DLDCA Bonus Week2

DLDCA TA's

August 2025

1 Hamming Codes

Hamming codes were invented in 1950 by Richard W. Hamming as a way of automatically correcting errors introduced by punched card readers, although they can be used for any binary data that is susceptible to errors. It makes use of strategically placed parity[1] bits to detect up to two errors and correct up to one error. How does parity help us detect errors? A bit flip increases/decrease number of 1's in the bitstring exactly by 1 which flips the parity. Thus if the parity bit does not match with the actual parity of the bitstring we can conclude it has an error.

In some sense, this can protect us from spontaneous data flips and also helps us detect when out data is corrupted.

2 How does it Work?

We are going to be focusing on 16-11 Hamming codes, that is , for every 16 bits of encoded data we have, we will have 11 bits of actual data and 5 bits for error checking.

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Table 1: A block of 16 bits of data

The 5 highlighted bits are the ones used for error correction.
These bits encode the parity of the underlying information. Each parity bit

[1] parity in this context refers to if or not the number of 1's in a bitstring is even or odd.

is responsible for its own subset of the block. The bit in position 1 is the parity bit for columns 2 and 4. This means that bit in position 1 is the XOR (exclusive OR, used to detect parity) of all bits in column 2 and 4. The bit in position 2 is meant for the last two columns. The bit in position 4 is for rows 2 and 4, and the bit in position 8 is for the last two rows.

The parity bit at position 0 is currently unused as it cannot be used for message bits, due to it being sandwiched between the other bits which are meant for error correction. However, extended hamming code makes use of this bit as an overall parity bit to detect cases with two errors. Therefore, if the subset parity bit checks find and error, but the overall parity bit says an error has not occured, there must have been two errors. However, Hamming codes cannot to detect more than 2 errors and incorrectly corrects the wrong bit in the case of an odd number of errors (3,5,7...) or detects a two error case with an even number of errors (2,4,6...).

3 The Task

Given the above information, make a self correcting circuit which holds 11 bits of data which uses your memory cell of choice (either T or D) units and also maintains this checks and corrects the error if it occurs.

The input to the module will be all 16 bits of data (parity bits are already filled in input), clk, and reset.

The output for the module should be the 16 bits of data (bits which can be corrected using parity bits must be corrected), and one more bit indicating if the data is corrupted or not.