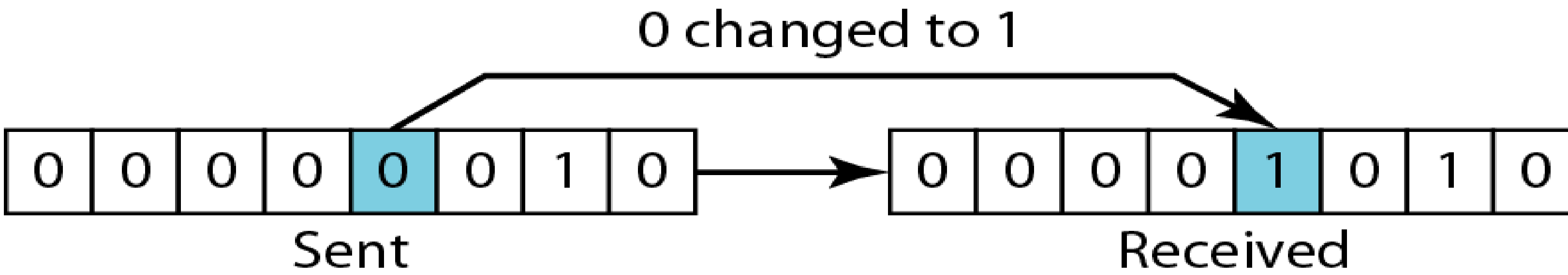
# Error Detection and Error Correction

# Introduction

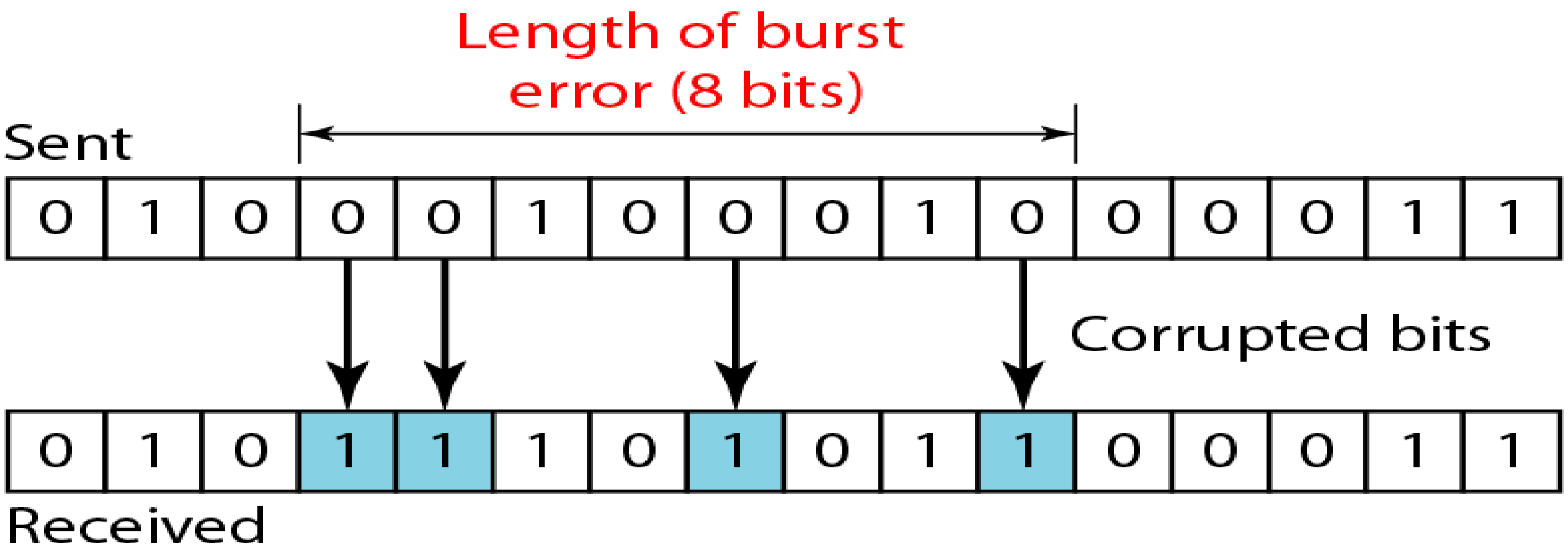
- Data can be corrupted during transmission.
- Applications require that errors be detected and corrected.
- Errors are introduced into frames Because of electrical interference and thermal noises

# Type of Errors

- Single Bit Error
  - In a single-bit error, only 1 bit in the data unit has changed.
- Burst Error
  - A burst error means that 2 or more bits in the data unit have changed
- To detect or correct errors, an extra (redundant) bits with data has to be sent.




*Single-bit error*




*Burst error*

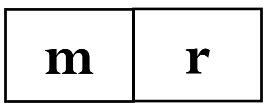
# Basic Idea of Error Detection

- To add redundant information to a frame that can be used to determine if errors have been introduced
- Imagine (Extreme Case)
  - Transmitting two complete copies of data
    - Identical → No error
    - Differ → Error
    - Poor Scheme ???
      - n bit message, n bit redundant information
      - Error can go undetected
  - In general, we can provide strong error detection technique
    - k redundant bits, n bits message, k << n
    - In Ethernet, a frame carrying up to 12,000 bits of data requires only 32-bit CRC

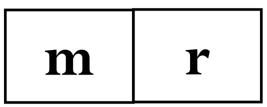
# Basic Idea of Error Detection

- Extra bits are redundant
  - They add no new information to the message
  - Derived from the original message using some algorithm
  - Both the sender and receiver know the algorithm

Sender




Receiver




Receiver computes *r* using *m*

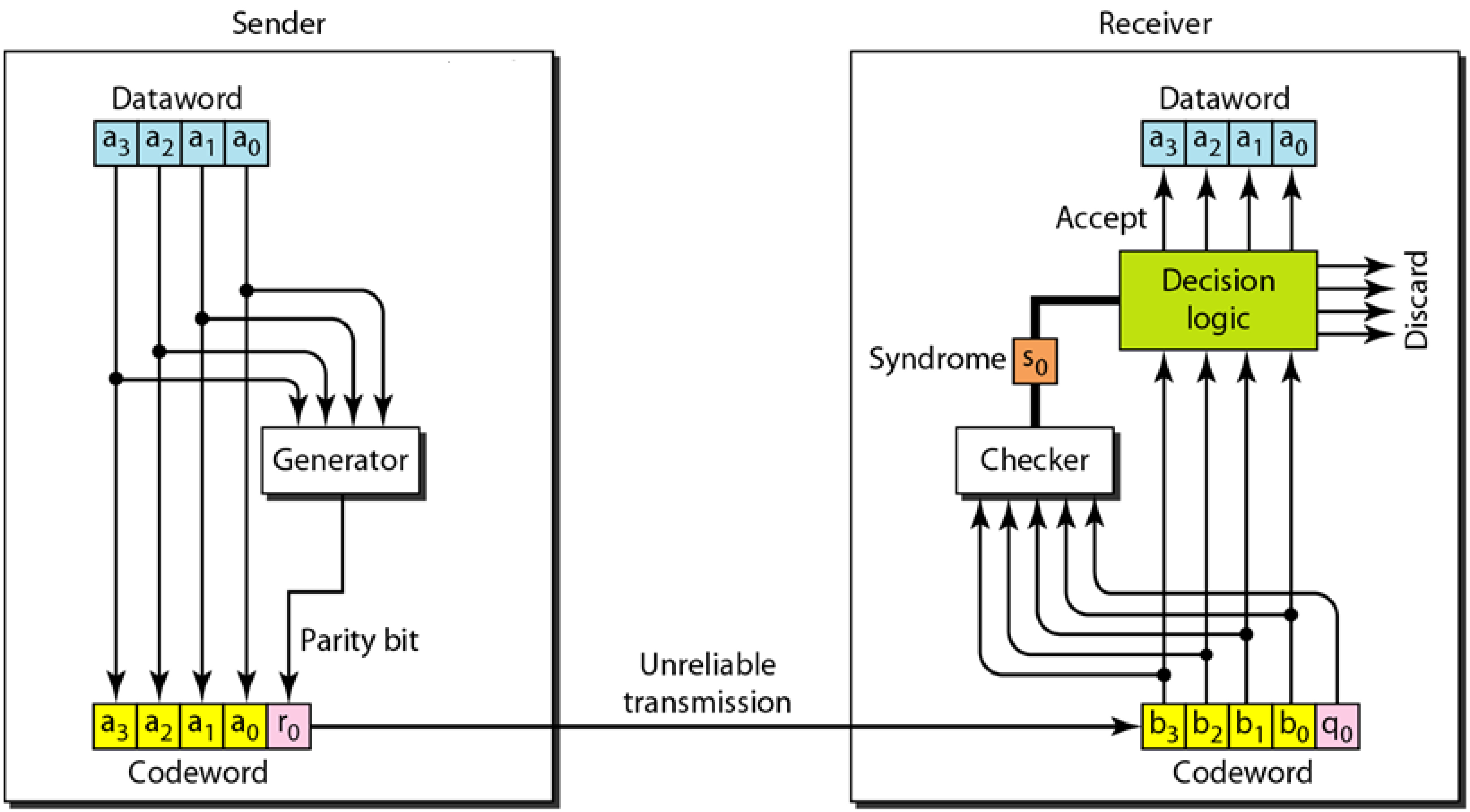
If they match, no error

# Error Detection Techniques

- Simple Parity Check
- 2D Parity Check
  - Used in BISYNC
- Cyclic Redundancy Check
  - Used in HDLC, DDCMP, CSMA/CD, Token Ring
- Checksum
  - Used in IP

- Parity
  - Even Parity
  - Odd Parity

# Simple Parity Check



# Example

- Data → 1011.
- Codeword → 10111   (Even Parity)

1. No error occurs; Codeword → 10111.
2. One single-bit error. Codeword → 10011.

   No dataword is created.
3. One single-bit error codeword →10110.

   No dataword is created

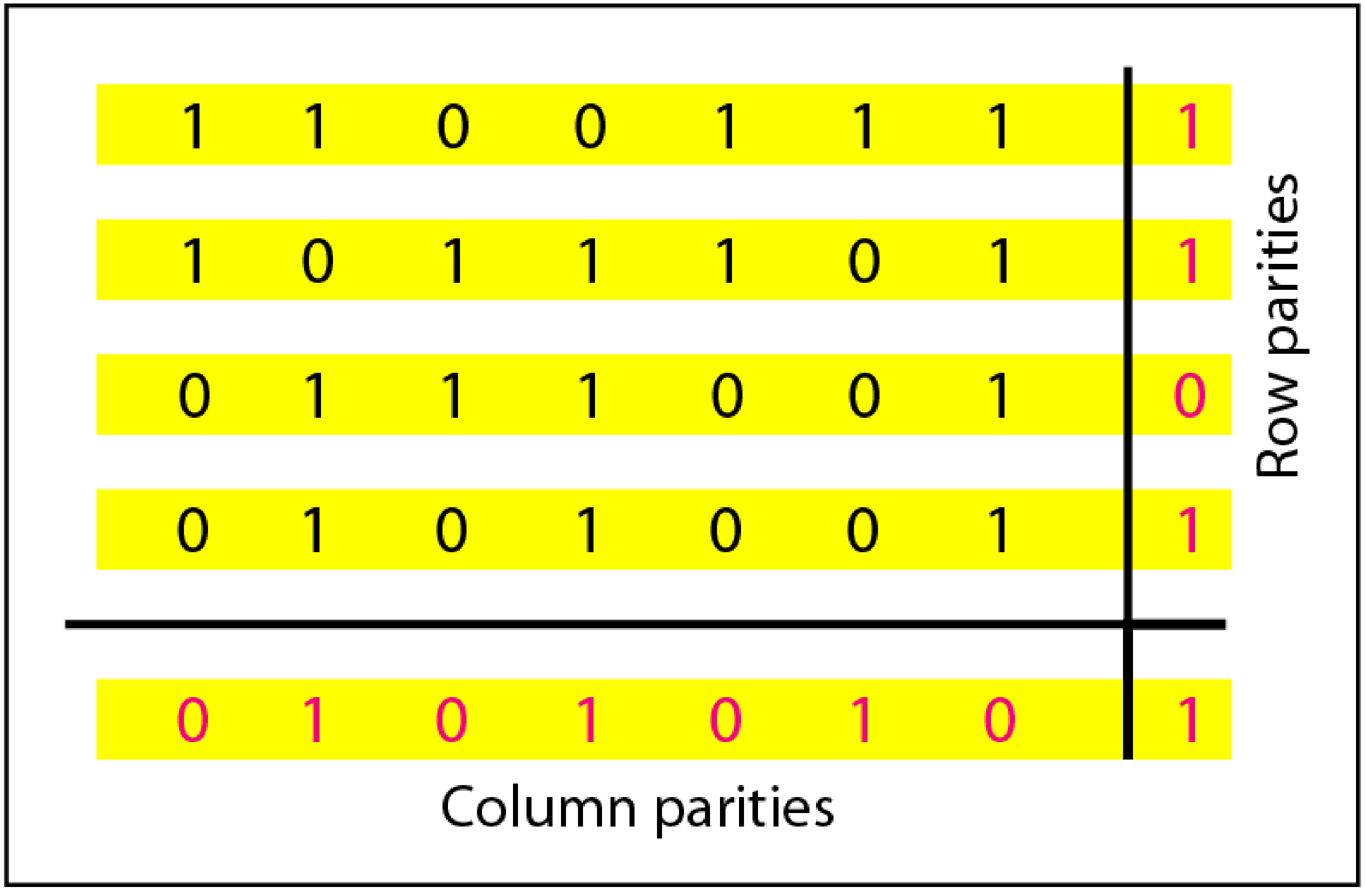
# Example

*4*. 2 Bit Errors codeword → 00110.
dataword → 0011 wrong

5. Three bits Errors codeword → 01011.
No dataword is created

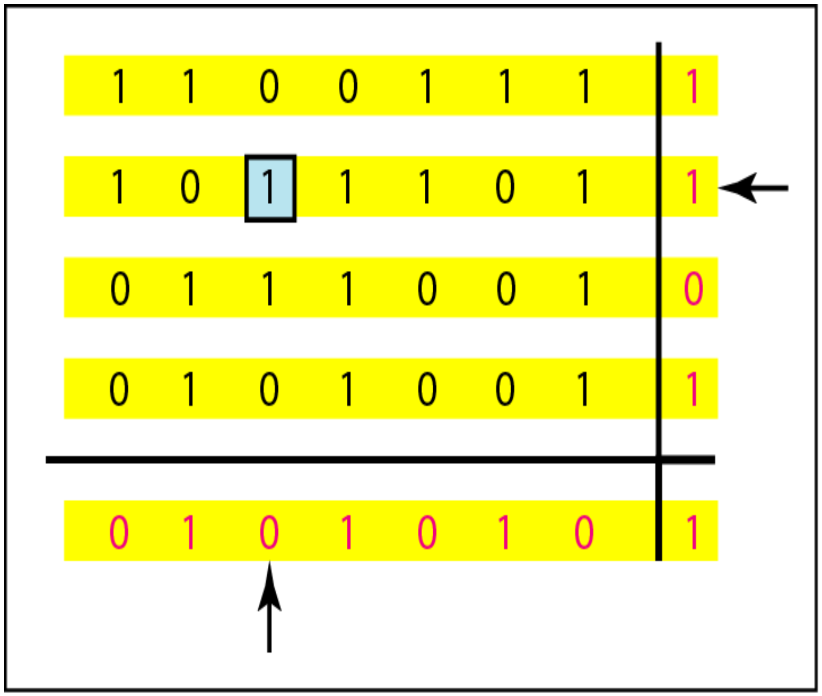
- A simple parity-check code can detect an odd number of errors.

# Two-dimensional parity-check

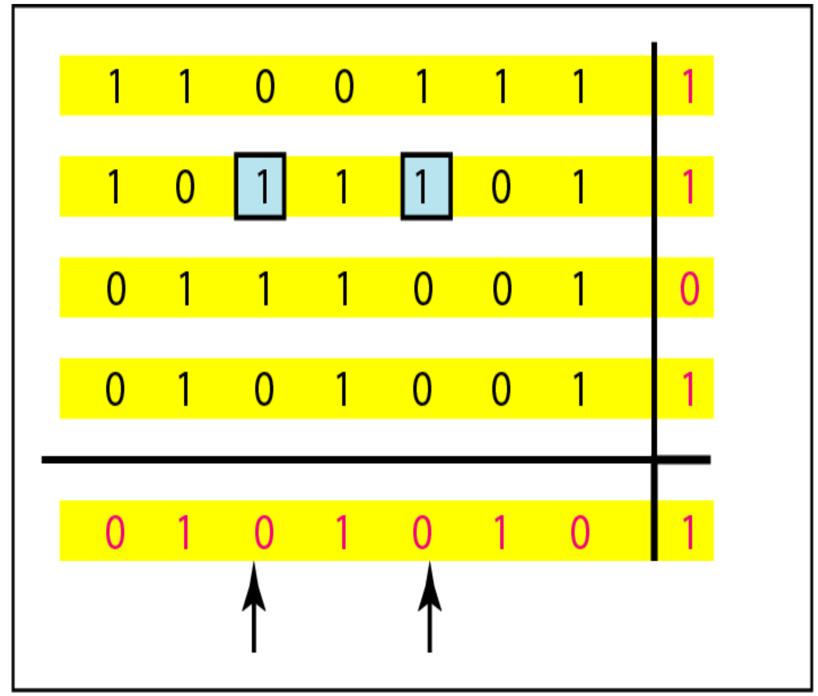



a. Design of row and column parities

# Two-dimensional parity-check

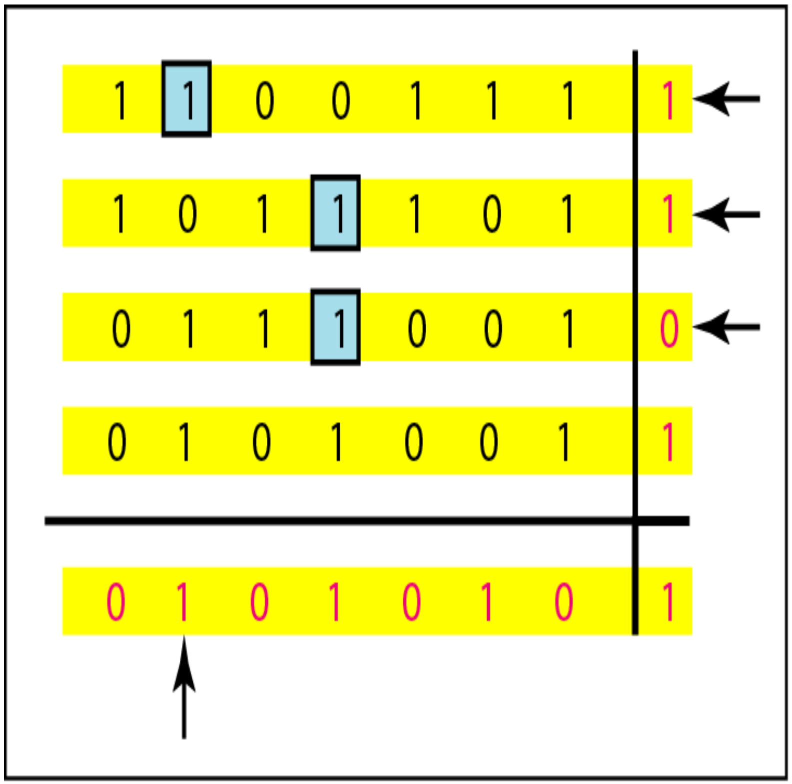



b. One error affects two parities

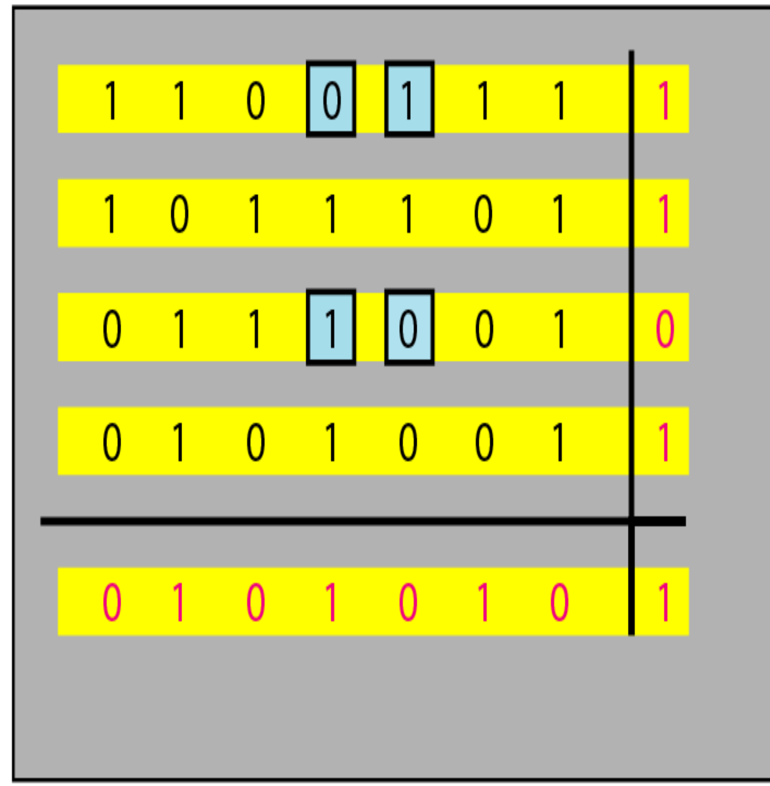



c. Two errors affect two parities

# Two-dimensional parity-check

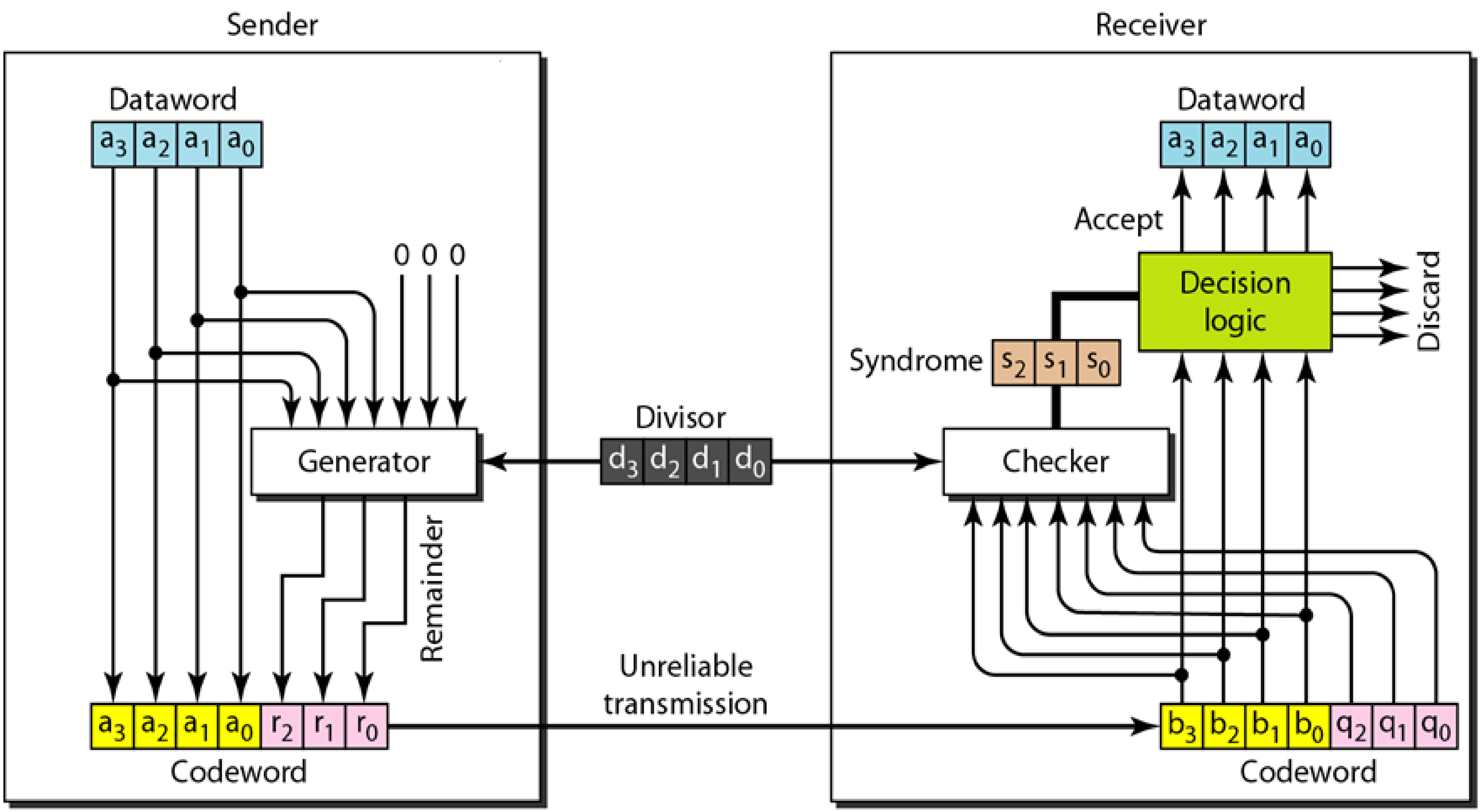



d. Three errors affect four parities

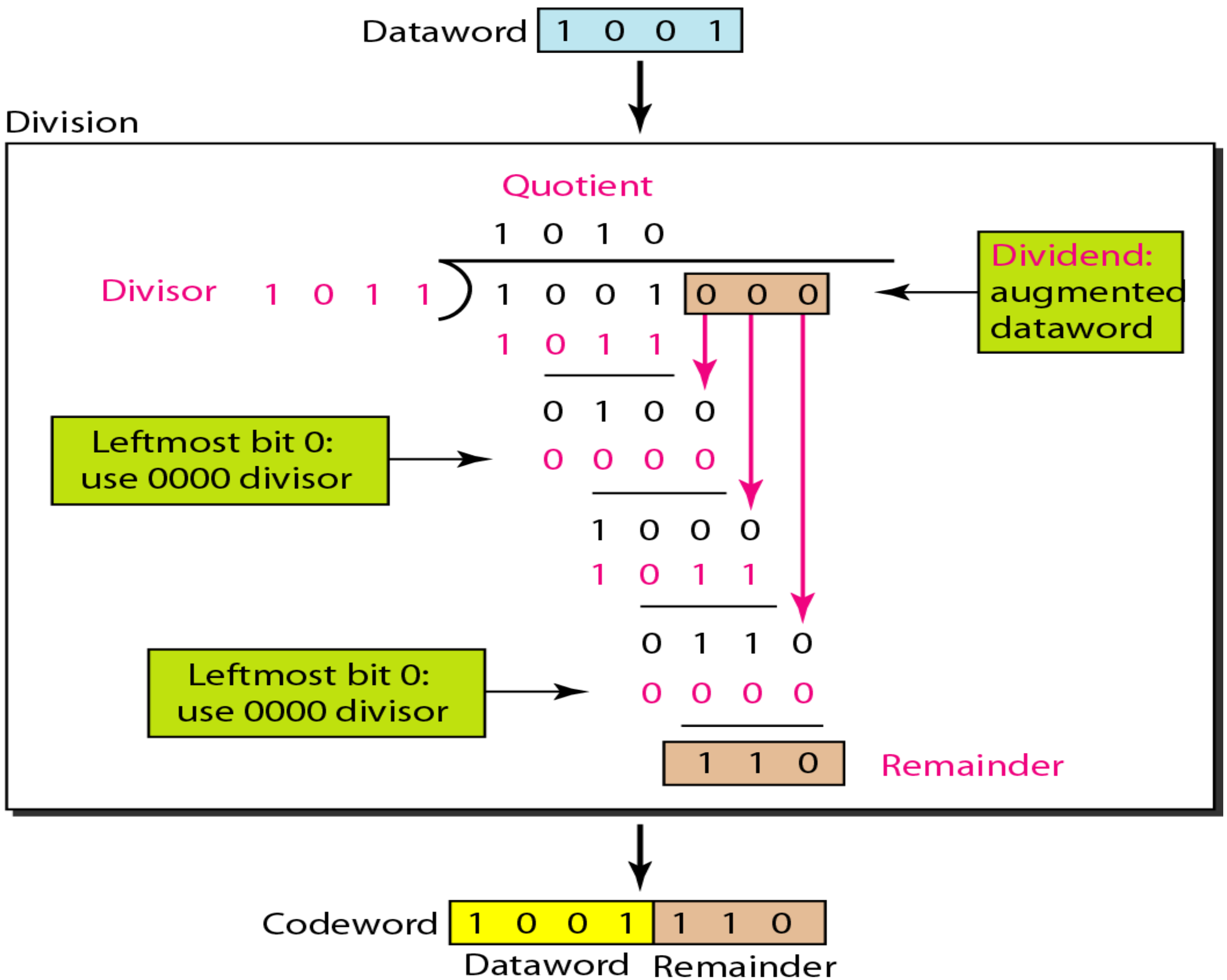



e. Four errors cannot be detected

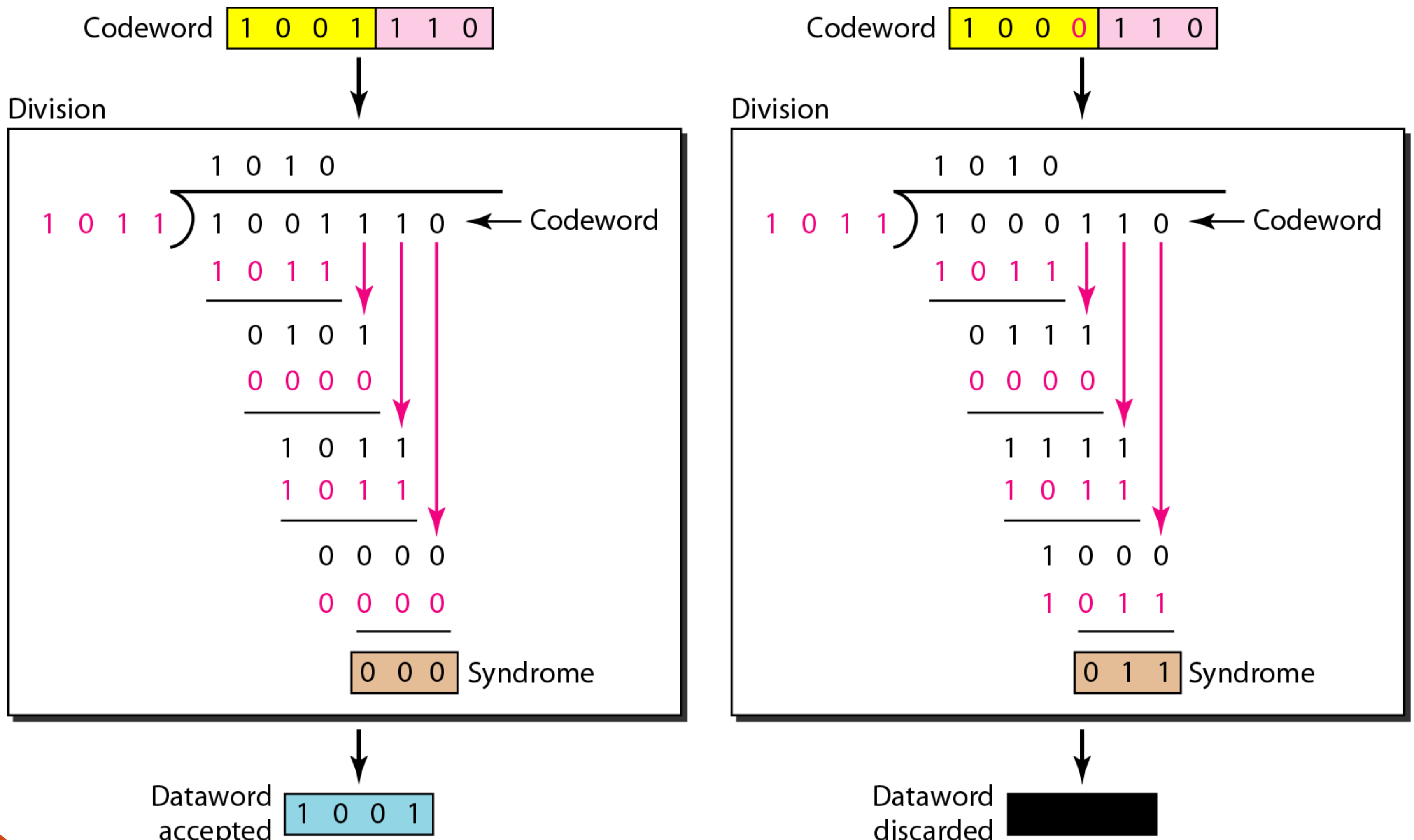
# Cyclic Redundancy Check



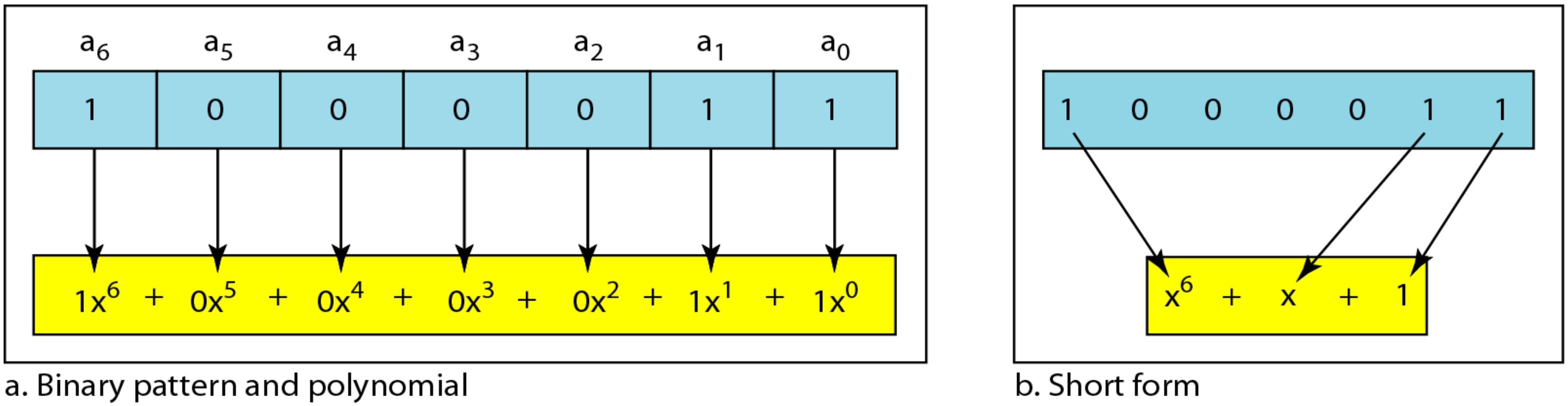
# Division in CRC sender



# Division in CRC Receiver



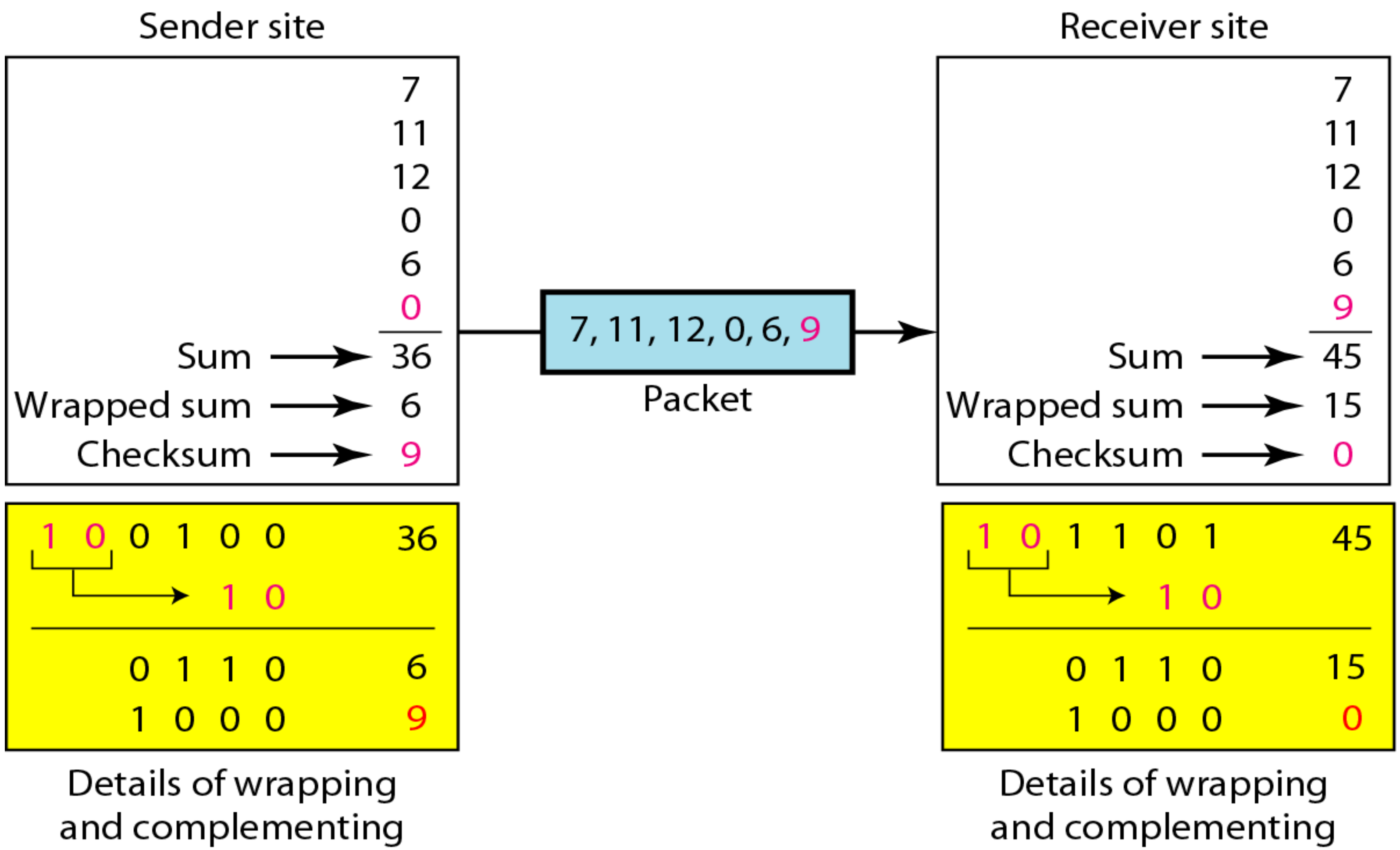
# Polynomial Representation of a binary word



# Standard Polynomials

| *Name* | *Polynomial* | *Application* |
|---|---|---|
| CRC-8 | $x^8 + x^2 + x + 1$ | ATM header |
| CRC-10 | $x^{10} + x^9 + x^5 + x^4 + x^2 + 1$ | ATM AAL |
| CRC-16 | $x^{16} + x^{12} + x^5 + 1$ | HDLC |
| CRC-32 | $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ | LANs |

# Checksum


Sender site
7
11
12
0
6
0
Sum
36
Wrapped sum
6
Checksum
9
1 0 0 1 0 0
36
1 0
0 1 1 0
6
1 0 0 0
9
Details of wrapping and complementing
7, 11, 12, 0, 6, 9
Packet
Receiver site
7
11
12
0
6
9
Sum
45
Wrapped sum
15
Checksum
0
1 0 1 1 0 1
45
1 0
0 1 1 0
15
1 0 0 0
0
Details of wrapping and complementing