CENG290 Data Communications
Labwork 8

Error Correction Using Multiple Parity Bits

Multiple parity bits can be used for error detection for any length of data. They may also help with error correction in 1-bit error cases. The steps to calculate the parity bits are as follows:

1. Create a codeword using data bits. Positions in this codeword that are powers of 2 (e.g. 1, 2, 4, 8, etc) are reserved for parity bits and the rest are for the original data bits. Therefore, this codeword will be slightly longer than the length of the original data.

2. Write all bits in binary (shown below). Calculate each parity bit one by one by using the corresponding least significant bit set (excluding the parity bit itself):

	1	2	3	4	5	6	7	8	9	10	11	12	...
p1	**x**	0	1	0	1	0	1	0	1	0	1	0	...
p2	0	**x**	1	0	0	1	1	0	0	1	1	0	...
p4	0	0	0	**x**	1	1	1	0	0	0	0	1	...
p8	0	0	0	0	0	0	0	**x**	1	1	1	1	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...

(a) Parity bit 1: 3, 5, 7, 9, 11, etc.

(b) Parity bit 2: 3, 6, 7, 10, 11, etc.

(c) Parity bit 4: 5, 6, 7, 12, etc.

(d) . . .

3. Total count of 1 bits in the given set determines the value of chosen parity bit. In case of even parity, even number of 1's are represented with 0 and odd number of 1's are represented with 1.

When a bit sequence including both the data and parity bits are received by the target, data bits must be checked for errors:

1. Calculate parity bits according to the received data bits.

2. If the calculated and received parity bits are the same, there is no error.

3. If the calculated and received parity bits differ, identify and correct the incorrect bit with positions of incorrect parity bits.

For example if the parity bits 1, 2 and 4 are wrong, then the 7th bit $(1 + 2 + 4)$ is wrong. If only one parity bit is wrong, then the wrong bit is the parity itself.

Exercises

1. Randomly generate two 8-bit and two 16-bit bit sequences in MATLAB using `round(rand(1, 8))` and `round(rand(1, 16))` commands and calculate parity bits <u>on paper</u>.

2. During transmission of the following 12-bit sequences (data + parity), one bit per sequence has been received incorrectly. Find the incorrect bits.

   (a) `001110011011`
   (b) `110110110010`
   (c) `111000111111`

3. Write a MATLAB function which calculates parity bits for a given fixed 8-bit length data:

   ```
   function result = calculateParity(data)
     ....
   end
   ```

4. (Optional) Modify your function to calculate parity bits for any length of data.