Data Communication (CSX-208)

Dr Samayveer Singh

Data Link Layer

Error Detection and Correction

Error Detection and Correction

- Networks must be able to transfer data from one device to another with complete accuracy.
- Data can be corrupted during transmission.
- For reliable communication, errors must be detected and corrected.
- Error detection and correction are implemented either at the data link layer or the transport layer of the OSI model.
- Types of error: Single-bit error and burst Error

Single-bit error

- In a single-bit error, only 1 bit in the data unit has changed.

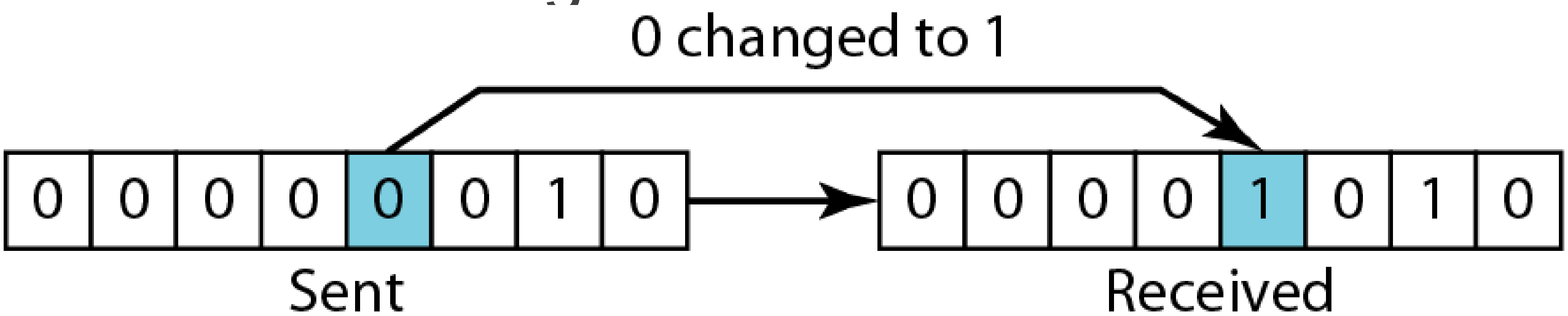


Figure: Single –bit error

Burst error

- A burst error means that 2 or more bits in the data unit have changed.

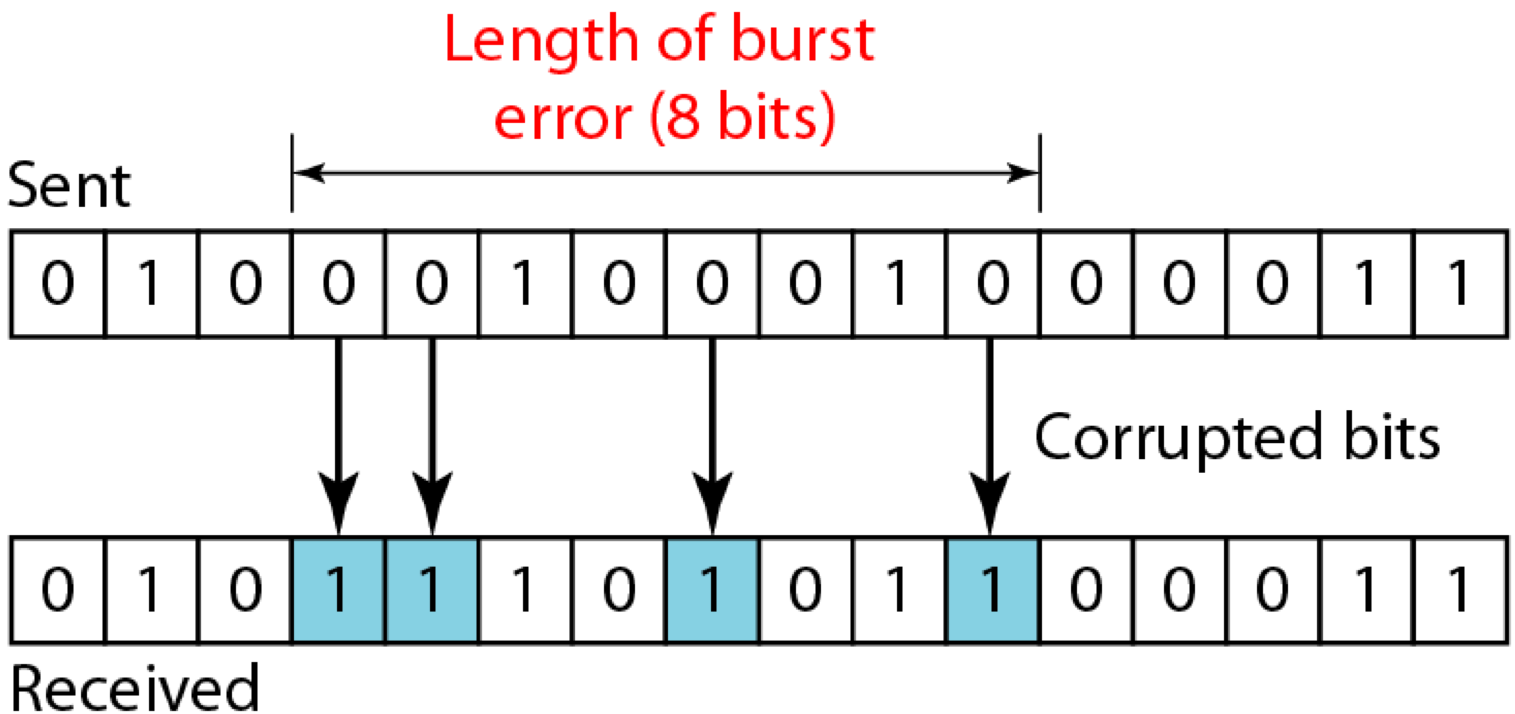


Figure: Burst error of length 8

Example:

- ★ The number of bits affected depends on the data rate and duration of noise.

- ➔ If data is sent at rate = 1Kbps then a noise of 1/100 sec can affect 10 bits.(1/100*1000)

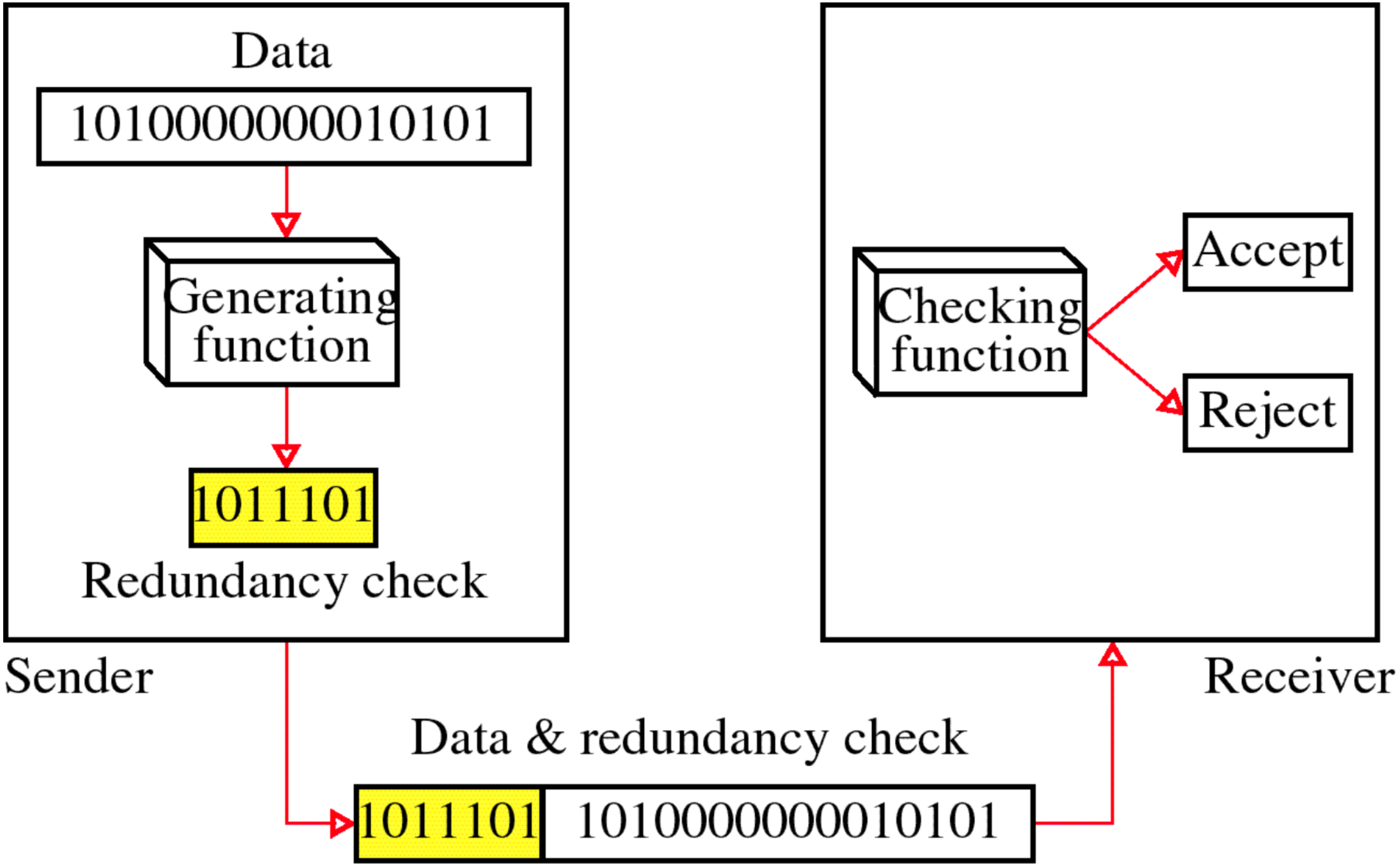
- ➔ If same data is sent at rate = 1Mbps then a noise of 1/100 sec can affect 10,000 bits.($1/100*10^6$)

Error detection

- Error detection means to decide whether the received data is correct or not without having a copy of the original message.

- Error detection uses the concept of redundancy, which means adding extra bits for detecting errors at the destination.

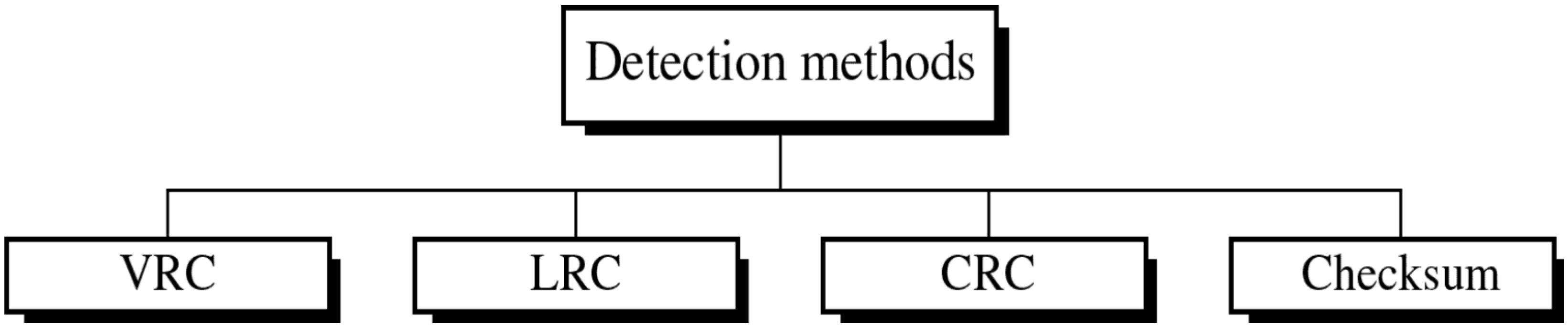
Redundancy



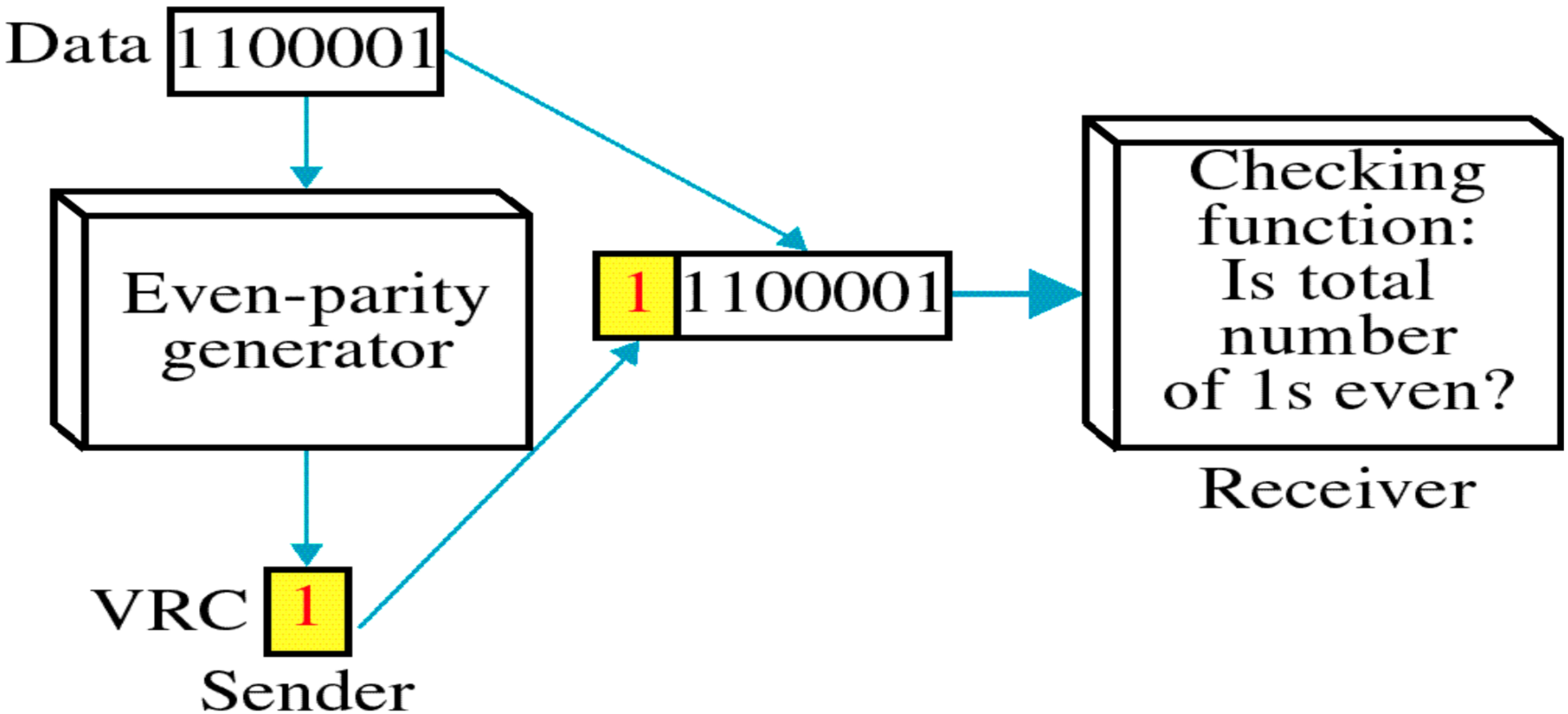
Data
1010000000010101
Generating function
1011101
Redundancy check
Sender
Checking function
Accept
Reject
Receiver
Data & redundancy check
1011101
1010000000010101

Detection methods

- Four types of redundancy checks are used in data communications

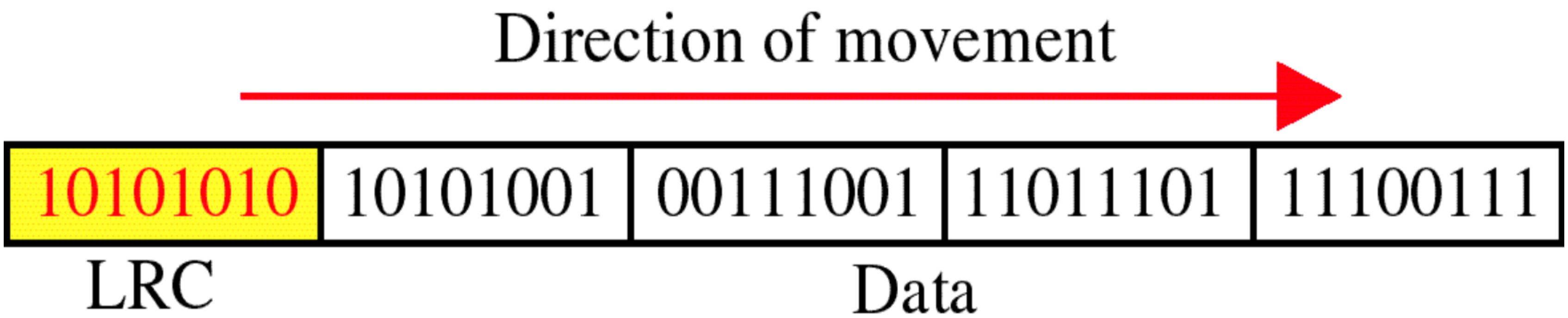


Vertical Redundancy Check (VRC)



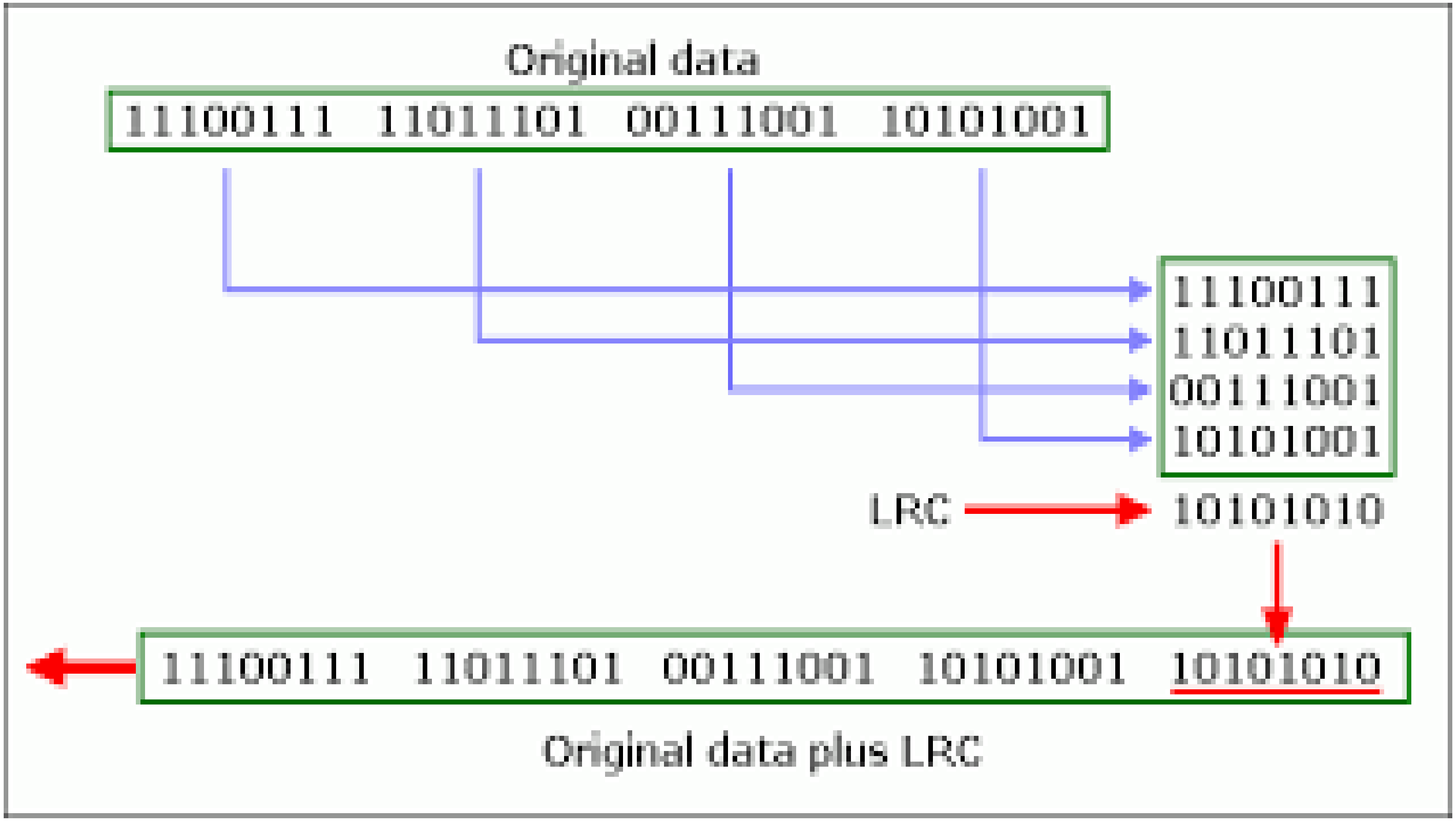
- It can detect single bit error
- It can detect burst errors only if the total number of errors is odd.

Longitudinal Redundancy Check (LRC)

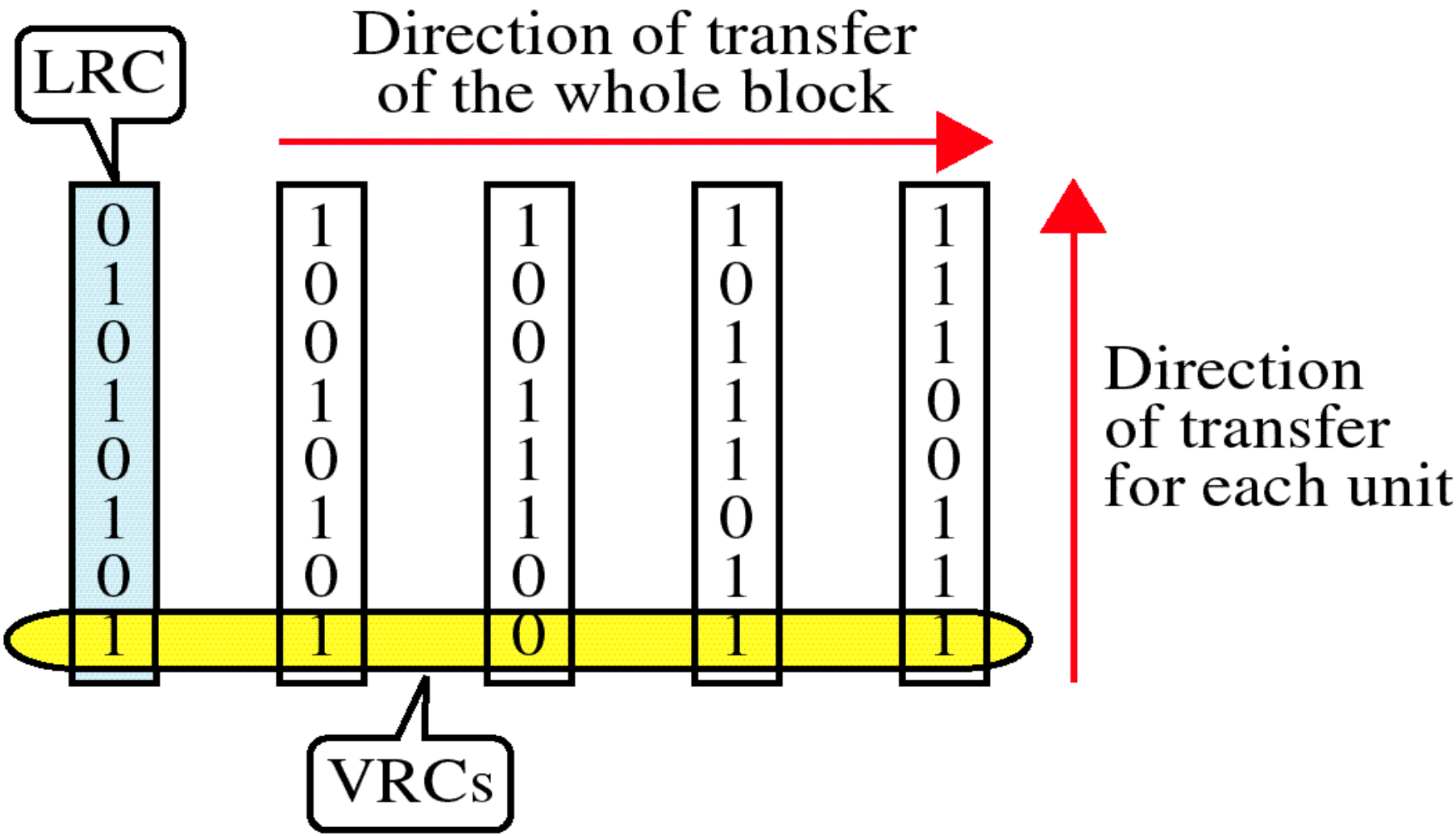


- If two bits in one data units are damaged and two bits in exactly the same positions in another data unit are also damaged, the LRC checker will not detect an error.
- Eg. Two data units are 11110000 and 11000011. If the first and last bits in each of them is changed. The data units are **0**111000**1** and **0**100001**0** and error cant be detected.

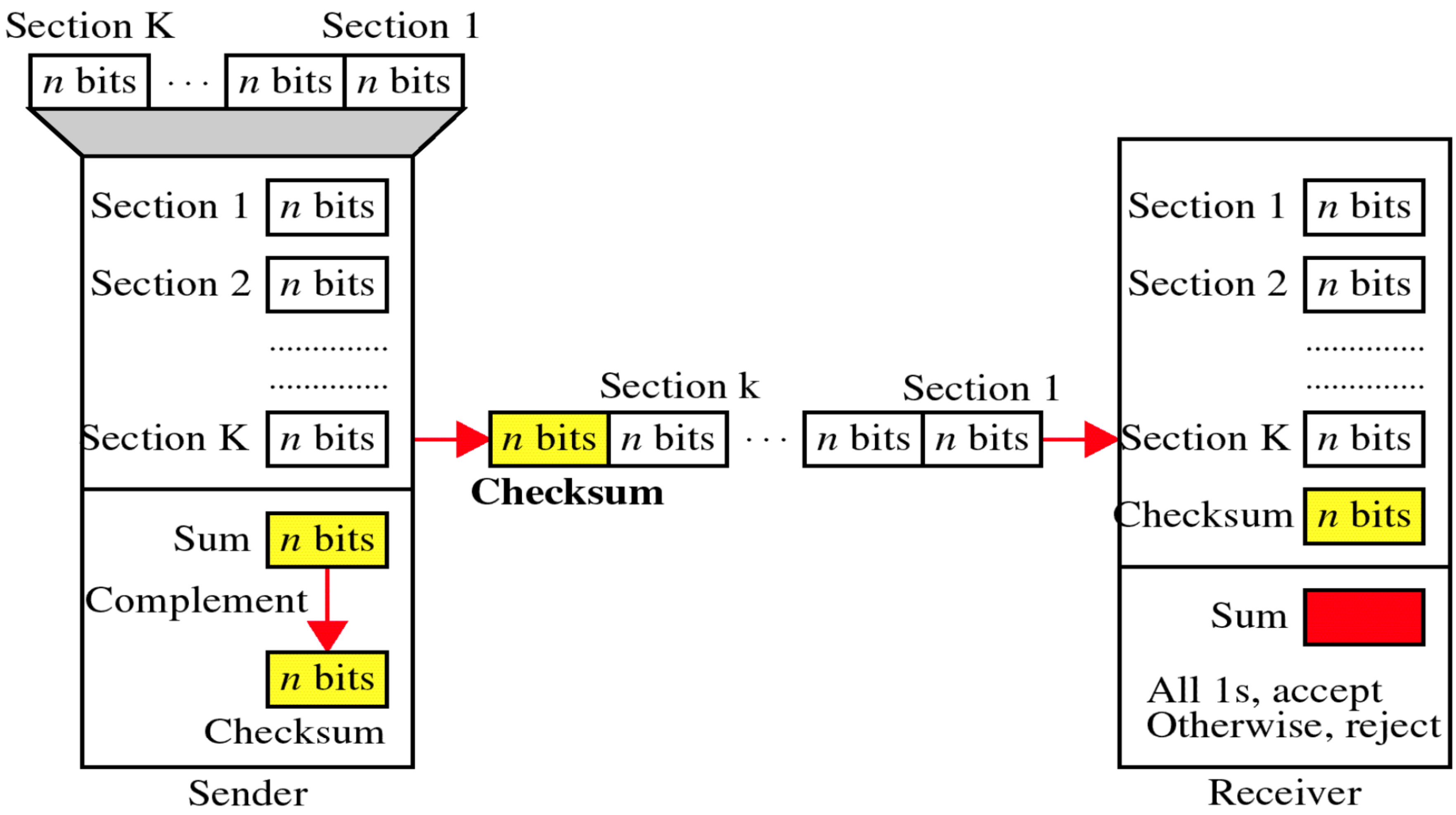
LRC


Original data
11100111 11011101 00111001 10101001
11100111
11011101
00111001
10101001
LRC
10101010
11100111 11011101 00111001 10101001 10101010
Original data plus LRC

VRC and LRC


Direction of transfer
of the whole block
LRC
0 1 0 1 0 1 0 1
1 0 0 1 0 1 0 1
1 0 0 1 1 1 0 0
1 0 1 1 1 0 1 1
1 1 1 0 0 1 1 1
Direction
of transfer
for each unit
VRCs

Checksum



Section K
Section 1
n bits
n bits
n bits
Section 1
n bits
Section 2
n bits
Section K
n bits
Sum
n bits
Complement
n bits
Checksum
Sender
Section k
Section 1
n bits
n bits
n bits
n bits
Checksum
Section 1
n bits
Section 2
n bits
Section K
n bits
Checksum
n bits
Sum
All 1s, accept
Otherwise, reject
Receiver

At the sender

- The unit is divided into k sections, each of n bits.
- All sections are added together using one’s complement to get the sum.
- The sum is complemented and becomes the checksum.
- The checksum is sent with the data

At the receiver

- The unit is divided into k sections, each of n bits.
- All sections are added together using one’s complement to get the sum.
- The sum is complemented.
- If the result is zero, the data are accepted: otherwise, they are rejected.

Example Part 1

- Suppose the following block of 16 bits is to be sent using a checksum of 8 bits.
- 10101001 00111001
- The numbers are added using one's complement
- 10101001
- 00111001

Sum 11100010

- Checksum **00011101**
- The pattern sent is 10101001 00111001 **00011101**

Example part 2

- Now suppose the receiver receives the pattern sent in Example *1* and there is no error.
- 10101001 00111001 00011101
- When the receiver adds the three sections, it will get all 1s, which, after complementing, is all 0s and shows that there is no error.
- 10101001
- 00111001
- 00011101
- Sum 11111111
- Complement **00000000** means that the pattern is OK.

Example 3

Now suppose there is a burst error of length five that affects four bits.

10101*111* *11*111001   00011101

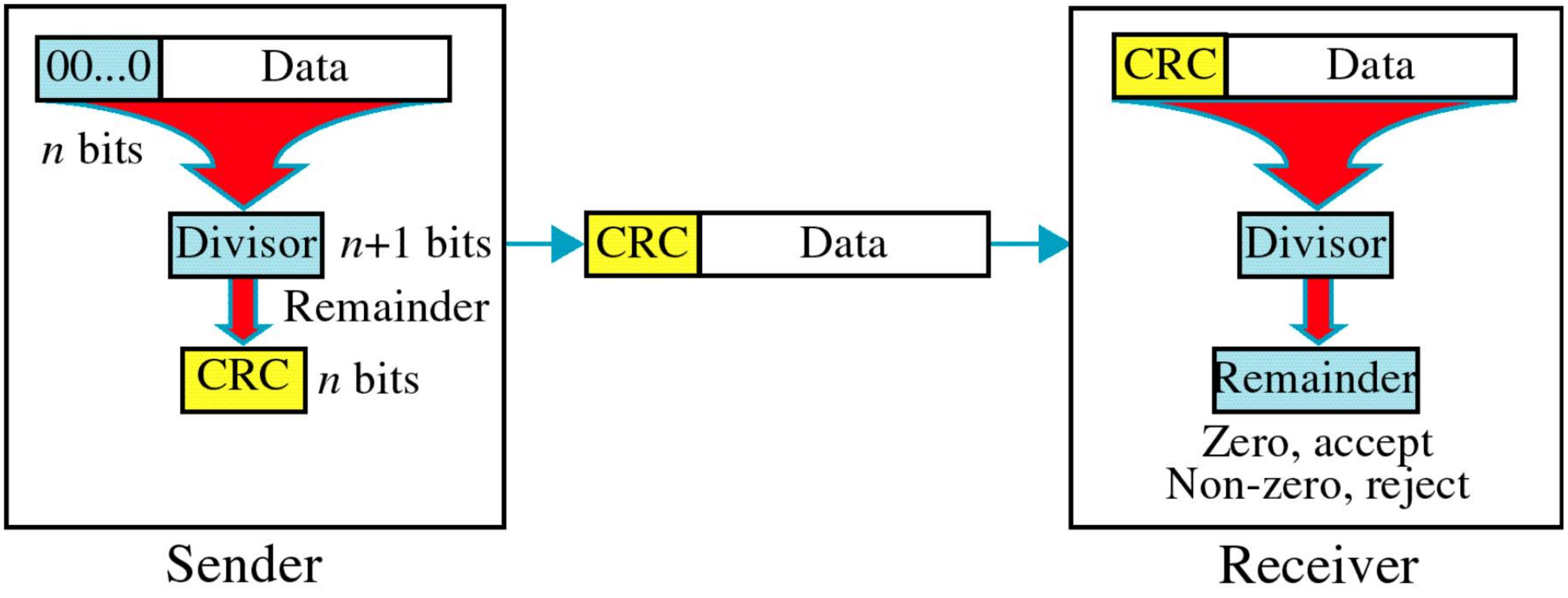
When the receiver adds the three sections together, it gets

		10101111
		11111001
		00011101

Result	1	11000101
Carry		1

Sum		11000110
Complement		00111001 means that the pattern is corrupted.

Cyclic Redundancy Check (CRC)

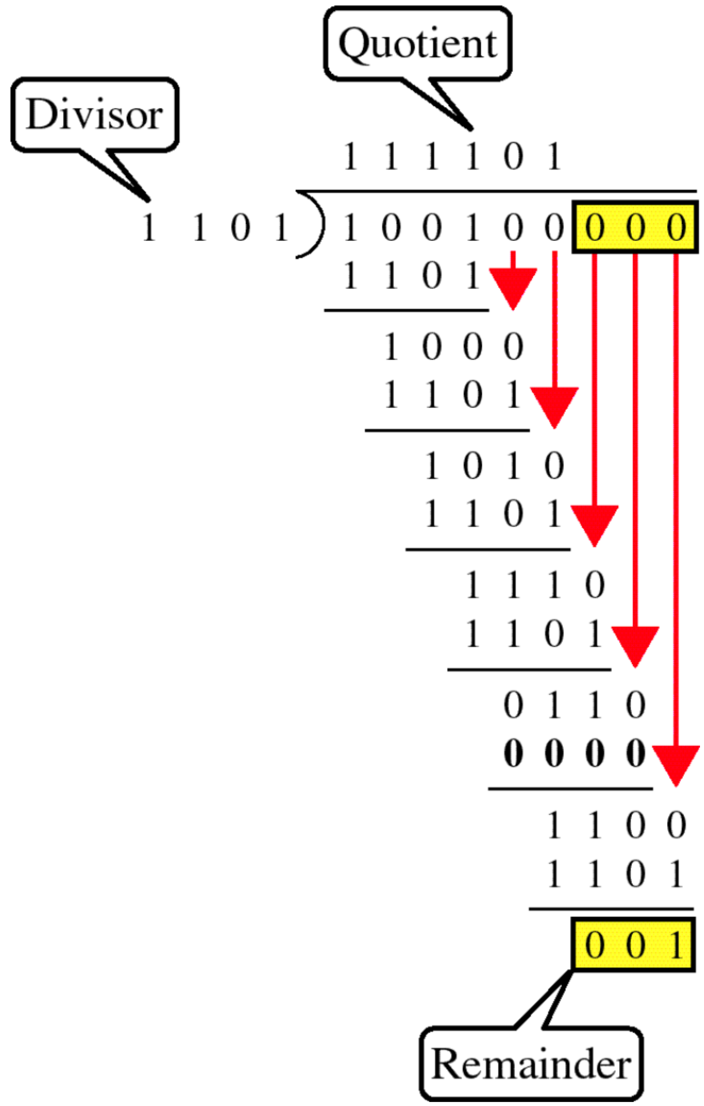


Cyclic Redundancy Check

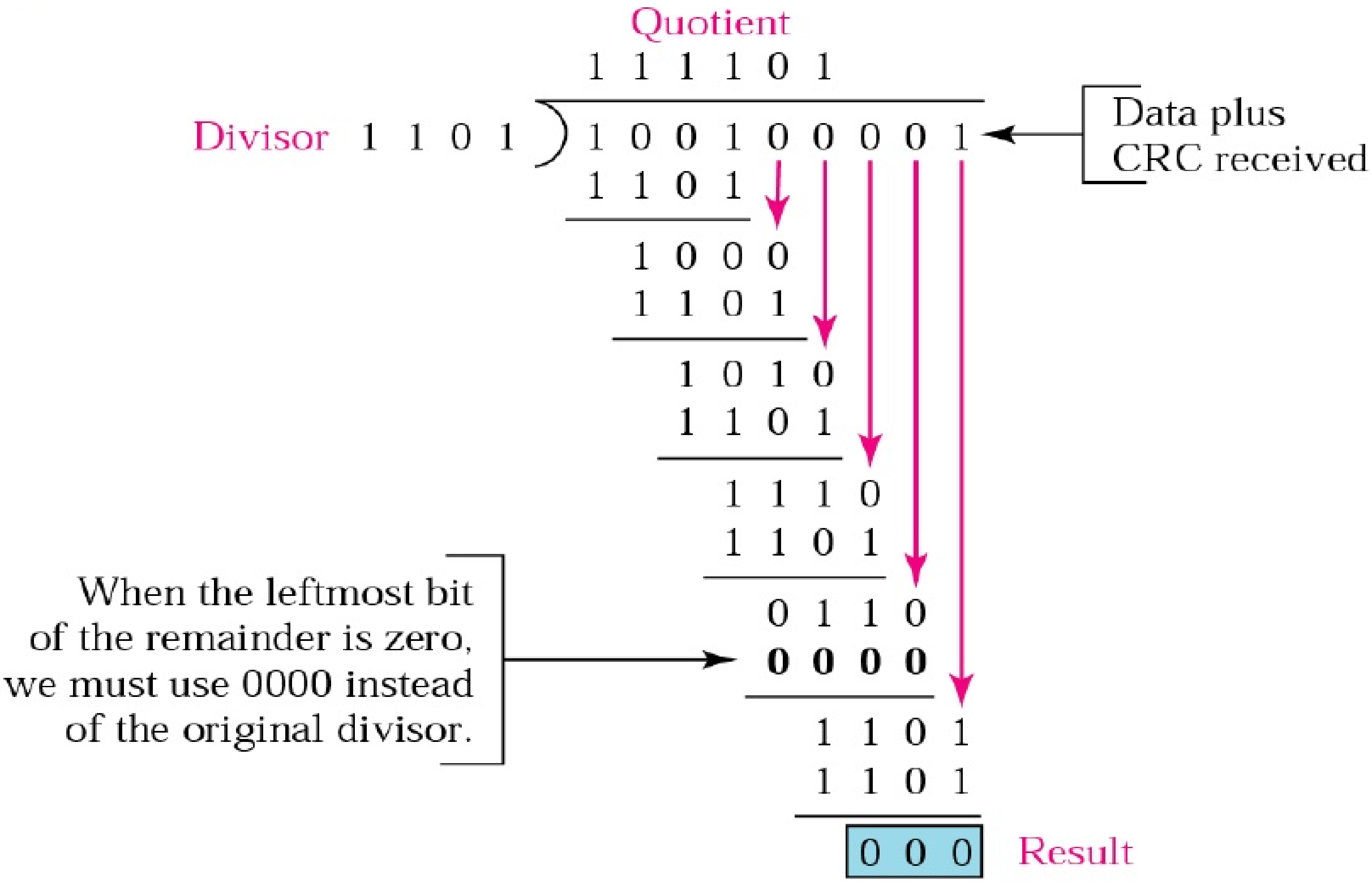
- Given a k-bit frame or message, the transmitter generates an n-bit sequence, known as a frame check sequence (FCS), so that the resulting frame, consisting of (k+n) bits, is exactly divisible by some predetermined number.

- The receiver then divides the incoming frame by the same number and, if there is no remainder, assumes that there was no error.

Binary Division



Binary Division



Error Correction

- ***It can be handled in two ways:***
  - Receiver can have the sender retransmit the entire data unit.
  - The receiver can use an error-correcting code, which automatically corrects certain errors.

- ***Single-bit error correction***
  - To correct an error, the receiver reverses the value of the altered bit. To do so, it must know which bit is in error.

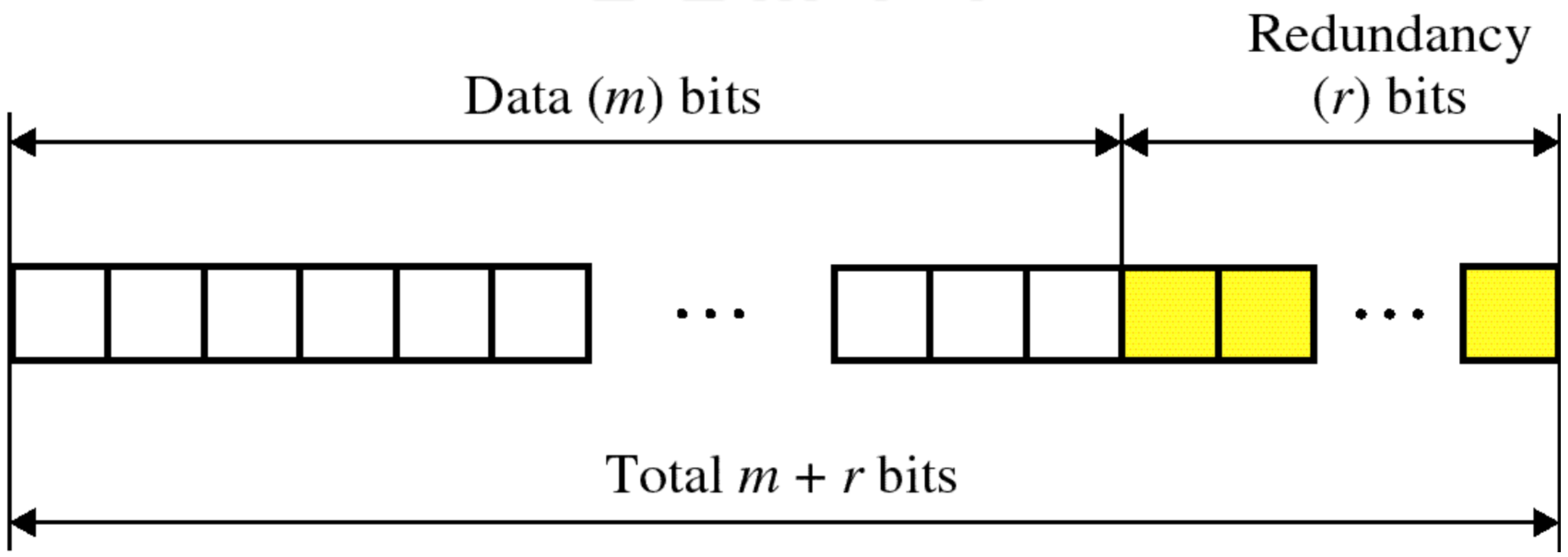
Number of redundancy bits needed

- Let data bits = m and redundancy bits = r

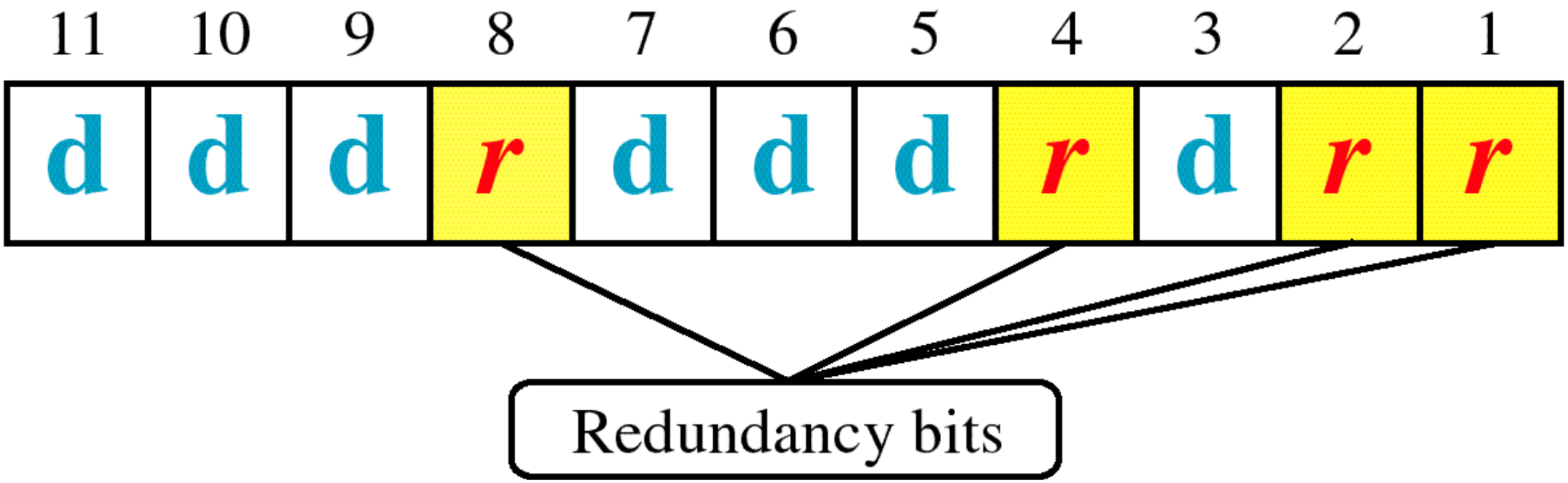
$\therefore$ Total message sent = $m+r$

The value of r must satisfy the following relation:

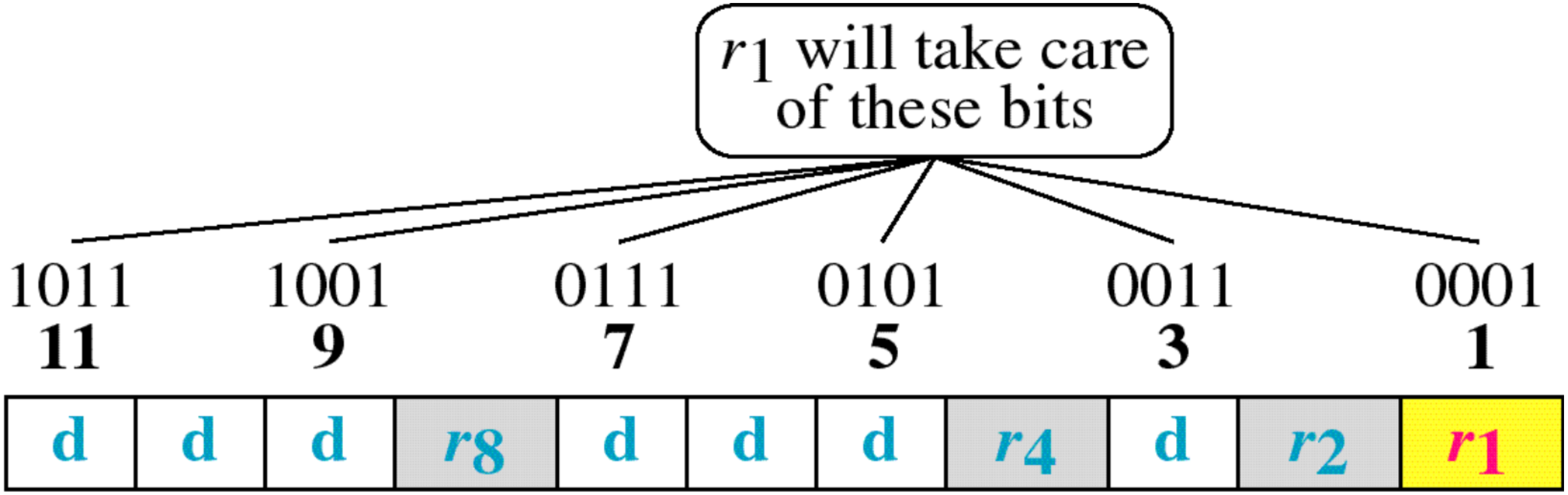
$$2^r \geq m+r+1$$

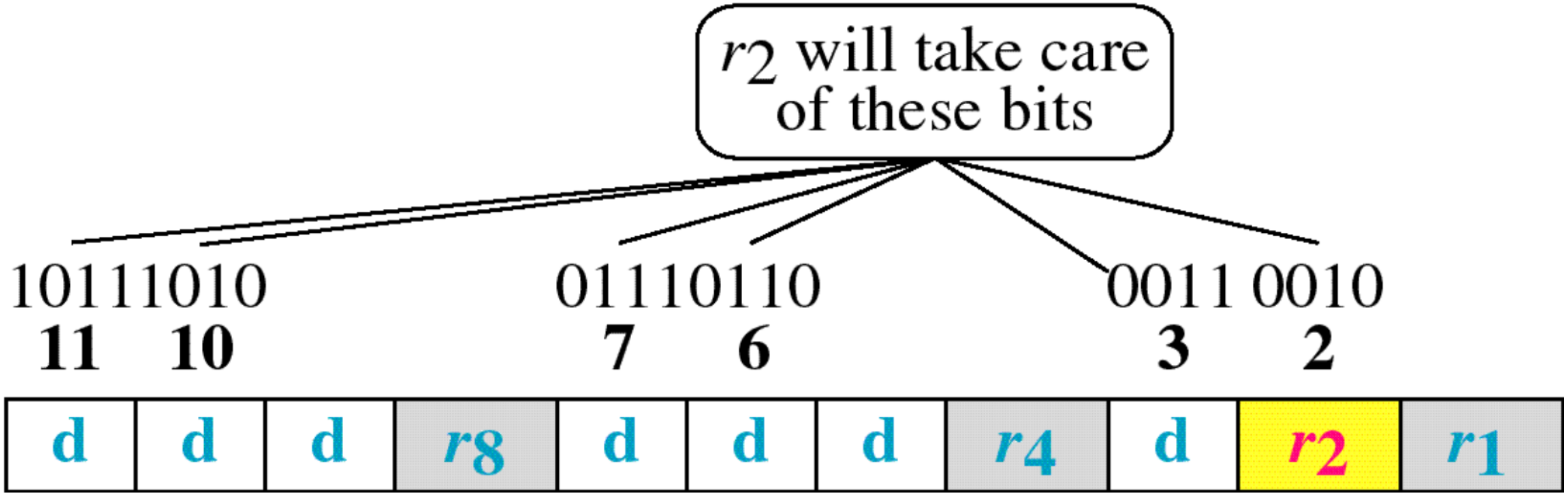


Hamming Code

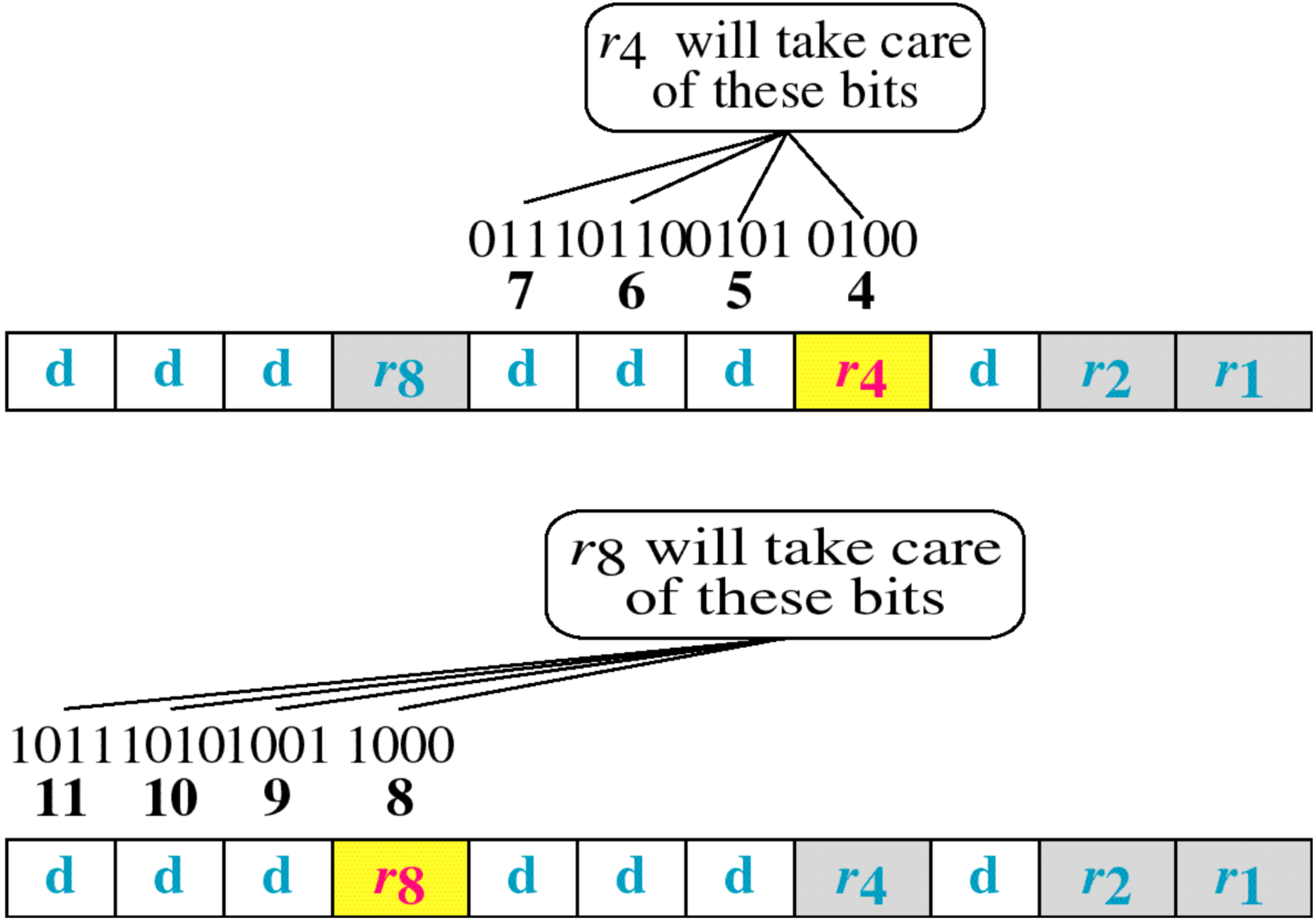


Hamming Code

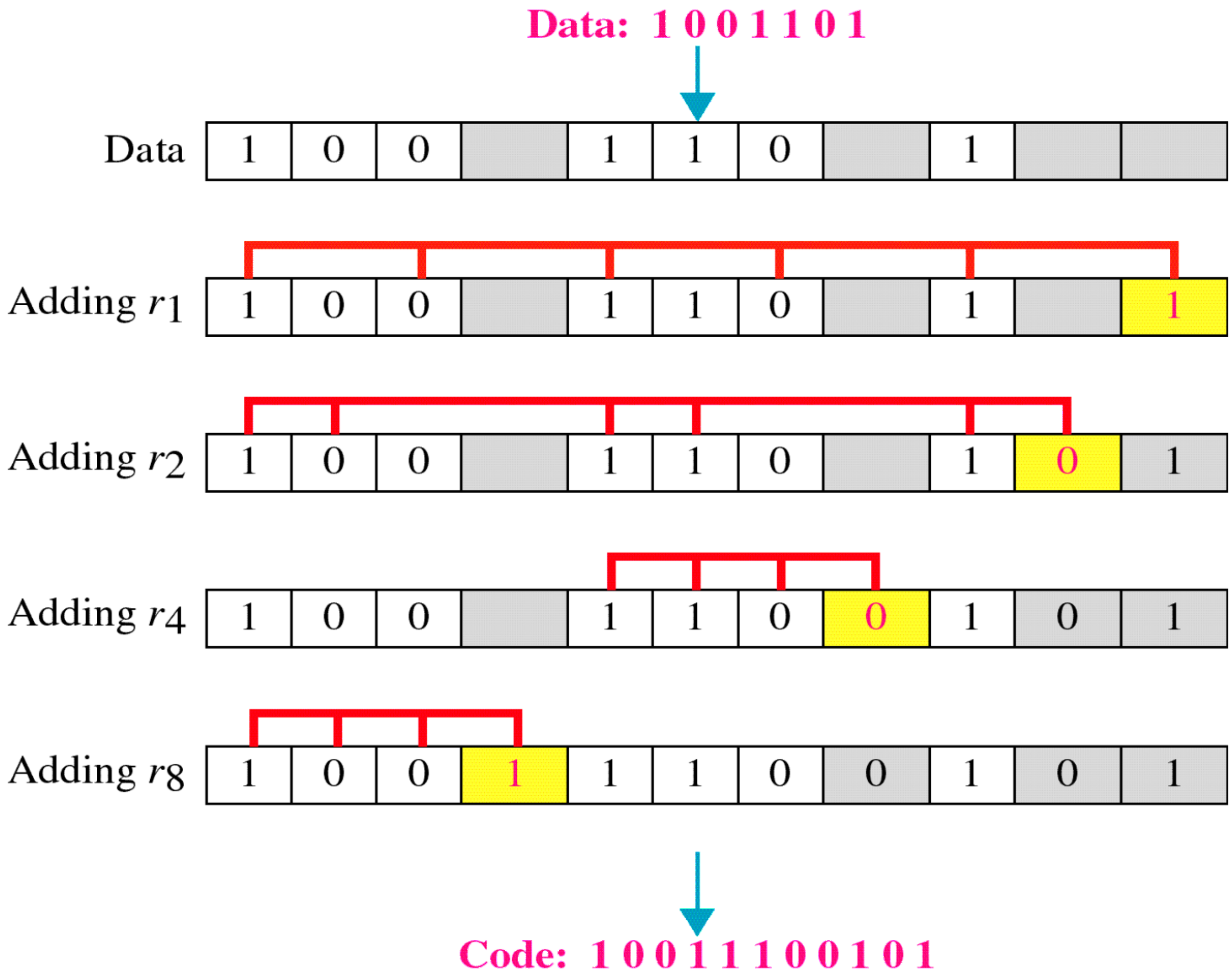




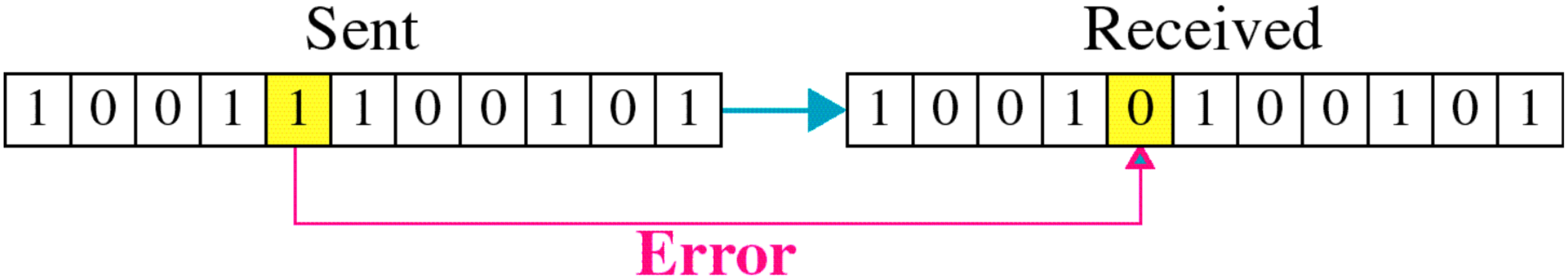
Hamming Code



Example of Hamming Code



Single-bit error



Error Detection