Quaid-e-Awam University of Engineering, Science and Technology

بسم الله الرحمن الرحيم

In the name of Allah, the Most Merciful, the Most Kind

Date: 23-09-2021

BCS 103
Digital Logic & Computer Architecture

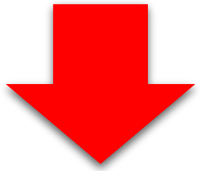
Lecture 11 and 12

Department of Computer Science

Classification of binary codes

The codes are broadly categorized into following four categories.

- **Weighted Codes**
  - BCD (8421)
  - 6311
  - 2421
  - 642-3
  - 84-2-1
- **Non-Weighted Codes**
  - Excess-3 Codes
  - Gray Codes
- **Alphanumeric Codes**
  - ASCII
  - EBCDIC



- **Error Detecting Codes (Parity)**

Error Detection / Correction Codes

What is Error?

The data can be corrupted during transmission (from source to receiver). It may be affected by external noise or some other physical imperfections. In this case, the input data is not same as the received output data. This mismatched data is called “Error”.

Types of Errors

In a data sequence, if 1 is changed to zero or 0 is changed to 1, it is called “Bit error”.

There are generally 3 types of errors occur in data transmission from transmitter to receiver. They are

- Single bit errors
- Multiple bit errors
- Burst errors

Error Detecting Codes

In digital communication system errors are transferred from one communication system to another, along with the data.

For effective communication, data should be transferred with high accuracy .

This can be achieved by first detecting the errors and then correcting them.

Error Detecting Codes

We use some redundancy codes to detect these errors, by adding to the data while it is transmitted from source (transmitter). These codes are called "**Error detecting codes**".

Types of Error detection

1. Parity Checking
2. Cyclic Redundancy Check (CRC)
3. Longitudinal Redundancy Check (LRC)
4. Check Sum

Error Correcting Codes

The codes which are used for both error detecting and error correction are called as “Error Correction Codes”. The error correction techniques are of two types. They are,

1. Single bit error correction
2. Burst error correction

The process or method of correcting single bit errors is called “**single bit error correction**”. The method of detecting and correcting burst errors in the data sequence is called “**Burst error correction**”.

Hamming Code

This error detecting and correcting code technique is developed by R. W. Hamming . This code not only identifies the error bit, but it also corrects it. This code uses a number of parity bits located at certain positions in the codeword.

This code can be applied to any number of data bits.

Hamming code or Hamming Distance Code is the best error correcting code we use in most of the communication network and digital systems.

What is a Redundancy Bit?

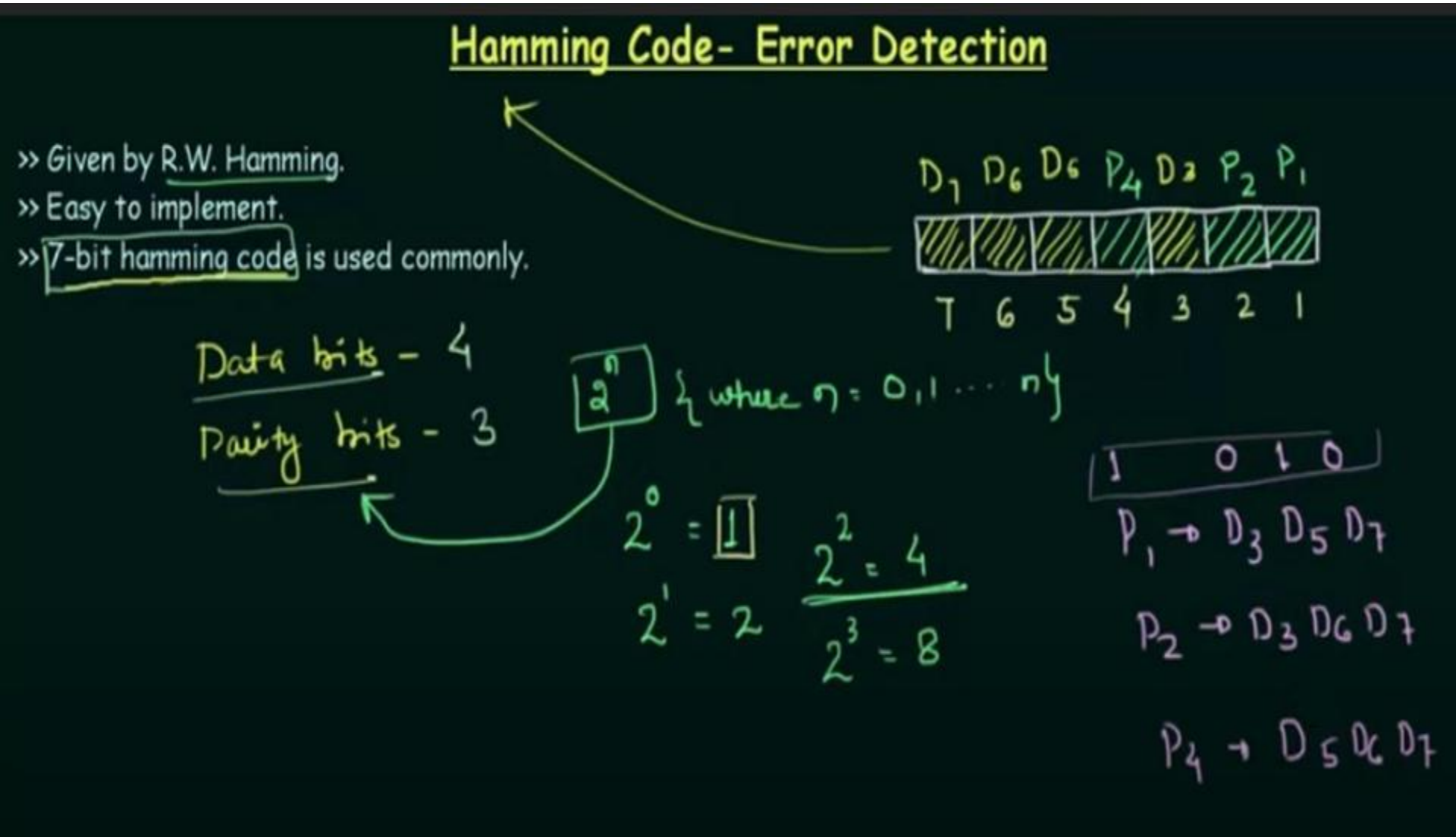
Redundancy means “The difference between number of bits of the actual data sequence and the transmitted bits”. These redundancy bits are used in communication system to detect and correct the errors, if any.

How the Hamming code actually corrects the errors?

In Hamming code, the redundancy bits are placed at certain calculated positions in order to eliminate errors. The distance between the two redundancy bits is called “Hamming distance”.

To understand the working and the data error correction and detection mechanism of the hamming code.

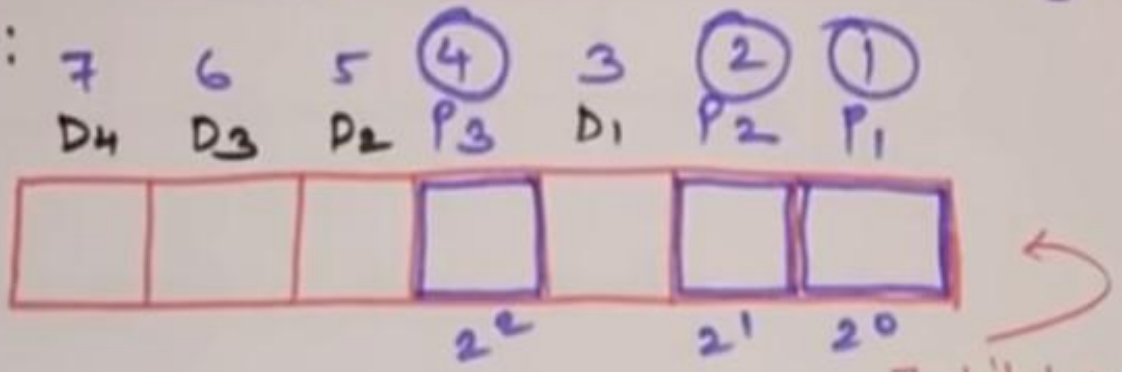
About Hamming Code



• HAMMING CODES •

- Hamming codes are linear block codes.
- It is an error correcting code.
- Parity bits are used here.
- They are inserted in between the data bits.
- The most commonly used is 7-bit hamming code.

→ Structure:



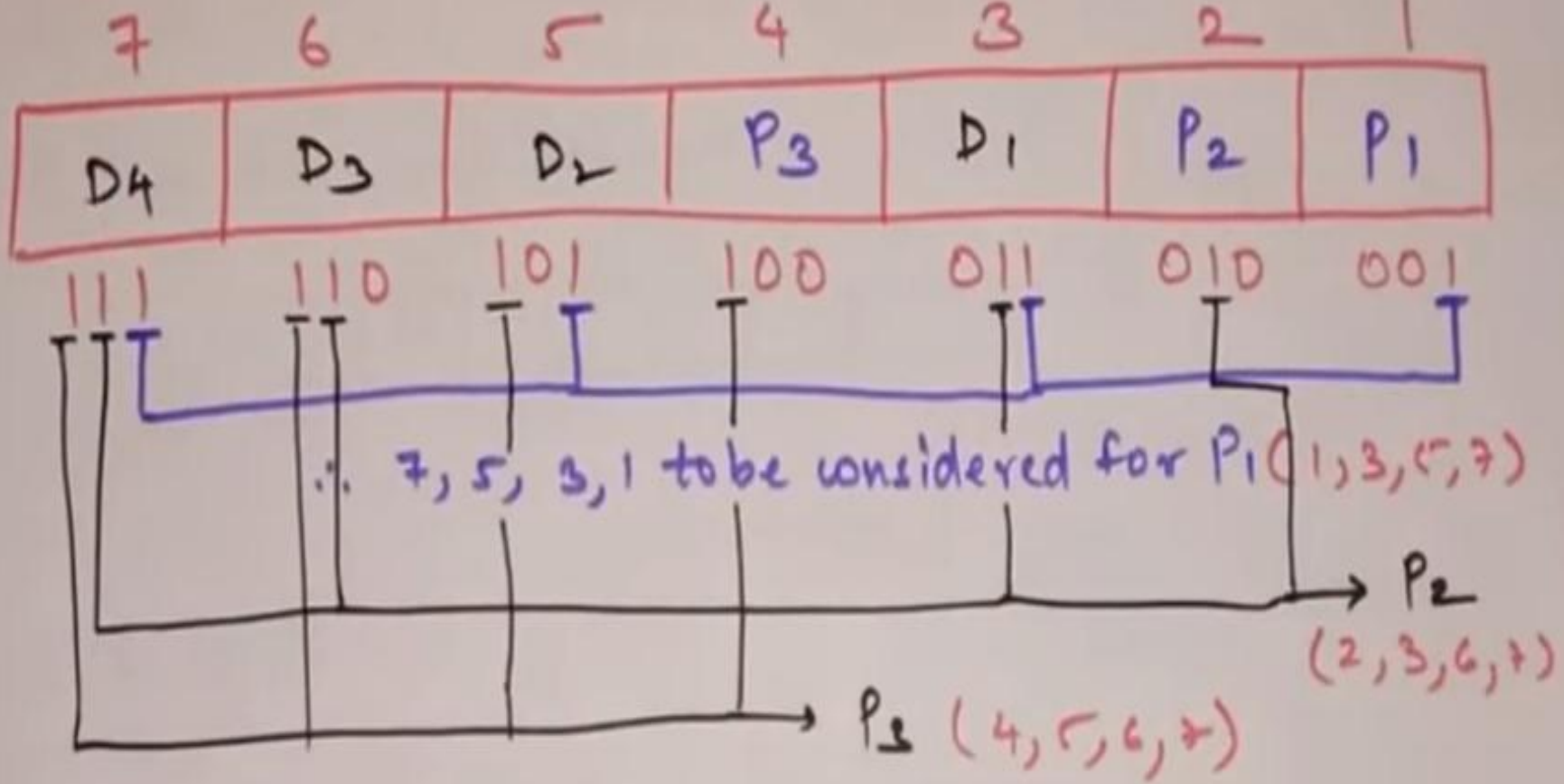
7-bit hamming code.

D → Data bits

P → Parity bits

2^m, m = 0, 1, 2, 3, ...

- Deciding the values for parity bits :



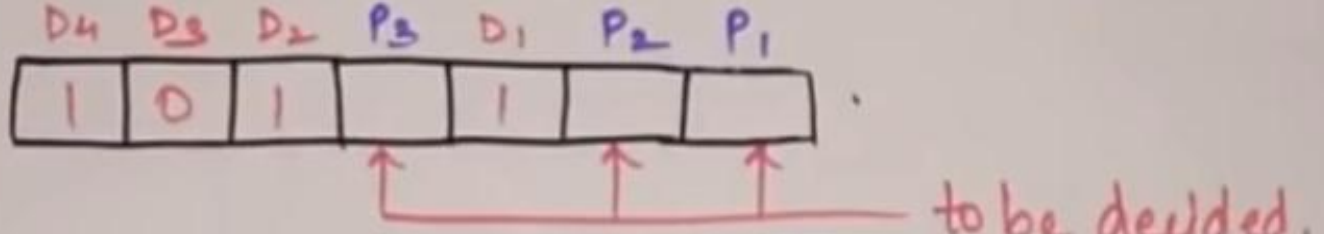
Two cases : Even Parity

Odd Parity

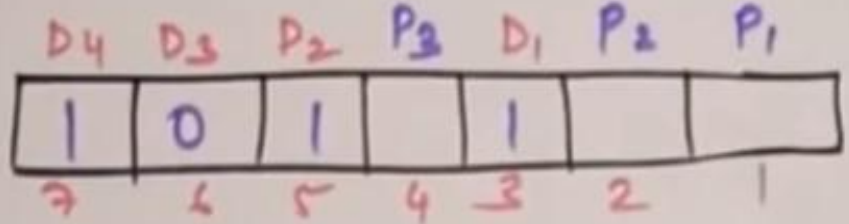
• A bit word 1011 is to be transmitted. Construct the even parity seven-bit hamming code for this data.

→

Step 1: The codeword format.



Step 2: Decide P_1

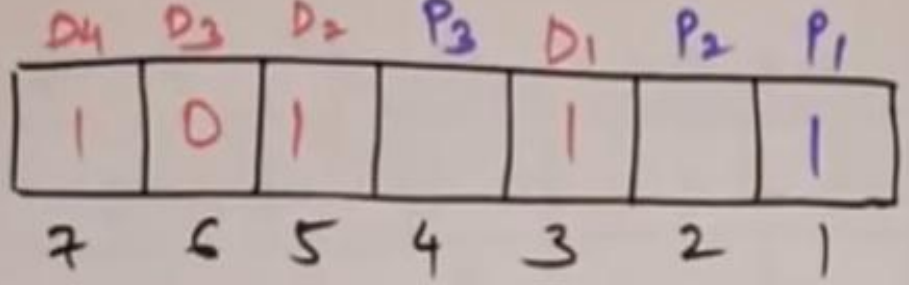


(Even Parity)

Considered bits : 1, 3, 5, 7 = , 1, 1, 1

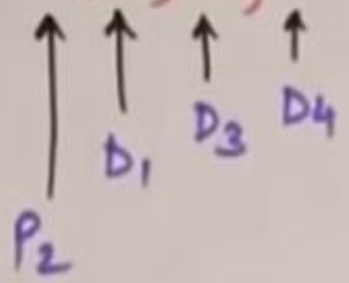
P_1 D_1 D_2 D_4

Step3: Decide P_2

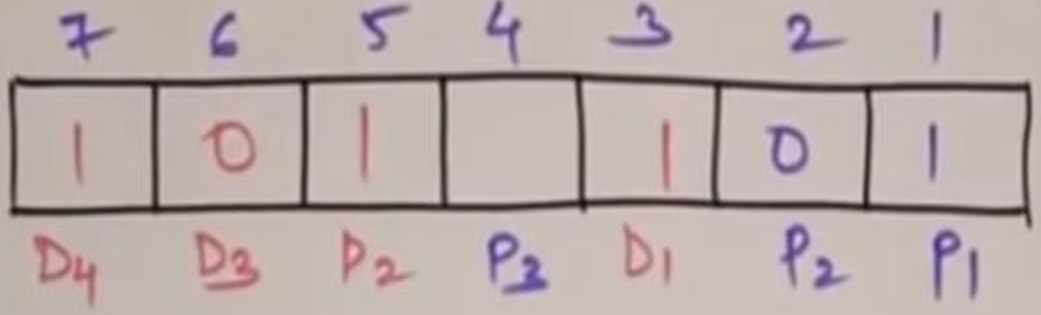


(Even Parity)

Considered bits: 2, 3, 6, 7 → , 1, 0, 1

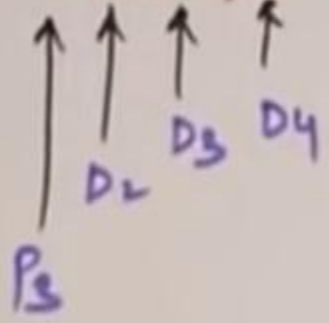


Step4: Decide P_3

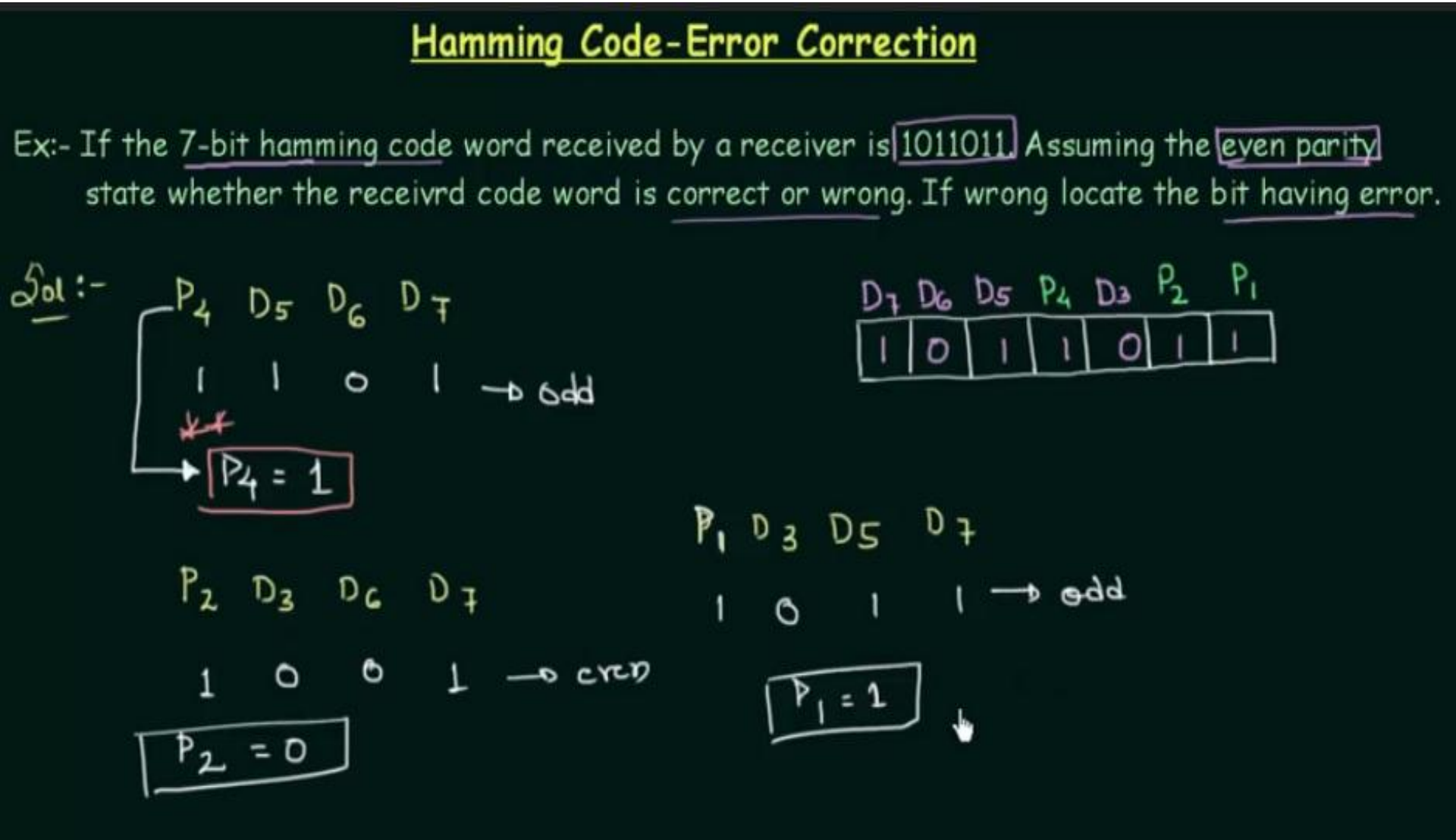


(Even Parity)

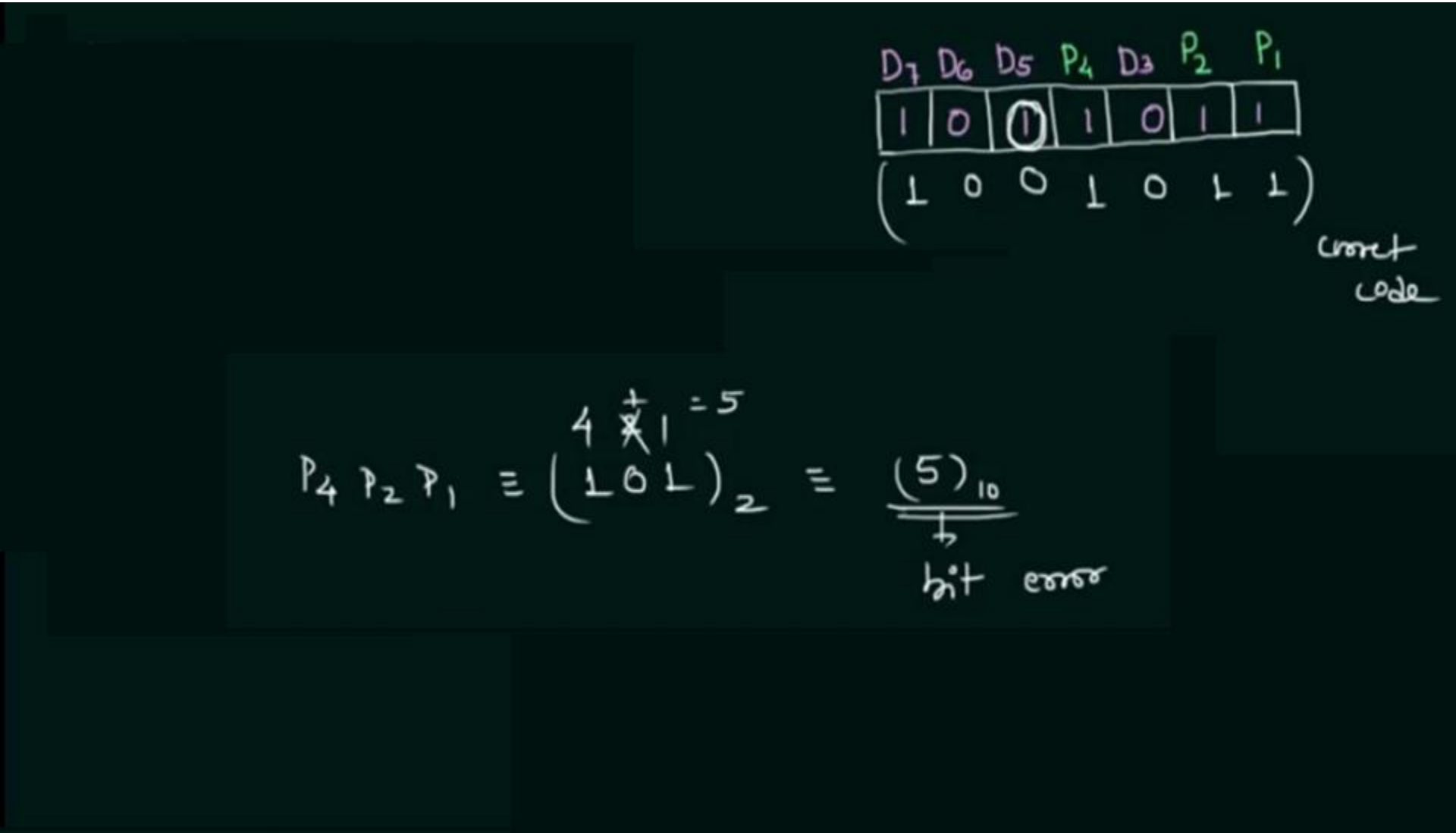
considered bits : 4, 5, 6, 7 → , 1, 0, 1



Example (Error Detection)



Example (Error Correction)



Thanks