1.3 Day 3 – Error Correction

When we store data on hard disk, USB drive etc., some bits may get corrupted. Similarly, when we store data in SRAM, some bits may be stored incorrectly. Errors may also occur when data is read from memories. When data is communicated over a noisy channel, some 1's may change to 0's and some 0's may change to 1's. Hamming codes were invented to detect and correct bit errors. Consider a n-bit data. We will add k additional "check" bits for the purpose of error detection. The state of the k check bits can tell us whether or not there is an error, and if an error has occurred, where the error is.

1. There is one case where there is no error
2. Since the total number of bits is $4 + k$, if a single error were to occur, it could happen in any of the $n + k$ positions.

Thus the minimum value of k must satisfy

$$2^k \geq n + k + 1 \tag{1.1}$$

When $n = 4$, the smallest value of k is 3. Thus, we have to store 4+3=7 bits instead of storing 4 information bits. If the information bits are labelled I3,I2,I1,I0 and the check-bits are labelled C2,C1,C0, they are stored as shown in the table below.

Bit Positions	7	6	5	4	3	2	1
Binary code for bit positions	111	110	101	100	011	010	001
What is stored	I3	I2	I1	C2	I0	C1	C0

The check bits are computed as follows. Note that they are essentially even parity bits.

$$C0 = EXOR(I0, I1, I3) \tag{1.2}$$
$$C1 = EXOR(I0, I2, I3) \tag{1.3}$$
$$C2 = EXOR(I1, I2, I3) \tag{1.4}$$

1.3.1 EXAMPLE

Suppose the information bits are 1101. Then we store 7 bits found as below.

Bit Positions	7	6	5	4	3	2	1
Binary code for bit positions	111	110	101	100	011	010	001
What is stored	1	1	0	0	1	1	0

Suppose a single bit error were to occur and we read/receive the pattern 1100010. We generate the check bits again as follows.

Bit Positions	7	6	5	4	3	2	1
Binary code for bit positions	111	110	101	100	011	010	001
What is stored	1	1	0	0	0	0	1

The received check-bits are 010. The computed check-bits are 001. The EXOR of these two gives us 011, which points to the position 3 where the error is.

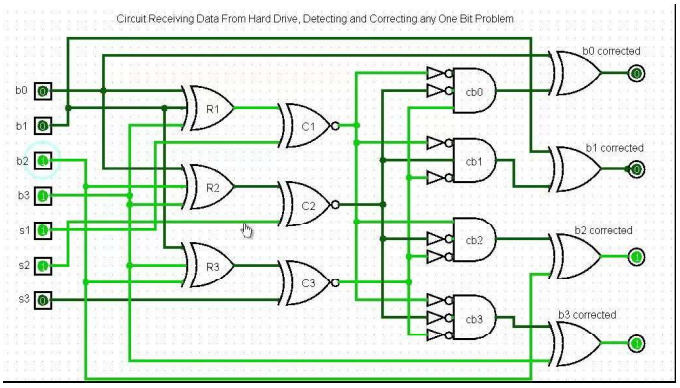

Figure 1.1: ECC circuit based on Hamming Code (7,4)

1.3.2 EXAMPLE

If the information bits are 1111, they are stored as:

Bit Positions	7	6	5	4	3	2	1
Binary code for bit positions	111	110	101	100	011	010	001
What is stored	1	1	1	1	1	1	1

Suppose the error occurs at position 1 and the received/read data is 1111110. The recomputed check bits are 111 (since there is no error in the information bits). The EXOR of 111 and 110 is 001. Once again, we can see that the code points us to the error in position 1.

1.3.3 Problem

To detect and correct bit errors in transmission/storage, it is common to use error detection and correction circuits. We will consider a single-bit error detection and single-bit error correction circuit. The purpose of the error correction circuit (ECC) is to read in a 7-bit Hamming code and output the information bits, with any correction necessary. The circuit is shown in Figure 1.1 below.

1.3.4 Assignment

Understand the circuit. Answer the following questions to yourself.

- What is the role of the XOR gates R1, R2, R3?
- What is the role of the XNOR gates C1, C2, C3? What is the role of the AND gates G0, G1, G2, G3?
- What is the role of the XOR gates at the output?

Enter the circuit using the online tool available at `https://circuitverse.org/`. Apply a sample input and observe the outputs of the circuit. Think of a way to apply all input combinations to the circuit.