# Enhanced Detection of Double Adjacent Errors in Hamming Codes through Selective Bit Placement

[1]Lintu K Babu , [2]Hima Sara Jacob
[1]M Tech Student, [2]Assistant Professor
[1] Department of Electronics And Communication
[1] Mangalam College Of Engineering, Ettumanoor, Kottayam, Kerala,India

***Abstract*— This paper explain the Enhanced detection of Double adjacent Errors in Hamming codes. In 1950's Richard Hamming invented the Hamming code. Designed to correct single bit errors. Hamming Codes are used in computing, telecommunication, and other applications. Codes are simple to construct for any word length. It can be easy to Encode and decode.2D Hamming Product Code is used to detect and correct errors. But, this fails to detect adjacent errors. Detecting double adjacent errors using Selective Bit Placement Algorithm. In this paper the algorithm for hamming code is discussed and then implementation of it in verilog is done to get the result. Here code is implemented in verilog in which 4 bit of information data is transmitted. Delay, area and power analyses were carried out using Xilinx ISE-13.2.The results obtained are presented and compared. The enhanced detection is achieved by selective bit placement strategy.**

***Index Terms*— Error correction codes (ECCs), Hamming codes.**

## I. INTRODUCTION

**Hamming codes** are a family of linear error-correcting codes, generalize the Hamming code invented by Richard Hamming in 1950. Hamming codes can detect two-bit errors or correct one-bit errors without detection of uncorrected errors. By contrast, the simple parity code cannot correct errors, and can detect only an odd number of bits in error. Hamming codes are perfect codes, that is, they achieve the highest possible rate for codes with their block length and minimum distance 3[1].

In all cases, the data is encoded when it is written into the memory and decoded when read. The encoding and decoding latency directly affect the memory access time. To minimize this effect ECCs [4] for which decoding is simple are used in most cases. In this paper, a technique to maximize the probability that a Hamming code detect double adjacent errors and that a parity extended Hamming code detects triple adjacent errors is presented. This is achieved by placing the bits of the word such that those adjacent errors provoke a syndrome value that is different from those caused by single errors [2].

## II. HAMMING CODES

Hamming codes are designed to correct single bit errors. Family of (n, k) block error-correcting codes with parameters:

Block length: $n = 2^m - 1$

Number of data bits: $k = 2^m - m - 1$

Number of parity check bits: $n - k = m$

Minimum distance: $d_{min} = 3$ (1)

Table I Hamming Codes Parameters

| k | n |
|---|---|
| 4 | 7 |
| 11 | 15 |
| 26 | 31 |
| 57 | 63 |
| 120 | 127 |
| 247 | 255 |
| 503 | 511 |

Table II Shortened Hamming Codes Parameters

| k | n |
|---|---|
| 8 | 12 |
| 16 | 21 |
| 32 | 38 |
| 64 | 71 |
| 128 | 136 |
| 256 | 265 |

In memory applications, the number of information bits *k* is commonly a power of two and Hamming codes are shortened to fit that word length as illustrated in Table II. Hamming codes are linear codes and can be generated and decoded using the generator and parity-check matrices, respectively. These matrices have a canonical form, but also equivalent non-systematic code matrices can be obtained by column permutations and other row operation.

Table III Generation Algorithm Code (7,4)

| **Position** | **c1** | **c2** | **c3** | **c4** | **c5** | **c6** | **c7** |
|---|---|---|---|---|---|---|---|
| **Binary** | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| **Content** | p1 | p2 | d1 | p3 | d2 | d3 | d4 |
| **c1(p1)** | -- | | 011 | | 101 | | 111 |
| **c2(p2)** | | -- | 011 | | | 110 | 111 |
| **c4(p3)** | | | | -- | 101 | 110 | 111 |

Algorithm to generate Hamming code words from information bits is as follows:

a) Positions are numbered from 1 to n;
b) Positions are written in their binary form (1, 10, 11, etc.)
c) Bits in positions 2^r are parity bits for those other positions

Where the binary form of those positions has the bit r + 1 set to one. For instance, in the Hamming code (7,4) with n = 7, k = 4 and m = 3, positions c1, c2 and c4 are parity bits (p1, p2 and p3) and the information bits (d1, d2, d3 and d4) are placed in order in the rest of positions as shown in Table III.
Parity bits are calculated as follows:

c1 =c3 + c5 + c7 or p1 = d1 + d2 + d4
c2 =c3 + c6 + c7 or p2 = d1 + d3 + d4
c4 =c5 + c6 + c7 or p3 = d2 + d3 + d4.

To checks code word, the parity bits can be recalculated. Result compared to the original set of parity bits. If they match, then no error was introduced (or it is not detected). Otherwise, an error is detected. The non-matching parity bits can provide the information of the bit that was flipped, and then error can be corrected.

As an example data bit a3a2a1a0 is 1100 and p2p1p0 is 001. Suppose 1100001 becomes 1000001.Recalculate p2p1p0 is 111. Difference (bit-wise XOR) is 110. This difference is called syndrome - indicates the bit in error. It is clear that a2 is in error and the correct data is a3a2a1a0 is 1100.The syndrome can be calculated directly in one step from the bits a3 a2 a1 a0 p2 p1 p0. This is best represented by the following matrix operation where all the additions are mod 2.

$$\begin{array}{c} a_3\,a_2\,a_1\,a_0\,p_2\,p_1\,p_0 \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \\ p_2 \\ p_1 \\ p_0 \end{bmatrix} = \begin{bmatrix} s_2 s_1 s_0 \end{bmatrix}$$

Fig 1. Syndrome calculation

If vector is null vector, then the current value of the word is an actual code word. In any other case, an error occurred in the code word. Example - 1100001 becomes 1010001 -a2 and a1 are erroneous - syndrome is 011. This indicates erroneously that bit a0 should be corrected. One way of improving error detection capabilities to adding an extra check bit which is the parity bit of all the other data and parity bits. This is an (8, 4) single error correcting/double error detecting (SEC/DED) Hamming code [3]. For example Single error is 11001001 becomes 10001001. Syndrome is 1110 - indicating that a2 is erroneous. Two errors - 11001001 becomes 10101001.Syndrome is 0011 indicating an uncorrectable error.

## III. SELECTIVE BIT PLACEMENT STRATEGY

As pointed out before, there are special combinations of double and triple bit errors that are detected and not miscorrected in the shortened Hamming code and the parity extended version, respectively. The bit positions for those combinations are randomly distributed through the word. The objective is to reorder the bits of the code word to maximize the adjacency of the special combinations.

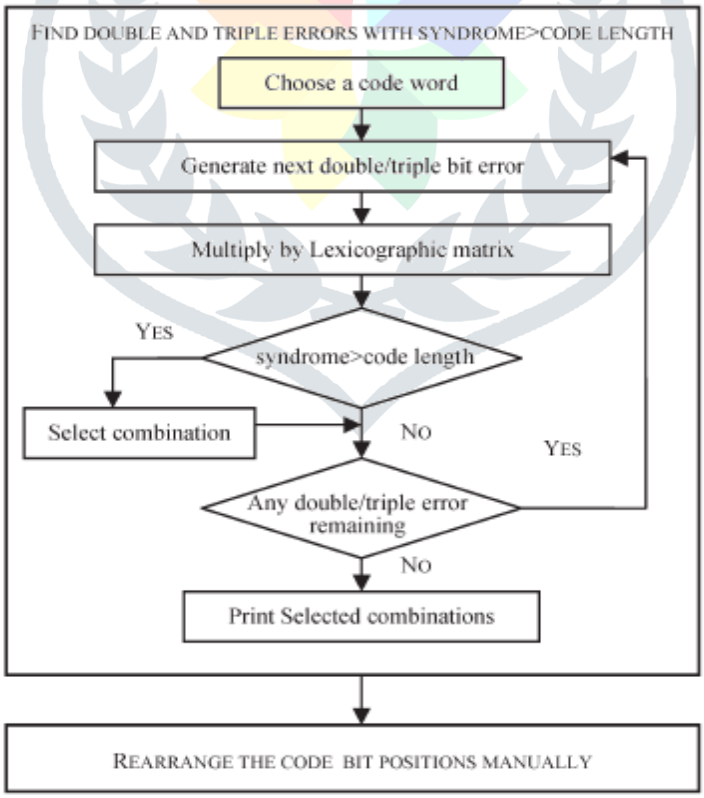


Fig 2. Selective bit placement strategy procedure

To achieve this goal, the following procedure is followed:

- Using a software program, find all double and triple error combinations which produce a syndrome that does not match any of the ones caused by a single error.
- To reorder the word manually to maximize the number of adjacent bits errors.
- To find all the possible combinations of double and triple error bits, the program executing steps:
- Hamming code is selected.
- All 2-error and 3-error combinations code word are generated.
- The error words generate multiplied with the lexicographic check matrix.
- Those bit error combinations producing a syndrome, that higher than the maximum bit position of the code are selected.
- Positions are printed out.

## IV. SIMULATION RESULTS

We have written a code in verilog for hamming code. The code is running succefully. In this code we are initializing a data of 4 bit and calculating its parity bit by the above general algorithm. It requires calculating the parity bit by matrix multiplication and modulo-2 addition. The parity bit required for four bit data is three bit. These bits are calculated and the We have done encoding and decoding of the data in the same code. A variable syndrome is used to decode the data and to calculate the parity bit. The syndrome has two useful properties. First if the syndrome is all zeros, the encoded data is error free. But if the syndrome has a nonzero value, then flip the encoded bit that is in the position of the column in [H] that matches the syndrome will result in a valid code word. we will show the input/output ports of hamming encoder and decoder.

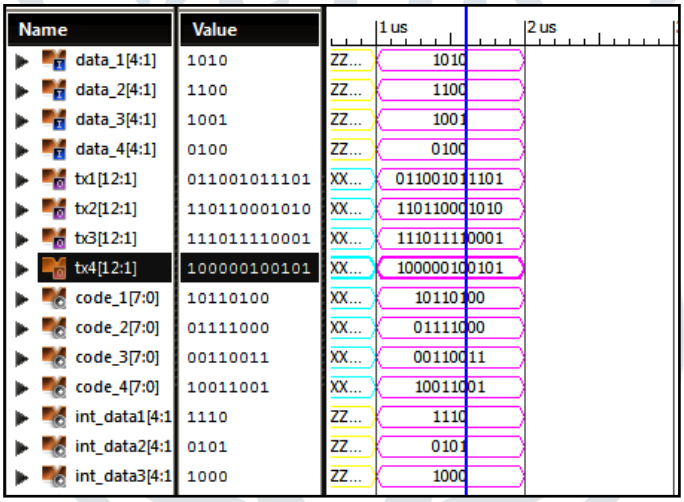


Fig 3. Wave form of hamming encoder

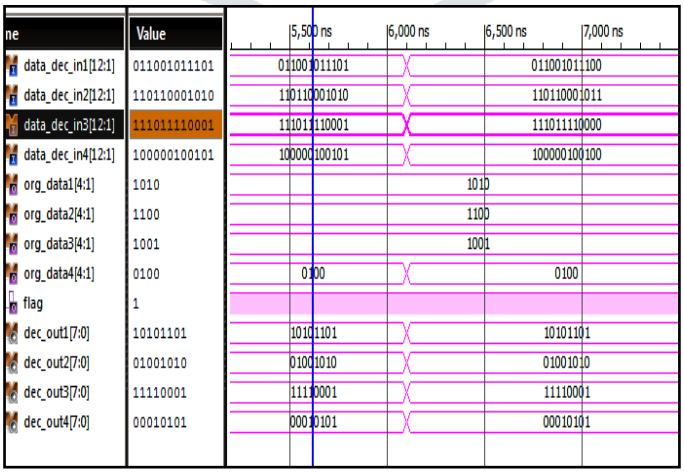


Fig 4. Wave form of hamming decoder

Table IV Comparison Table

| PARAMETER | CONVENTIONAL | | ENHANCED | |
|---|---|---|---|---|
| | ENCODER | DECODER | ENCODER | DECODER |
| POWER(W) | .069 | .137 | .072 | .138 |
| AREA NO CELLS | 185 | 334 | 187 | 328 |
| DELAY (NS) | 12.5 | 21.31 | 12.308 | 22.02 |

## V. CONCLUSION

In this paper, technique to maximize the probability of detecting errors in hamming codes. The enhanced detection is achieved by selectively placing the bit. Effective method of achieving a large error detection rate. The proposed scheme does not require any additional circuitry. The area, power and speed will be the same as with the traditional method. Delay, area and power analyses were carried out using Xilinx ISE-13.2.The results obtained are presented and compared. The enhanced detection is achieved by selective bit placement strategy.

## REFERENCES

[1] Alfonso Sanchez - Macian "Enhanced detection of double and triple adjacent errors in hamming codes through selective bit placement" IEEE Transactions on device and materials, vol 12.

[2] R.W Hamming "Error detecting and error correcting codes", Bell Syst.Tech J., vol.29, pp.147-160.

[3] S. Lin and D. J. Costello,"Error Control Coding", 2nd ed. Englewood Cliffs, NJ: Prentice-Hall.

[4] R. C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," IEEE Trans.Device Mater. Rel.,vol. 5, no.3, pp.301–316, Sep.2005.