**Vivekanand Education Society’s Institute of Technology Department of Computer Engineering**



**Subject: - Processor Architecture Lab**

**Class:- S.E. (D7) Semester:- IV Div:- A**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Roll | Name: |  |  |  |  |  |
| No: 46,27 | Shikhar Niranjan & Harish Kumar | | | | |
|  |  | | | | |
| Exp. | Title**:** |  |  |  |  |  |
| No: 16 |  | | | | | |
|  | Mini Project | | | | | |
|  |  | | | | | |
| DOP: | 19/03/2020 | | | DOS: | 01/04/2020 | |
| GRADE: |  | LAB OUTCOMES : |  | SIGNATURE: | | |
| LO1 |
|  |

**Aim –**

Designing a circuit for a 7-bit hamming code generator for a 4-bit input with logic diagram.

**Theory –**

**Hamming Code**

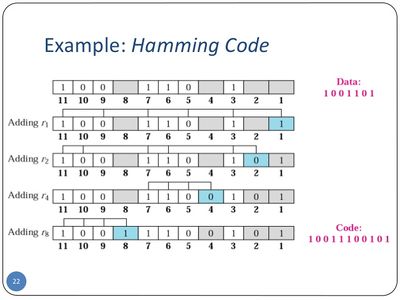
Hamming code is a block code that is capable of detecting up to two simultaneous bit errors and correcting single-bit errors.

## Encoding a message by Hamming Code

The procedure used by the sender to encode the message encompasses the following steps −

* **Step 1** − Calculation of the number of redundant bits.
* **Step 2** − Positioning the redundant bits.
* **Step 3** − Calculating the values of each redundant bit.

Once the redundant bits are embedded within the message, this is sent to the user.



(image courtesy – Google)

## Step 1 − Calculation of the number of redundant bits.

If the message contains m𝑚number of data bits, r𝑟number of redundant bits are added to it so that m𝑟 is able to indicate at least (m + r+ 1) different states. Here, (m + r) indicates location of an error in each of (𝑚 + 𝑟) bit positions and one additional state indicates no error. Since, r𝑟 bits can indicate 2r𝑟 states, 2r𝑟 must be at least equal to (m + r + 1). Thus the following equation should hold  2r ≥ m+r+1

## Step 2 − Positioning the redundant bits.

The r redundant bits placed at bit positions of powers of 2, i.e. 1, 2, 4, 8, 16 etc. They are referred in the rest of this text as r1 (at position 1), r2 (at position 2), r3 (at position 4), r4 (at position 8) and so on.

## Step 3 − Calculating the values of each redundant bit.

The redundant bits are parity bits. A parity bit is an extra bit that makes the number of 1s either even or odd. The two types of parity are −

* **Even Parity** − Here the total number of bits in the message is made even.
* **Odd Parity** − Here the total number of bits in the message is made odd.

Each redundant bit, ri, is calculated as the parity, generally even parity, based upon its bit position. It covers all bit positions whose binary representation includes a 1 in the ith position except the position of ri. Thus −

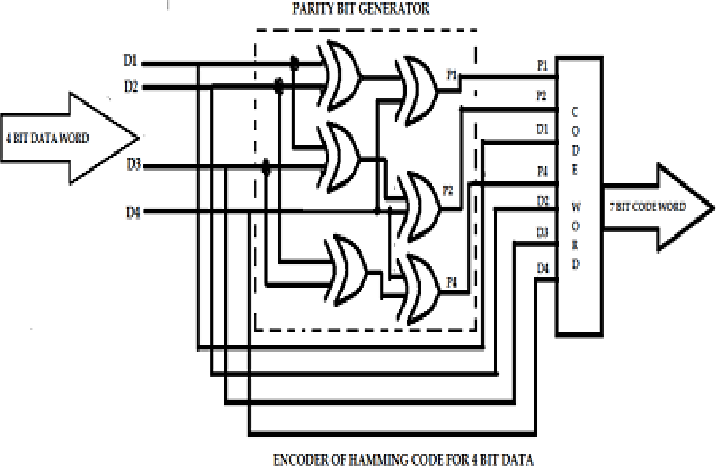
* r1 is the parity bit for all data bits in positions whose binary representation includes a 1 in the least significant position excluding 1 (3, 5, 7, 9, 11 and so on)
* r2is the parity bit for all data bits in positions whose binary representation includes a 1 in the position 2 from right except 2 (3, 6, 7, 10, 11 and so on)
* r3 is the parity bit for all data bits in positions whose binary representation includes a 1 in the position 3 from right except 4 (5-7, 12-15, 20-23 and so on)

## Decoding a message in Hamming Code

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

* **Step 1** − Calculation of the number of redundant bits.
* **Step 2** − Positioning the redundant bits.
* **Step 3** − Parity checking.
* **Step 4** − Error detection and correction

**Logic Diagram -**



## Steps –

## Step 1 − Calculation of the number of redundant bits

Using the same formula as in encoding, the number of redundant bits are ascertained.

2r ≥ m + r + 1 where m is the number of data bits and r is the number of redundant bits.

## Step 2 − Positioning the redundant bits

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

## Step 3 − Parity checking

Parity bits are calculated based upon the data bits and the redundant bits using the same rule as during generation of c1,c2 ,c3 ,c4 etc. Thus

c1 = parity(1, 3, 5, 7, 9, 11 and so on)

c2 = parity(2, 3, 6, 7, 10, 11 and so on)

c3 = parity(4-7, 12-15, 20-23 and so on)

## Step 4 − Error detection and correction

The decimal equivalent of the parity bits binary values is calculated. If it is 0, there is no error. Otherwise, the decimal value gives the bit position which has error. For example, if c1c2c3c4 = 1001, it implies that the data bit at position 9, decimal equivalent of 1001, has error. The bit is flipped to get the correct message.

Hamming code is very useful for error detection and correction.

**Conclusion –** The biggest benefit of the hamming code method is effective on networks where the data streams are given for the single-bit errors. The biggest drawback of the hamming code method is that it can solve only single bits issues.