Note for lecture 03:

Definition: a "<u>parity check code</u> (or linear code)" is a linear transformation from the string of data bits to the string of data bits and parity checks.

Example: single parity check code

$$\underbrace{1011000}_{\text{data bits}} \xrightarrow{\text{transform}} \underbrace{1011000}_{\text{data bits}}\underbrace{1}_{\text{parity check}}$$

In general, there can be K data bits and
L parity checks.

Definition: a "<u>code word</u>" is the result of the transform of a certain parity check code.

→ A code word is the data bits plus parity checks for
① single parity check
② horizontal & vertical parity check
③ CRC

a form of a code word: $\boxed{\overbrace{10011\cdots0}^{K}\,\Big|\,\overbrace{10\cdots1}^{L}}$

The sender of data sends the code word to the receiver, who ~~then~~ will decode the code word to
① see if there is any bit error in the code word;
② retreive the original data bits.

The receiver will not be able to detect bit error if the error has changed the <u>code word</u> (say, C_A) to another <u>code word</u> (say, C_B), in which case, ~~for~~ from the receiver's viewpoint, it is entirely possible that code word C_B was obtained by a certain string of data bits and was not because of the error.

Therefore, a useful criterion to measure the effectiveness of a parity check code is to look at the smallest number of bit changes that can convert one code word into another, which we say to be <u>the minimum distance of a code</u>. A longer minimum distance is better, because it would take more bit errors to make a data receiver unable to detect an error; in other words, such a parity check code is more resilient to bit errors!

Exercise: show that the minimum distance of a code using a single parity check is 2.

Answer: We can first show that no two code words in this case can be differed by one.

prove by contradiction

Suppose code words X_1 and X_2 are differed by only one bit, then

→ X_1 and X_2 cannot both have even number of 1s.

→ either X_1 or X_2 must have odd number of 1s, which cannot be a code word.

→ a contradiction.

Next, we give a witness, i.e., there exists two code words such that the distance in between is 2.

$$S_1(D) = D^2 + D \longrightarrow C_1(D) = 0$$

$$S_2(D) = D^2 + 1 \longrightarrow C_2(D) = 0$$

and the distance between

$$S_1(D)\cdot D + C_1(D)$$

and

$S_2(D)\cdot D + C_2(D)$ is 2 ※

Reasoning CRC:

represent a code word by $X(D)$

$$X(D) = \underbrace{S(D)}_{\text{its coefficients are data bits}} \cdot D^L + \underbrace{C(D)}_{\text{its coefficients are parity checks}}$$

by definition of a code word ⇒

By choosing a generator polynomial, $g(D)$ and compute $\frac{S(D)\cdot D^L}{g(D)}$,

we have $S(D)\cdot D^L = g(D)\cdot z(D) + C'(D)$

use ~~let~~ the remainder polynomial $C'(D)$ ~~be~~ as parity checks. $\Rightarrow C(D) = C'(D)$

$$\Rightarrow X(D) = S(D)\cdot D^L + C(D)$$

$$= g(D)\cdot z(D) + C'(D) + C(D)$$

$$= g(D)\cdot z(D) \quad \text{(according to modular 2 computation)}$$

which means $g(D)$ divides $X(D)$.

Send $X(D)$ to the receiver.

use $e(D)$ to represent errors introduced along the sending path.

Then the receiver gets $y(D) = X(D) + e(D)$

compute $\frac{y(D)}{g(D)}$, and we see if $e(D) = 0$

then $g(D)$ must divide $y(D)$ and remainder $= 0$.