INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY, BANGALORE

2019

A PROJECT REPORT ON

Implementation of Error Detection and Correction Algorithm In ARM CORTEX-M4

Submitted by:

Ajay Gummuluri [MT2018503]

Anmol Jain [MT2018505]

Modupalli Rammohan [MT2018514]

M.V. Sai Krishna [MT2018527]

Single Bit Data Errors:

The change in one bit in the whole data sequence is called “Single bit error”. Occurrence of single bit error is very rare in serial communication system. This type of error occurs only in parallel communication system, as data is transferred bit wise in single line, there is chance that single line to be noisy.

Hamming Code:

This error detecting and correcting code technique is developed by R.W.Hamming. This code not only identifies the error bit, in the whole data sequence and it also corrects it. This code uses a number of parity bits located at certain positions in the code word. The number of parity bits depends upon the number of information bits. The hamming code uses the relation between redundancy bits and the data bits and this code can be applied to any number of data bits.

In this project, we are focussing on single bit error detection and correction.

Procedure:

Step 1: An 8 bit input data is to be encoded in to 12 bit data by adding parity check bits to it. For an 8 bit data, the parity check bits should be placed at positions 1,2,4,8. The data and parity bit positions in 12 bit encoded data are as follows : d11,d10,d9,d8,c7,d6,d5,d4,c3,d2,c1,c0.

Step 2: The way in which we expand the 8 bit value is by separately clearing all the bits apart from the ones we will use for the check bits. We clear all the bits in the LSB, then the result we get we shift left two bit positions. Next, we clear all bits in the value r0 except d1, d2, and d3. The result we get is combined with d0 using an ORR operation. In the last step, same procedure is followed but with the positions d4 – d7.

Step 3: Now, we have to calculate the parity for every check bit and place that parity value in those positions. It is done as follows:
For C0, parity is calculated for the positions 1,3,5,7,9,11 because they have 1 at LSB bit.
For C1, parity is calculated for the positions 2, 3, 6, 7, 10, 11 because they have 1 at 1^{st} bit
For C2, parity is calculated for the positions 4,5,6,7 because they have 1 at 2^{nd} bit.
For C3, parity is calculated for the positions 8, 9, 10, 11 because they have 1 at 3^{rd} bit.

Now, we have 12 bit encoded data. For error detection and correction, we will flip one of the bit positions.

Step 4: To detect the error, the procedure is as follows:
Clear all the parity check bits at the respective positions in the encoded data. It is done by doing AND operation between the data and the same data with 0's at check bit positions.
Now, calculate the parity for the check bit positions by using the same procedure mentioned in step 3.
Now the binary value of C3C2C1C0 subtracted by 1 will give the position of the error.

Step 5: For correcting it, a value of 1 is stored at LSB position in a register. It is then shifted by the value of error position detected. Now, EOR operation is performed between the value in that register and the erroneous data. This will give the Corrected value.