1 0 1

$2^p \geq 6 + p + 1 \qquad p = 4. \qquad 6 + 4 = 10$

10	9	8	7	6	5	4	3	2	1
1	0	P_4	1	1	0	P_3	1	P_2	P_1

$P_1 \rightarrow$ 1 3 5 7 9
0 1 0 1 0

$P_2 \rightarrow$ 2 3 ~~5~~ 6 7 10
0 1 1 1 1

$P_3 \rightarrow$ 4 5 6 7
0 0 1 1

$P_4 \rightarrow$ 8 9 10
1 0 1

1 0 1 1 1 0 0 1 0 0

Suppose received is 1 1 1 1 1 0 0 1 0 0 $\qquad$ 1 0 0 1 $= 9_{10}$

10	9	8	7	6	5	4	3	2	1
1	1	1	1	1	0	0	1	0	0
		P_4				P_3		P_2	P_1

$P_1 \rightarrow$ 1 3 5 7 9 $\qquad P_1 \rightarrow 1$
0 1 0 1 1

$P_2 \rightarrow$ 2 3 6 7 10 $\qquad P_2 \rightarrow 0$
0 1 1 1 1

$P_3 \rightarrow$ 4 5 6 7 $\qquad P_3 \rightarrow 0$
0 0 1 1

$P_4 \rightarrow$ 8 9 10 $\qquad P_4 = 1$
1 1 1