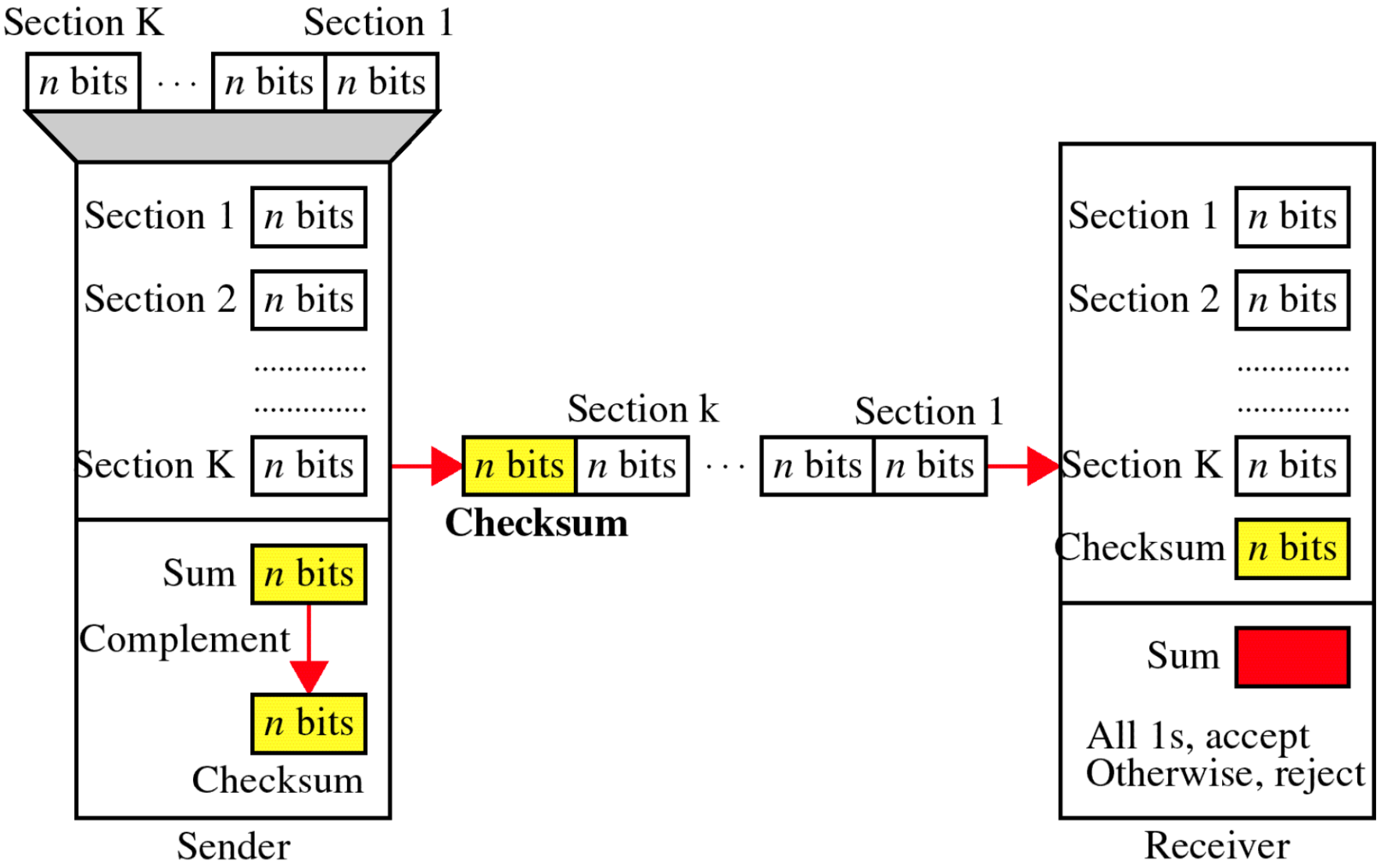
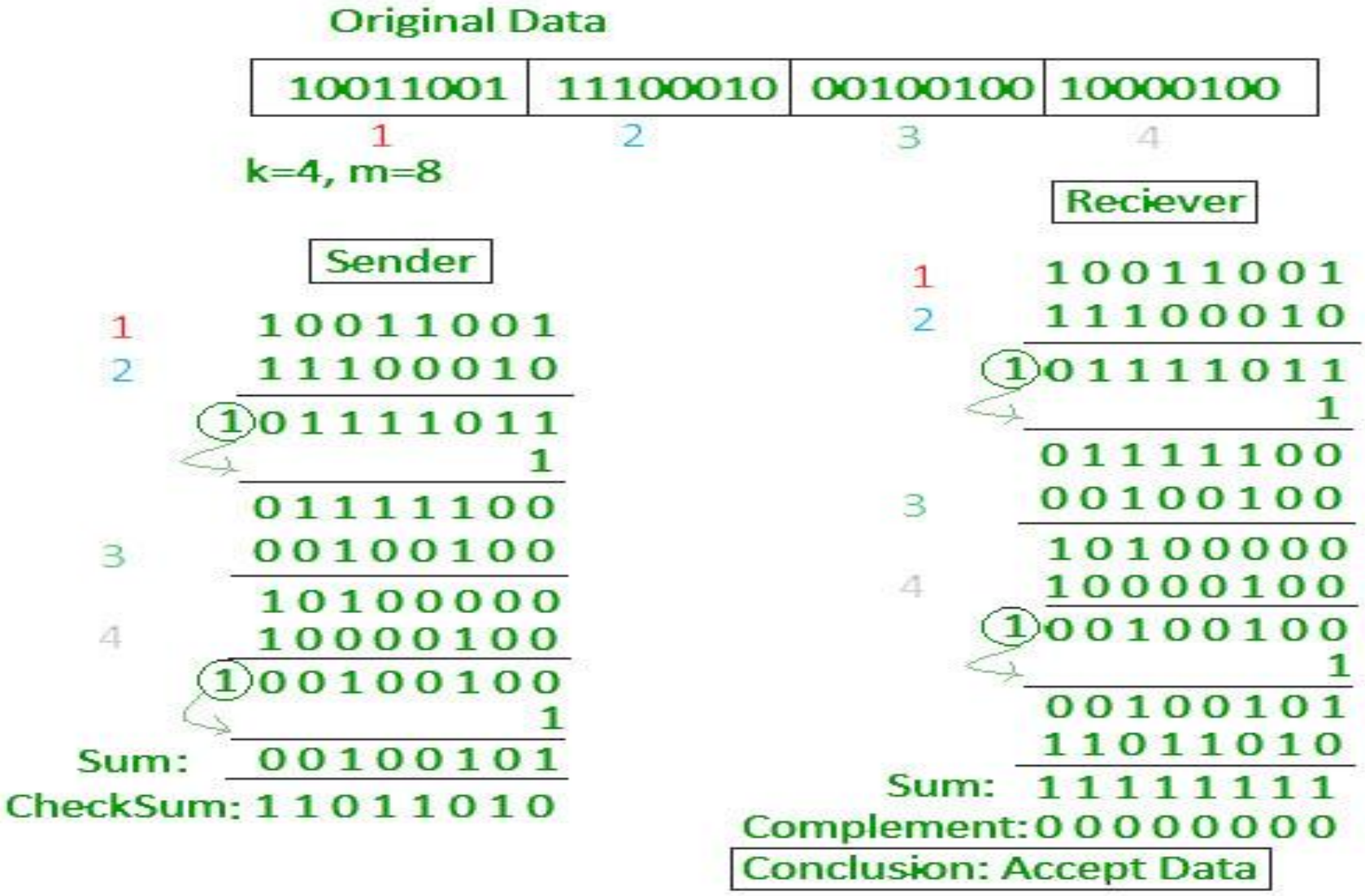
Checksum


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

Checksum


Original Data
10011001
11100010
00100100
10000100
1
2
3
4
k=4, m=8
Sender
1 10011001
2 11100010
1 01111011
1
01111100
3 00100100
10100000
4 10000100
1 00100100
1
Sum: 00100101
CheckSum: 11011010
Reciever
1 10011001
2 11100010
1 01111011
1
01111100
3 00100100
10100000
4 10000100
1 00100100
1
00100101
11011010
Sum: 11111111
Complement: 00000000
Conclusion: Accept Data

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.

Performance

- The checksum detects all errors involving an odd number of bits.
- It detects most errors involving an even number of bits.
- If one or more bits of a segment are damaged and the corresponding bit or bits of opposite value in a second segment are also damaged, the sums of those columns will not change and the receiver will not detect a problem.

Problem

- If the data unit to be transmitted is 10101001 00111001, the following procedure is used at Sender site and Receiver site.

Problem

- Sender side

```
10101001        subunit 1
00111001        subunit 2
11100010        sum (using 1s complement)
00011101        checksum (complement of sum)
```

Problem

- Receiver Side

```
10101001          subunit 1
00111001          subunit 2
00011101          checksum
11111111          sum
00000000          sum's complement

Result is zero, it means no error.
```

Problem 2

- If the data transmitted along with checksum is 10101001 00111001 00011101. But the data received at destination is **0**0101001 **1**0111001 00011101.

Problem 2

00101001	1st bit of subunit 1 is damaged
10111001	1st bit of subunit 2 is damaged
00011101	checksum
11111111	sum
00000000	Ok 1's complement

- Although data is corrupted, the error is undetected.

Error Correction

It can be handled in two ways:

1) receiver can have the sender retransmit the entire data unit.
2) The receiver can use an error-correcting code, which automatically corrects certain errors.