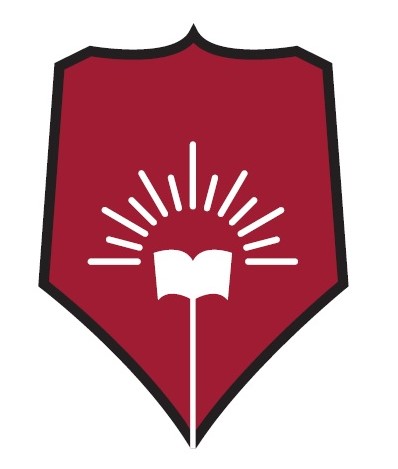
**Experiment No.: 9**

**Implement the Hamming**

**code **

1. **Aim:** Implement the Hamming code using C code.
2. **Objectives:** To introduce concepts of Hamming code.
3. **Outcomes:** The learner will be able to

* Analyze the concept of error detection and correction.
* Recognize the need for Hamming code.

1. **Hardware/Software required:** Turbo C, C++ Integrated Development Environment & compile.

1. **Theory:**

Hamming code is a popular error detection and error correction method in data communication. Hamming code can only detect 2 bit error and correct a single bit error which means it is unable to correct burst errors if may occur while transmission of data.

Hamming code uses redundant bits (extra bits) which are calculated according to the below formula:-

**2r ≥ m+r+1**

Where **r** is the number of redundant bits required and **m** is the number of data bits.

**For r = 1:**

2^1≥4+1+1

2≥ 6 (𝒏𝒐𝒕 𝒕𝒓𝒖𝒆)

**For r = 2:**

2^2≥4+2+1

4≥ 7 (𝒏𝒐𝒕 𝒕𝒓𝒖𝒆)

**For r = 3:**

2^3≥4+3+1

8≥ 8 ( 𝒆)

So, the total number of redundant bits will be 3.Therefore, total length of Hamming code will be 4 + 3 = 7 bits

Let the redundant bits be name as r0, r1 and r2

So, position of r0 = 2^0 = 1

position of r1 = 2^1 = 2

position of r2 = 2^2 = 4

The Hamming code will look like this:



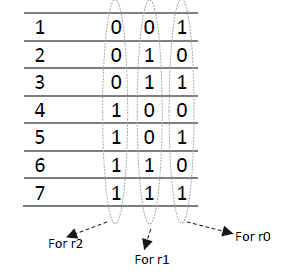
Fig. 1.1 model of hamming code for hamming (4,7)

After putting the values of m1, m2, m3 and m4 from data bits:



Fig. 1.2 model of hamming code for Hamming (4,7) with data bits

Now, we need to find the values of r0, r1 and r2. For that lets make a table decimal to binary conversion table from 1 to 7.



**Note: Even parity.**

**For r0:**

Check for the numbers in the above table, which has ‘1’ at the most significant bit i.e. the last bit form left.

1,3,5 and 7 has ‘1’ at most significant bit.

Copy the values at the positions 1,3,5 and 7 from Fig 1.2.

Therefore, we have: **r0 1 0 1**

Since, for even parity, number of 1’s should be even and we have even number of 1’s.

Therefore, **r0 = 0**

**For r1:**

Check for the numbers in the above table, which has ‘1’ at the 2nd bit form left.

2,3,6 and 7 has ‘1’ at 2nd bit position.

Copy the values at the positions 2,3,6 and 7 from fig 1.2.

Therefore, we have: r1 1 1 1

Since, for even parity, number of 1’s should be even and we have odd number of 1’s.

Therefore, r1 = 1

For r2:

Check for the numbers in the above table, which has ‘1’ at the first bit form left.

4,5,6 and 7 has ‘1’ at first bit position.

Copy the values at the positions 4,5,6 and 7 from fig 1.2.

Therefore, we have: r2 0 1 1

Since, for even parity, number of 1’s should be even and we have even number of 1’s.

Therefore, r2 = 0

Therefore, The Hamming code is:



Or, The Hamming code for 1011 is 0110011

Code:

Output:

1. **Conclusion :**

(Students should write conclusion on their own)

1. **Viva Questions:**

* What is Hamming code? How does it works?
* Explain Hamming code generation and correction with example?
* State advantages and disadvantages of Hamming codes.

1. **References:**
2. A.S. Tanenbaum, “Computer Networks”, Pearson Education, (4e).
3. B.A. Forouzan, “Data Communications and Networking”, TMH (5e).
4. James F. Kurose & K W Ross: Computer Networking: A Top Down Approach, Pearson Education (LPE).