Module 2 – Data Link Layer

Vipin Das
Dept.of CSE
SAINTGITS College of Engineering.

Module - 2 (Data Link Layer)

Data link layer - Data link layer design issues, Error detection and correction, Sliding window protocols, High-Level Data Link Control(HDLC)protocol. Medium Access Control (MAC) sublayer –Channel allocation problem, Multiple access protocols, Ethernet, Wireless LANs - 802.11, Bridges & switches - Bridges from 802.x to 802.y, Repeaters, Hubs, Bridges, Switches, Routers and Gateways.

Data link layer design issues.

- Physical layer delivers bits of information to and from data link layer.
- **The functions of Data Link Layer are:**
- Providing a well-defined service interface to the network layer.
- Dealing with transmission errors.
    - Find erros.
    - Retransmit the data
- Regulating the flow of data so that slow receivers are not swamped by fast senders.
-

Contd..

- **Data Link layer**
- Takes the packets from Network layer, and
- Encapsulates them into frames
- Each frame has a
  - frame header – a field for holding the packet
  - frame trailer.

- Frame Management is what Data Link Layer does.

1.Services provided to the network layer

Data link layer is to transfer the data from the network layer on the source machine to the network layer on the destination machine.

Possible services offered

- Unacknowledged Connection Less
- Acknowledged Connection Less
- Acknowledged Connection Oriented
- **Acknowledgement**
  - A transmission from the receiver indicating that data is received.
- **Connection oriented**
  - The sender and the receiver agree upon certain parameters before actual data transmission.
  - To ensire reliablity

2. Framing

- Transmission of the data link layer starts with breaking up the bit stream into discrete FRAMES.
- Computation of a CHECKSUM for each frame.
- CHECKSUM is a value which is based on the frame.
- A single bit change would change the checksum.
- Include the checksum into the frame before it is transmitted.
- Receiver computes its checksum error for a receiving frame

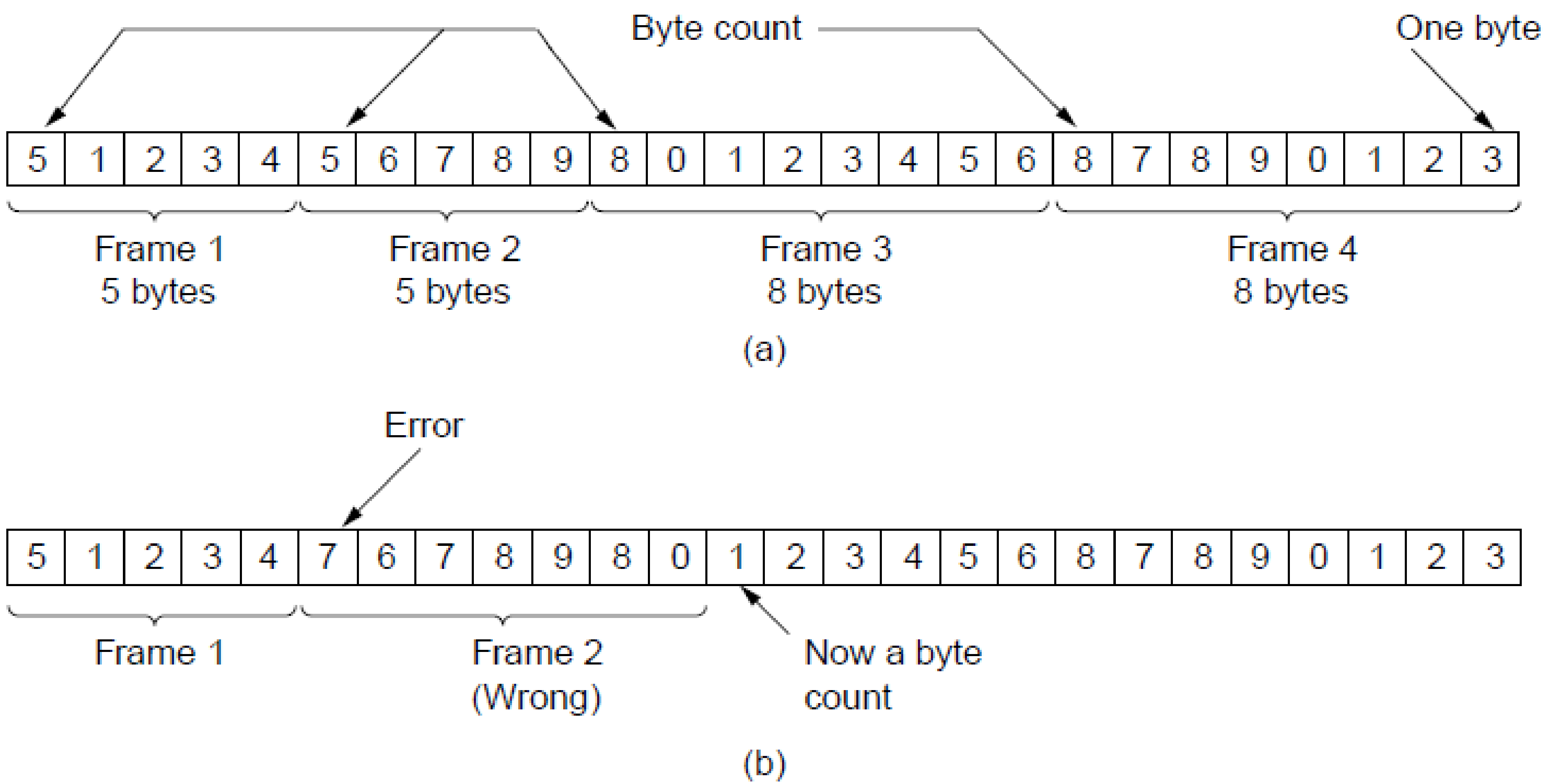
Framing methods.

1.Including count.

•Count indicates the number of bytes in a frame.

•The receiver would interpret the first field as count of byte values.

•The count value can get corrupted during transmission.



Byte count
One byte
5 1 2 3 4 5 6 7 8 9 8 0 1 2 3 4 5 6 8 7 8 9 0 1 2 3
Frame 1
5 bytes
Frame 2
5 bytes
Frame 3
8 bytes
Frame 4
8 bytes
(a)
Error
5 1 2 3 4 7 6 7 8 9 8 0 1 2 3 4 5 6 8 7 8 9 0 1 2 3
Frame 1
Frame 2
(Wrong)
Now a byte
count
(b)

Framing methods.

2.Byte stuffing

- A special byte sequence is included at the starting and at the end of the frame.
- The special sequence is also known as FLAG byte.
- This process gets repeated .At the receiver the special sequence is neglected.
- **What if the user data has the same sequence ?**
- Soln :- Put another special sequence before the data.
- Also known as escape sequence.

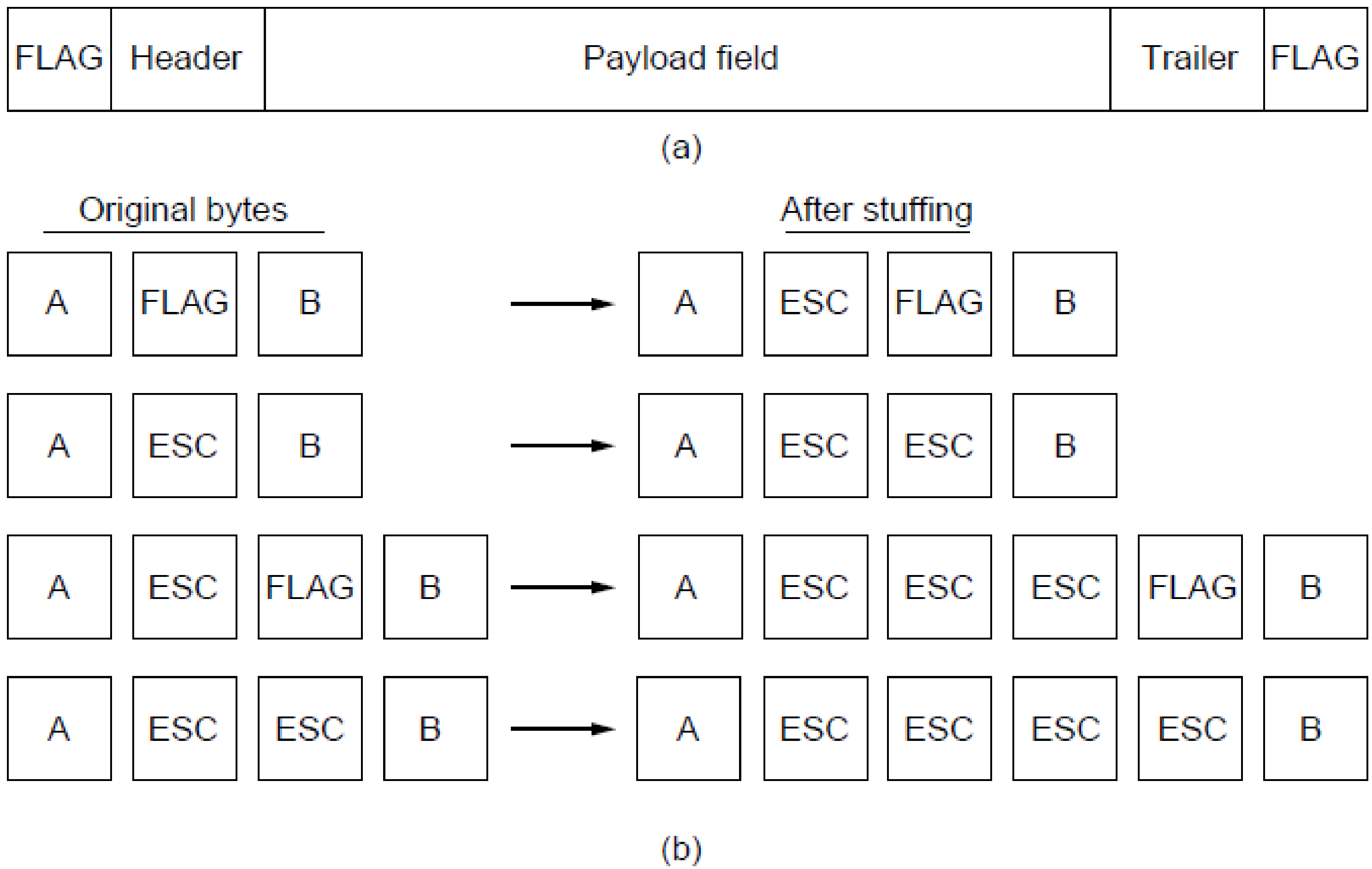


Fig a .A frame delimited by flag bytes.

Fig b.Four examples of byte sequences before and after byte stuffing.

Framing methods

3.Bit stuffing.

- Instead of using one full byte as an escape sequence it is possible to use a bit.
- Each frames begins and ends with a special bit pattern:
- 01111110 ----Flag Byte
- Whenever the sender's data link layer encounters five consecutive 1s in the ***user data*** it automatically stuffs a 0 bit into the outgoing bit stream.
- Done to exclude the chance of 01111110 sequence in user data.

On the receiving side

5 consecutive 1’s

Next bit 0 : Stuffed, so discard it

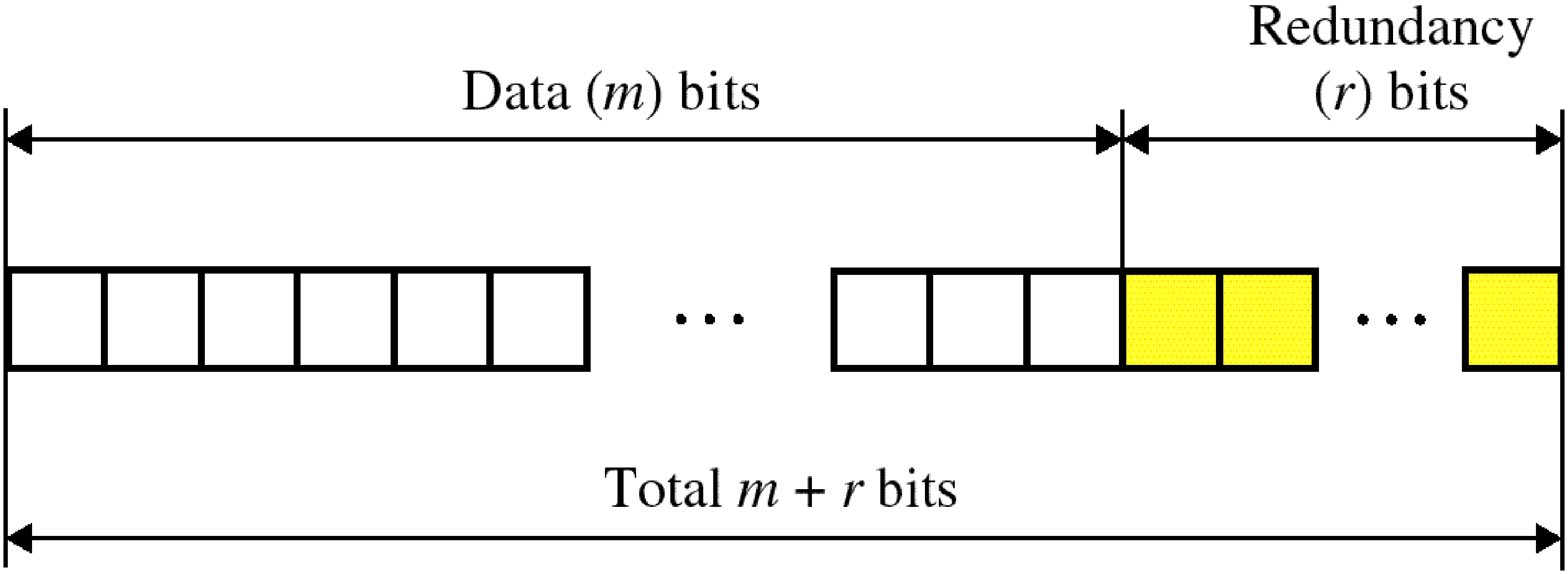
1 : Either End of the frame marker/Start

Or Error has been introduced in the bitstream.

Error detection and correction

- Errors occur as a result of bits getting flipped.[0->1 or 1->0].
- These errors cannot be completely avoided.
- The only way to address is by detecting errors and finding ways to correct them.
- **If error detection and correction has to be enforced ,extra bits needs to be added with the data.**
- These extra bits are called ***redundant bits.***[In some cases referred as parity bits]
- The number of redundant bits depend on the scheme which is being used.
- In worst case , if total **B** bits are transmitted **B/2 will be redundant bits and B/2 will be original data.**
- **A frame consists of m ->message bits,r->redundant bits and total frame size n=m+r**
-

Ref: Andrew S Tanenbaum


Redundancy
(r) bits
Data (m) bits
Total m + r bits

Error correction codes

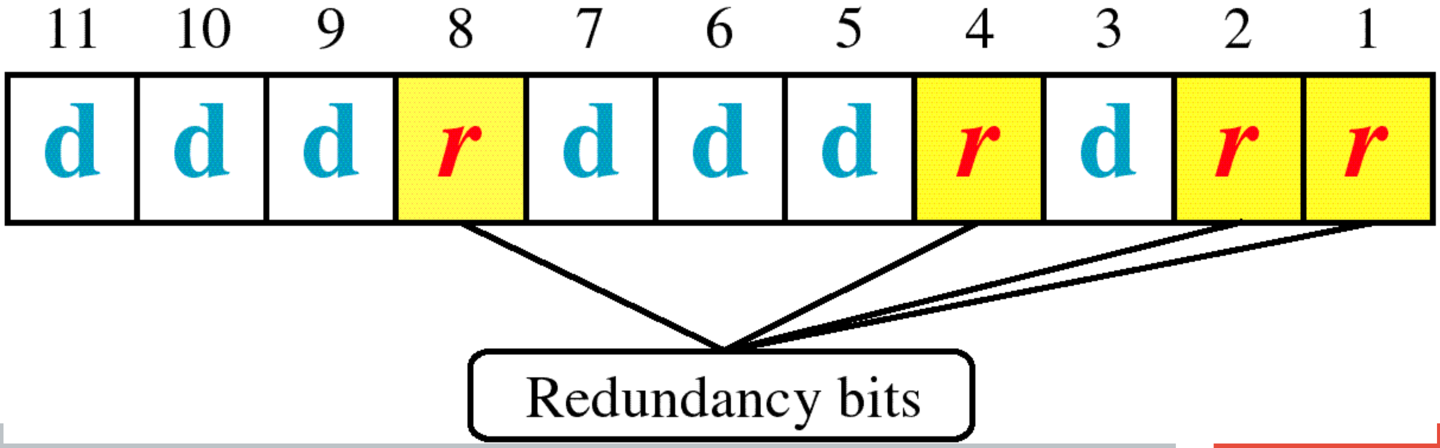
1.Hamming codes.

•Code words are constructed by a combination of redundant bits and message bits.

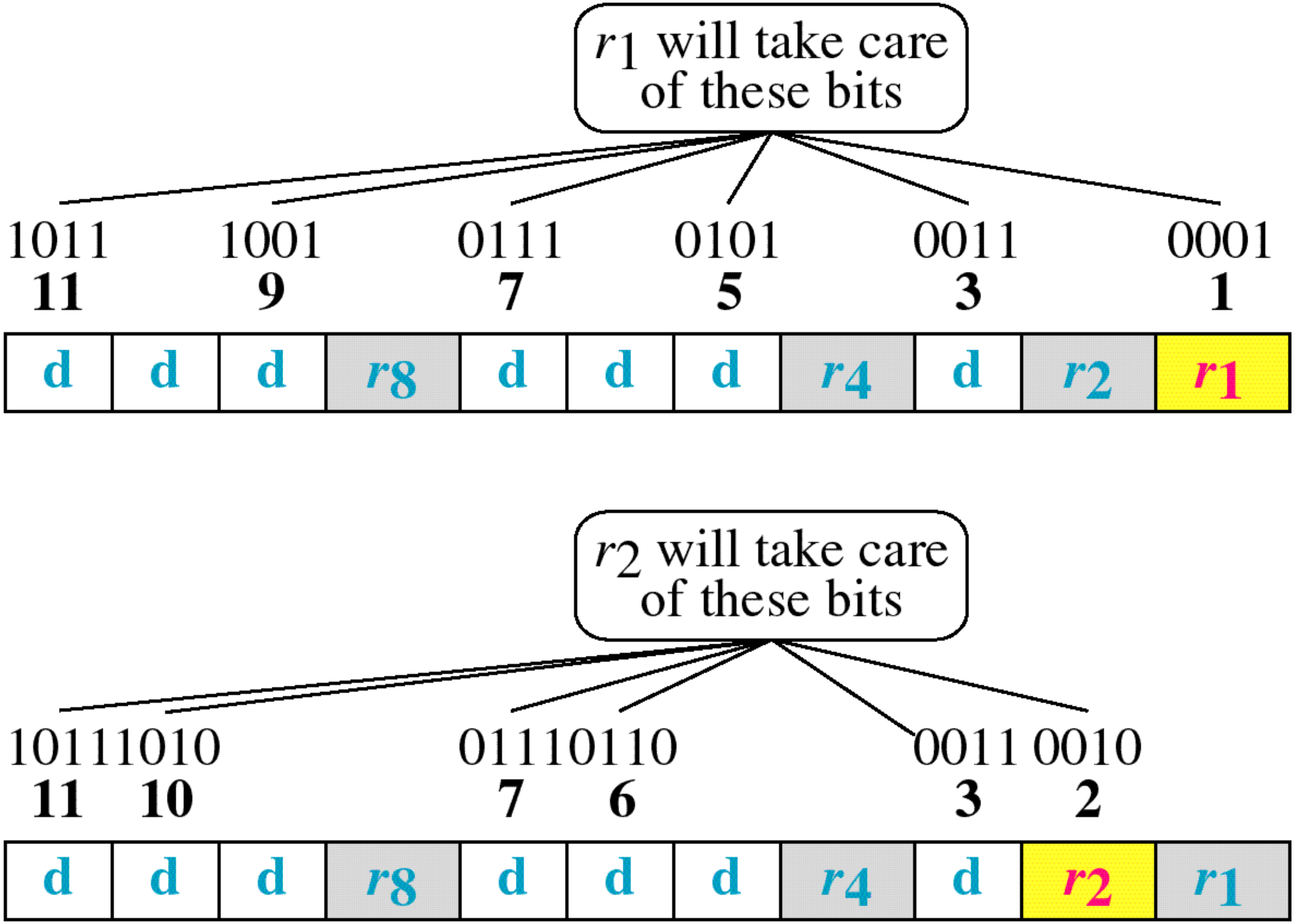
•The codes are represented as **(n,m)**

•Ex (11,7) -Frame has total 11 bits ,only 7 are message.

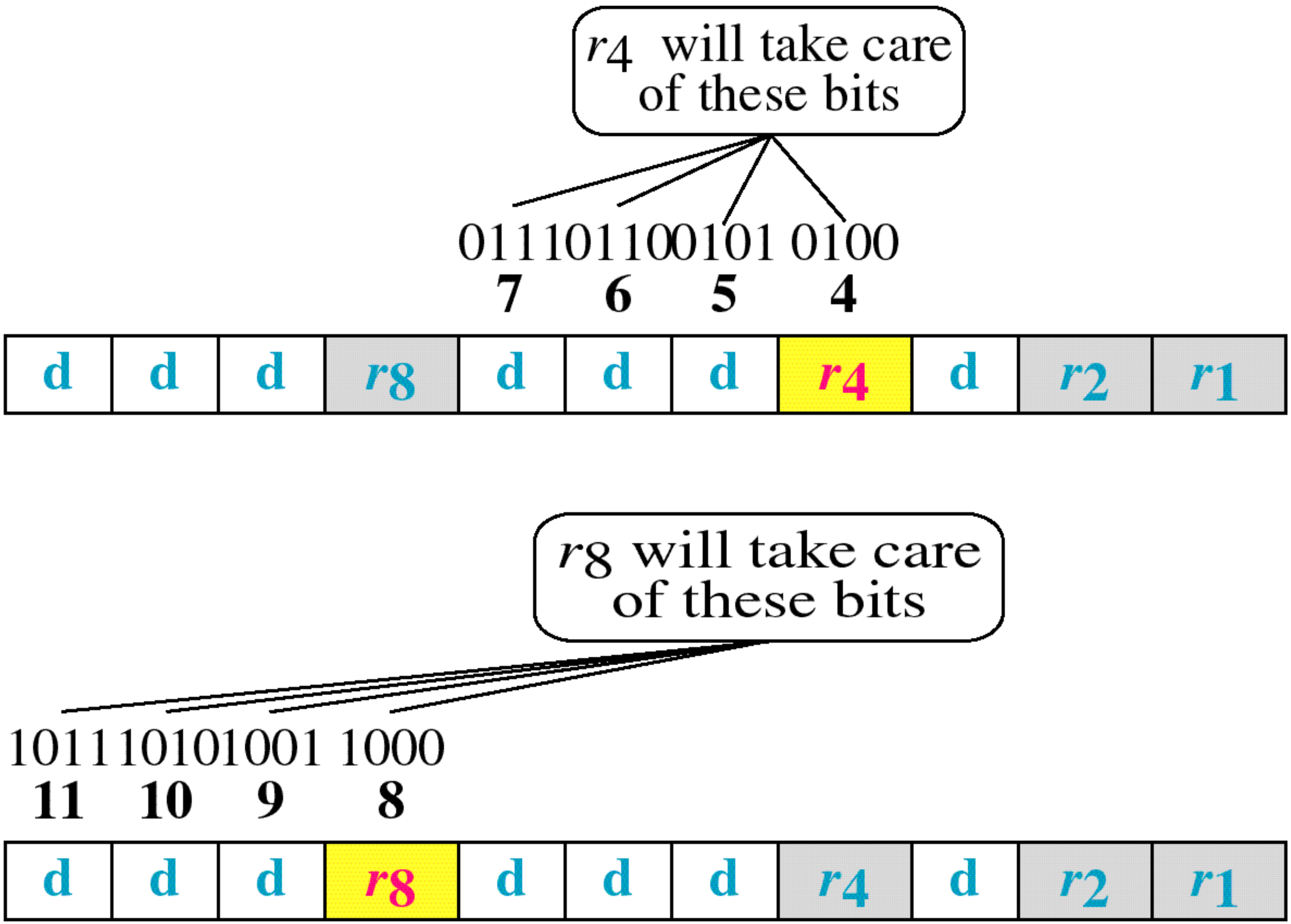
•In the frame, bit positions 2^0,2^1,2^2 ,..2^*k* are occupied by redundant bits.

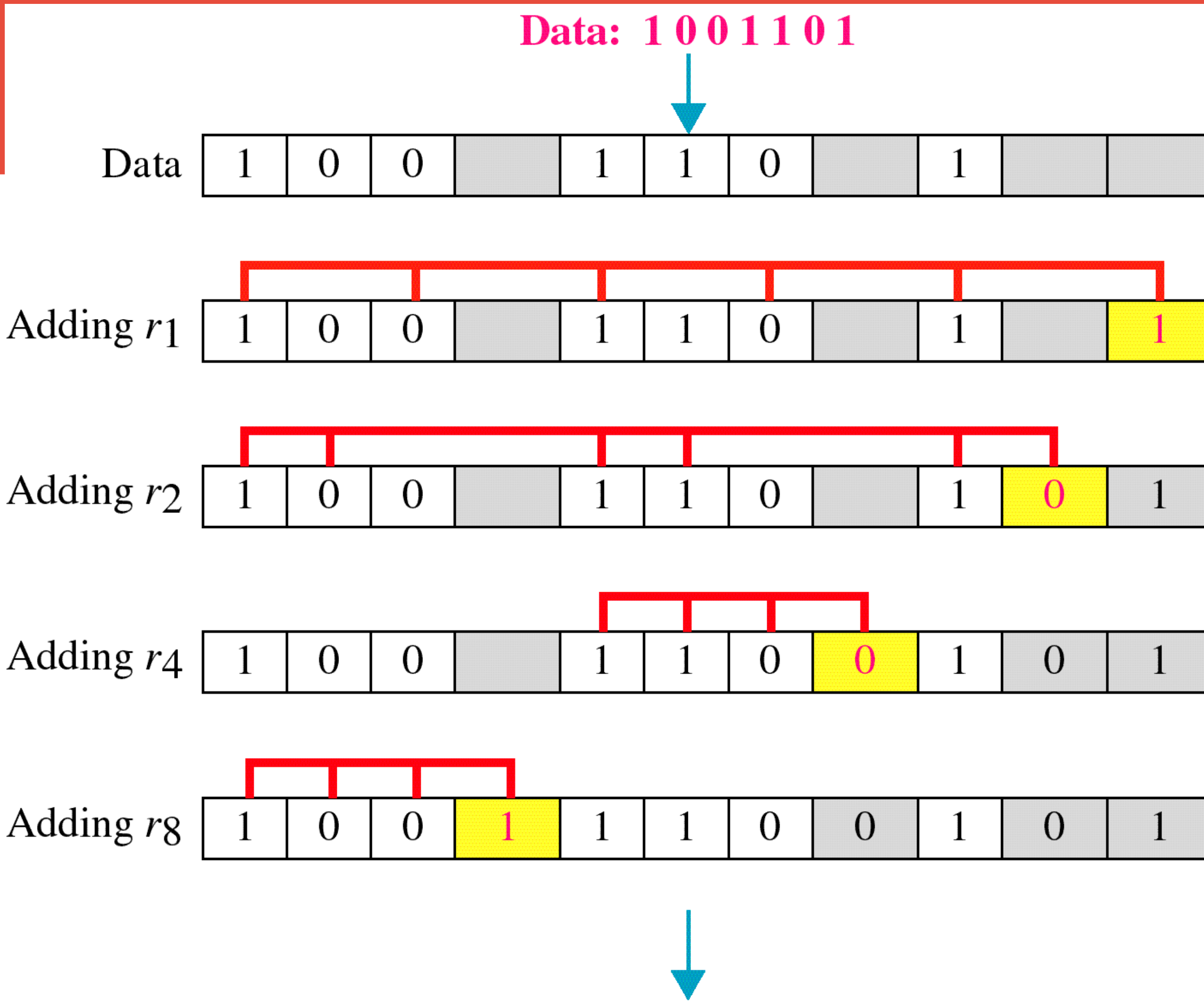


Contd..



r1 will take care of these bits
1011 1001 0111 0101 0011 0001
11 9 7 5 3 1
d d d r8 d d d r4 d r2 r1
r2 will take care of these bits
1011 1010 0111 0110 0011 0010
11 10 7 6 3 2
d d d r8 d d d r4 d r2 r1

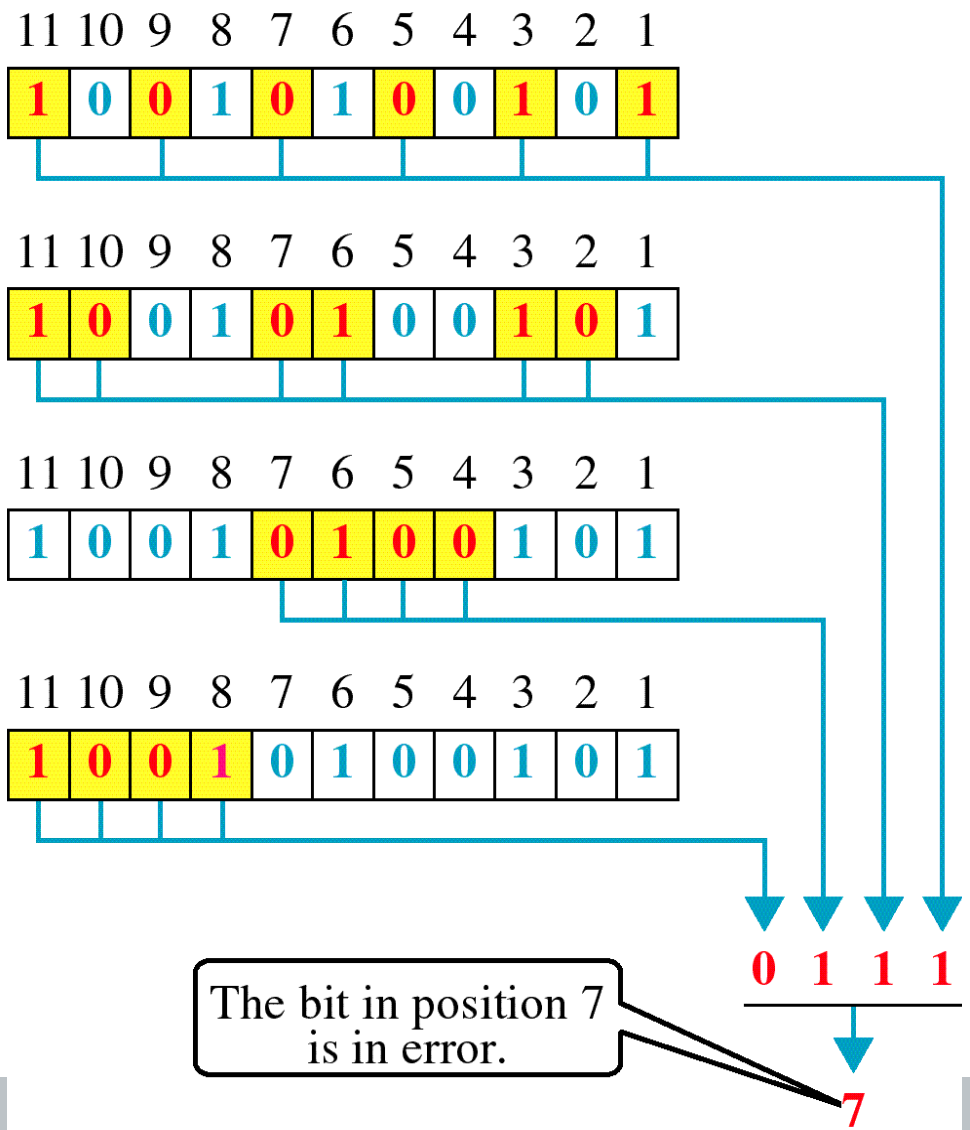

r4 will take care of these bits
011101100101 0100
7 6 5 4
d d d r8 d d d r4 d r2 r1
r8 will take care of these bits
101110101001 1000
11 10 9 8
d d d r8 d d d r4 d r2 r1



XOR between bit positions to find the value of redundant bit

Some text books assume bit position Zero at left side.

At the receiver

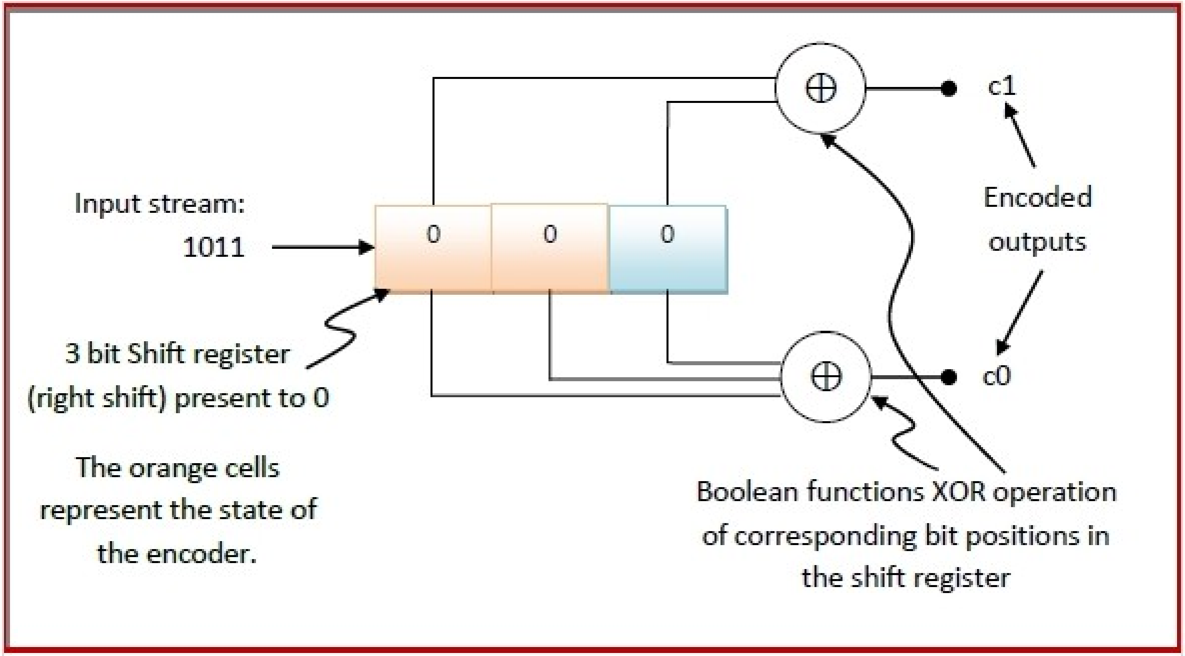


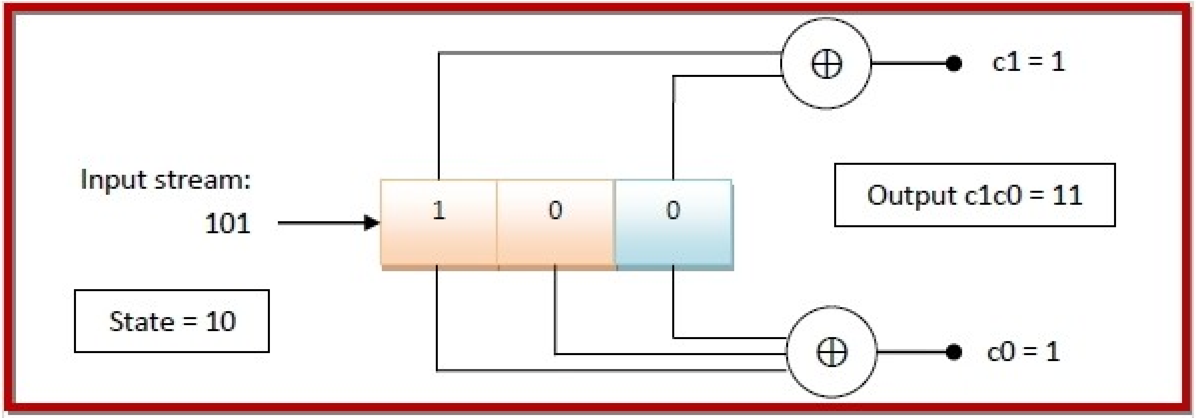
Hamming distance

- The distance between two codewords.
- Simple XOR between the code words will give the hamming distance.
- Ex:-00001111 ,00000000 Hamming distance of 4. (XOR output will contain four 1's)
- If hamming distance is d,the code can detect k errors where k<=d-1.
- If hamming distance is d,the code can correct c errors where c<=(d-1)/2.
-

2.Convolutional Codes.

- Convolutional codes system produces code words based on a combination of previous input and current state.
- A shift register will be employed .
- Each input bit produces more than one output bits.
- The system is represented as (n,k,T)
- n->Number of output
- k->No of shift [Normally 1]
- T->Max size of shift register.


c1
Input stream:
1011
0
0
0
Encoded
outputs
3 bit Shift register
(right shift) present to 0
c0
The orange cells
represent the state of
the encoder.
Boolean functions XOR operation
of corresponding bit positions in
the shift register


c1 = 1
Input stream:
101
1
0
0
Output c1c0 = 11
State = 10
c0 = 1

3.Reed Solomon Code

- Represented as RS(n,m) where m is the size of the message.
- n is the total size after adding redundant bits.
- Can correct upto *t* errors where t=(n-m)/2.
- Uses polynomial based functions to generate code words.
- Standard polynomial functions are made to encode and decode the transmission.

Contd..

4.Low density parity check.

Each output bit is formed by a fraction of the input bits.

Leads to a low density of 1s.

Assume data to be sent is 1001

c1	c2	c3	c4
1	0	0	1

It has to meet the equation

$$c1 \oplus c2 \oplus c3 \oplus c5 = 0 \qquad e1$$
$$c1 \oplus c2 \oplus c4 \oplus c6 = 0 \qquad e2$$
$$c1 \oplus c3 \oplus c4 \oplus c7 = 0 \qquad e3$$

$$1 \oplus 0 \oplus 0 \oplus c5 = 0$$
$$1 \oplus 0 \oplus 1 \oplus c6 = 0$$
$$1 \oplus 0 \oplus 1 \oplus c7 = 0$$

This results in the folowing seven bit sequence.

```
c1  c2  c3  c4  c5  c6  c7
1   0   0   1   1   0   0
```

At the receiver the same equations are employed.

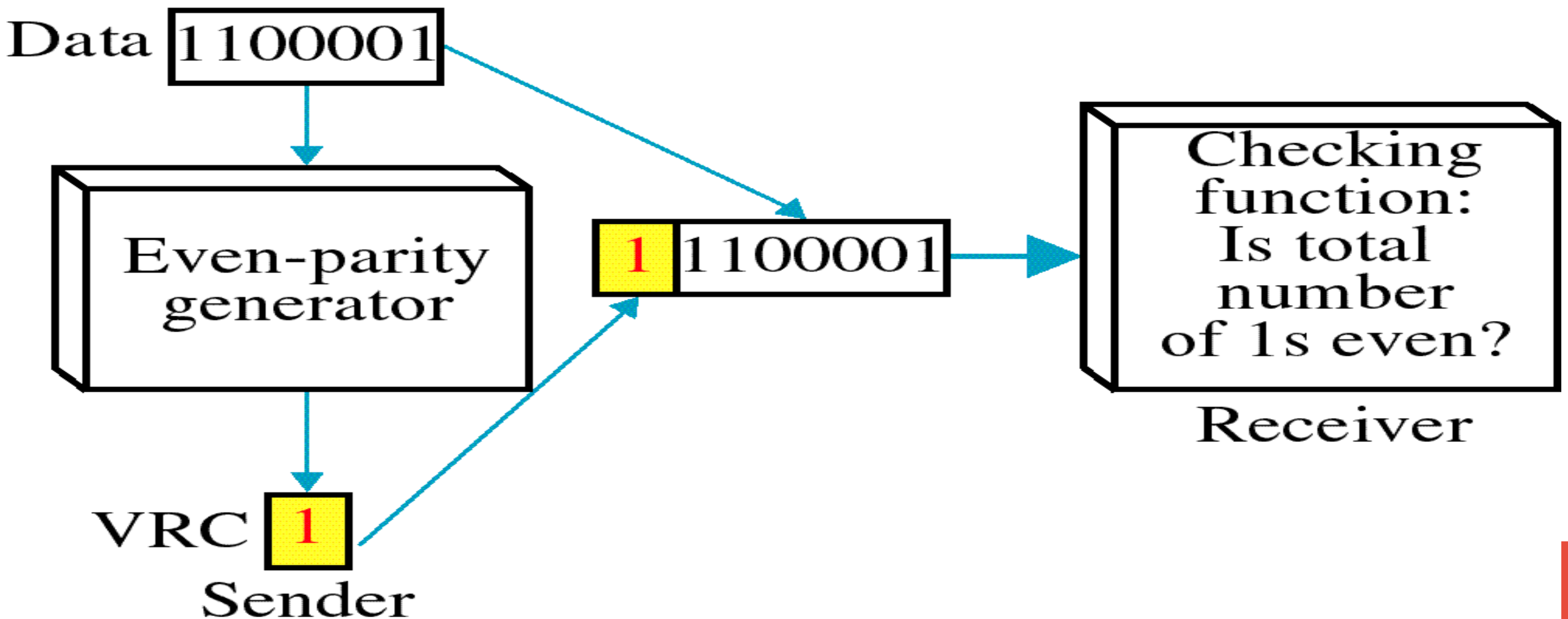
Any change of the redundant bits will notify us of the error.

Error detection mechanism

1.Parity bits

Special bits to identify the presence of errors.

Even parity -> The number of 1’s is even.



2.Checksums

- Works on input sections of data.
- Produces a unqiue value.

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.
- This result is added with checksum.
- The sum is complemented.
- If the result is zero, the data are accepted: otherwise, they are rejected.

3.Cyclic Redundancy Check [CRC]

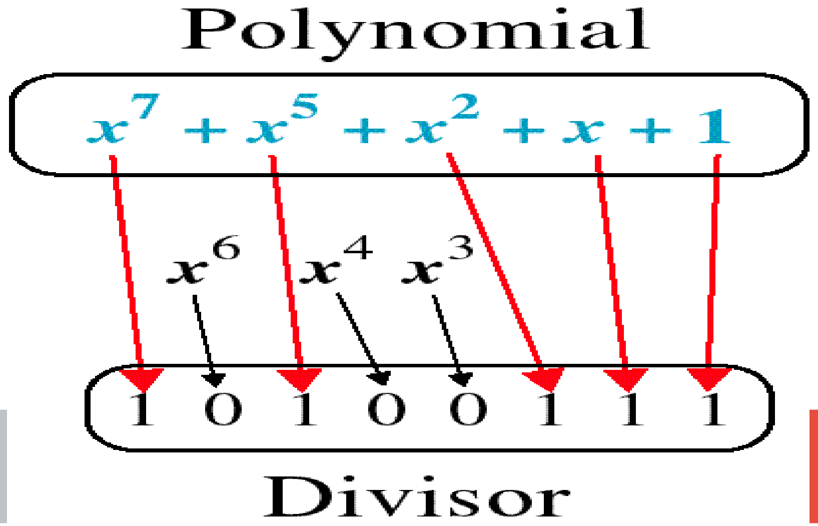
•Also known as polynomial strings.

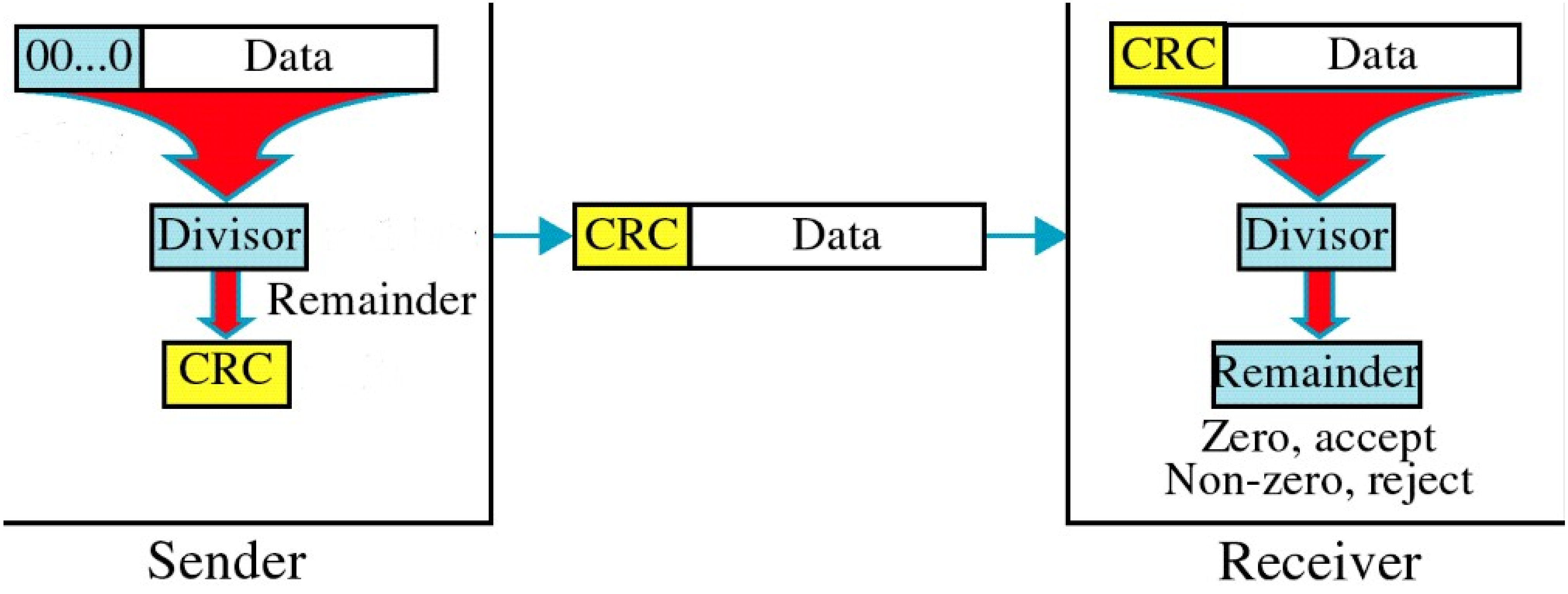
•Uses polynomial binay division to detect errors.

• The data to be transmitted is divided with special polynomials known as generator polynomial.

•Generator polynomials should be smaller than data.

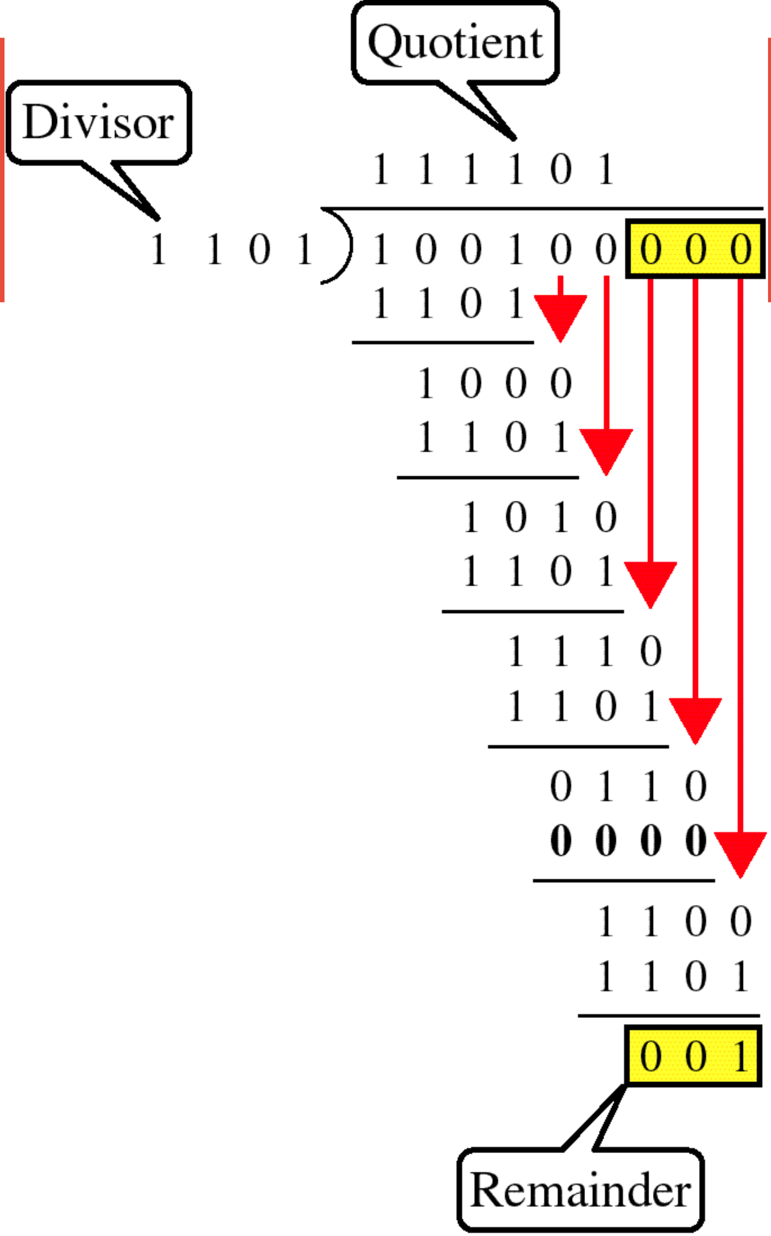
•The remainder is CRC which is added with the data and is sent.





If the generator polynomial has a degree of *d*,append *d* zeros at the LSB part.
[Here it appears to be at the MSB though!!]

At receiver



If the remainder is zero,there is no error

Some standard polynomials

CRC-12

$$x^{12} + x^{11} + x^3 + x + 1$$

CRC-16

$$x^{16} + x^{15} + x^2 + 1$$

CRC-ITU

$$x^{16} + x^{12} + x^5 + 1$$

CRC-32

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$