Computer Network(CSC 503)

Shilpa Ingoley

Lecture 10

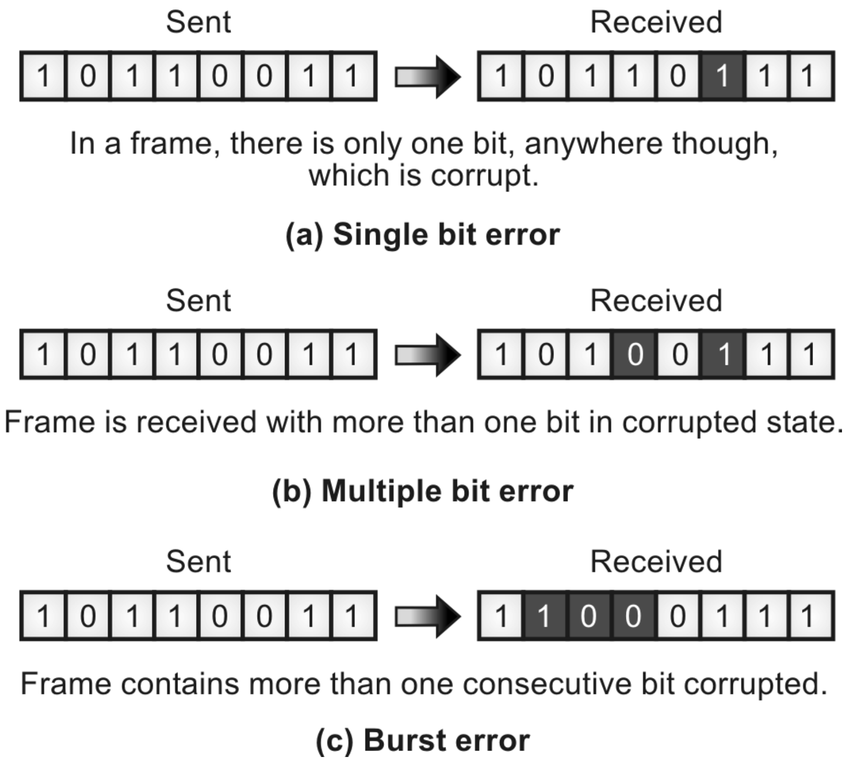
Chap3: Data link Layer

- Error Detection and Correction(Hamming Code, CRC, Checksum)

Types of Error

Following 3 types

1. **Single bit error**
2. **Multiple bits error**
3. **Burst error**



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.
- Error Detection and Correction method →Hamming Code(**single bit error correction and double bit error detection)**

Hamming Code

- It is technique developed by R.W. Hamming for error correction.
- **Redundant bits**
- Redundant bits are **extra** binary bits that are generated and added to the information carrying bits of data transfer to ensure that no bits were lost during the data transfer.
- The number of redundant bits can be calculated using the following formula:

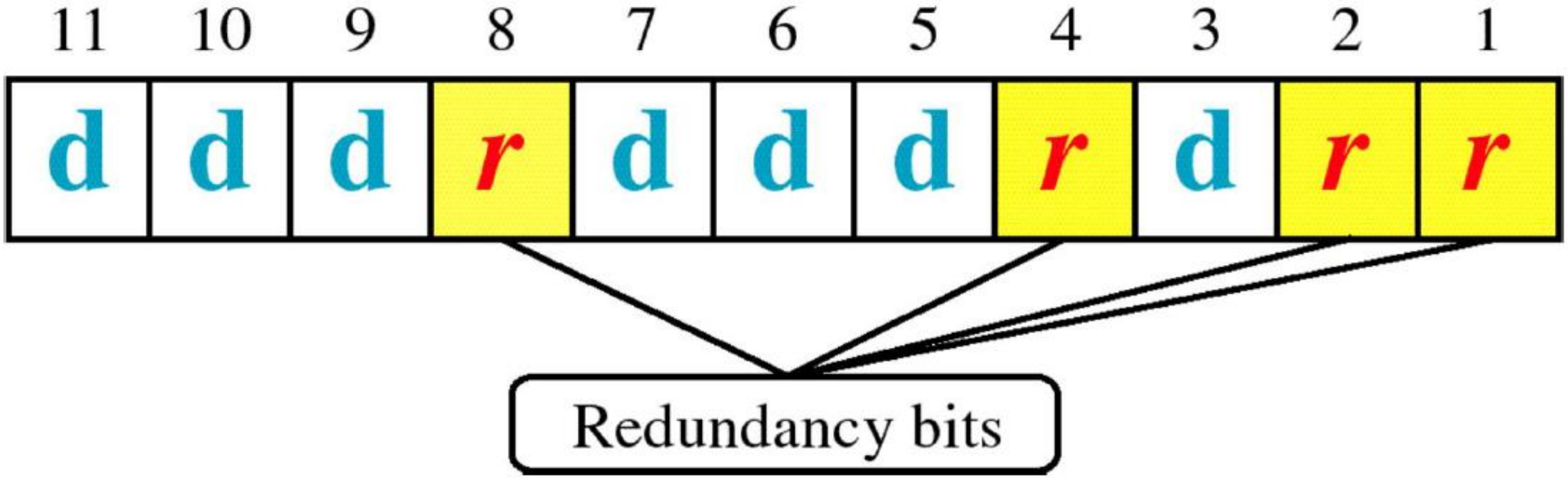
$$2^r \geq m + r + 1$$

where, r = redundant bit, m = data bit

- Suppose the number of data bits is 7, then the number of redundant bits can be calculated using:
- Thus, the number of redundant bits= 4

$$= 2^4 \geq 7 + 4 + 1$$

Hamming Code



Contd…

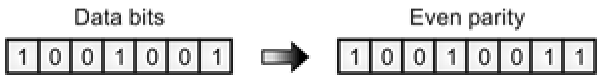
Parity bits

- A parity bit is a bit appended to a data of binary bits to ensure that the total number of 1's in the data is even or odd.
- Parity bits are used for error detection. There are two types of parity bits:

Even parity bit:

In this, the number of 1's are counted. If that count is odd, the parity bit value is set to 1, making the total count of occurrences of 1's an even number.

If the total number of 1's in a given set of bits is already even, the parity bit's value is 0.



Odd Parity bit

- In the case of odd parity, for a given set of bits, the number of 1's are counted. If that count is even, the parity bit value is set to 1, making the total count of occurrences of 1's an odd number.
- If the total number of 1's in a given set of bits is already odd, the parity bit's value is 0.

Data Bits:1011

- 1→0 0 1
- 2→0 1 0
- 3→0 1 1
- 4→1 0 0
- 5→1 0 1
- 6→1 1 0
- 7→1 1 1

Sender

7-bit hamming code C (7,4)

D7	D6	D5	P4	D3	P2	P1

p1=XOR(3,5,7)=1
P2=XOR(3,6,7) =0
P4=XOR(5,6,7) = 0

Put the values of P1,P2 and P4

1	0	1	0	1	**0**	**1**

Receiver

7-bit hamming code C (7,4)

D7	D6	D5	P4	D3	P2	P1
1	0	1	0	1	**0**	**1**

Compute
C1=XOR(1,3,5,7)
C2=XOR(2,3,6,7)
C3=XOR(4,5,6,7)

C= C3 C2 C1
C = 0 0 0← No Error

If Data Received is:

1	0	0	0	1	**0**	**1**

C1=XOR(1,3,5,7)= 1
C2=XOR(2,3,6,7) = 0
C3=XOR(4,5,6,7) = 1
C= C3 C2 C1
C = 1 0 1 ← **Error in 5th bit**

Data bits: 00100011

15-bit hamming code C (15, 11)

D15	D14	D13	D12	D11	D10	D9	P8	D7	D6	D5	P4	D3	P2	P1

Put the values of data Compute P1,P2 P4 and P8

	D12	D11	D10	D9	P8	D7	D6	D5	P4	D3	P2	P1
	0	0	1	0	P8	0	0	1	P4	1	P2	P1

p1=XOR(3,5,7,9,11)=0
P2=XOR(3,6,7,10,11) =0
P4=XOR(5,6,7,12) = 1
P8=XOR(9,10,11,12) = 1

	0	0	1	0	1	0	0	1	1	1	0	0

1→ 0 0 0 1
2→ 0 0 1 0
3→ 0 0 1 1
4→ 0 1 0 0
5→ 0 1 0 1
6→ 0 1 1 0
7→ 0 1 1 1
8→ 1 0 0 0
9→ 1 0 0 1
10→1 0 1 0
11→1 0 1 1
12→1 1 0 0

13→1 1 0 1
14→ 1 1 1 0
15→1 1 1 1

Contd…

- **Double bit error detection**

- The Hamming code can be modified to **detect double bit errors** by adding a parity bit as the MSB, which is the XOR of all other bits.

- Suppose that the message **00100011** needs to be encoded using even parity Hamming code.

- Hence, the message sent will be(after adding the redundant or parity bits).

P13	0	0	1	0	1	0	0	1	1	1	0	0

- After adding P13 = XOR(1..12) = 1, the new codeword to be sent will be

1	0	0	1	0	1	0	0	1	1	1	0	0

Contd…

Double bit error detection

		P13	D12	D11	D10	D9	P8	D7	D6	D5	P4	D3	P2	P1

1	0	0	1	0	1	0	0	1	1	1	0	0

- P13 = XOR(1,2,..12)

At Receiver side

- P=XOR(1,2 ….13)

- If C=0 and P=0 → No Error occur
- If C !=0 and P =1 → A Single Error occurred which is detected and can be corrected
- If C !=0 and P =0 → A double Error occurred which is detected but cannot corrected
- If C =0 and P =1 → An Error occur in P13 bits

Note: This scheme can not detect more than two errors

More Examples :

For 7 bits data

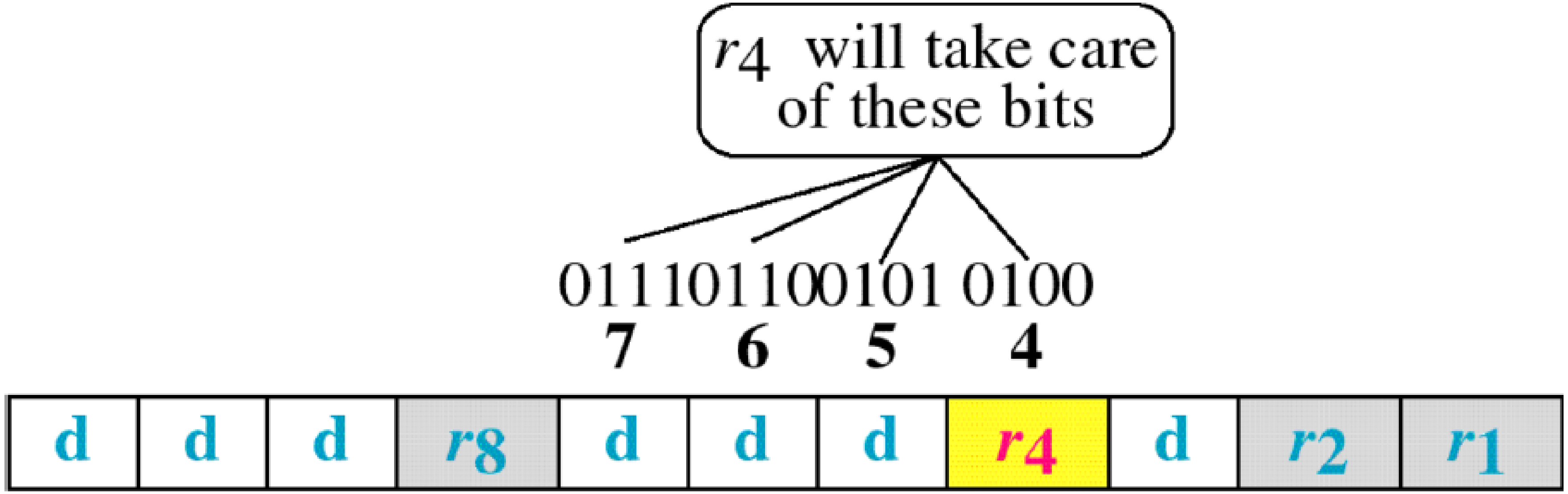
r_1 will take care of these bits.

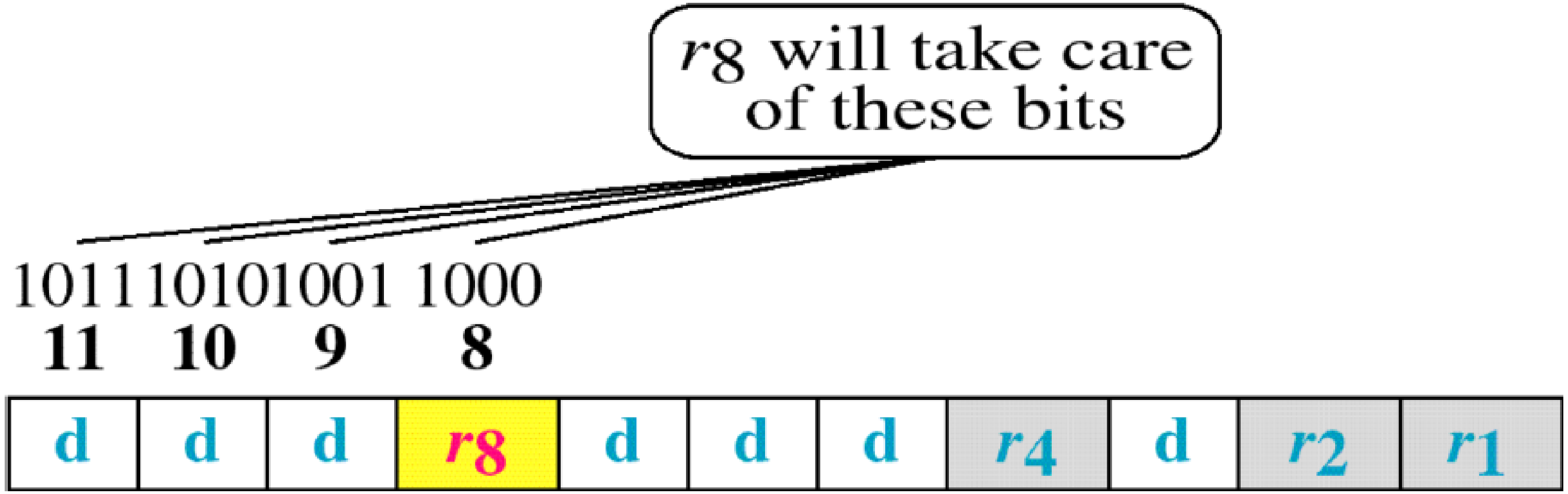
11		9		7		5		3		1
d	d	d	r_8	d	d	d	r_4	d	r_2	r_1

r_2 will take care of these bits.

11	10			7	6			3	2	
d	d	d	r_8	d	d	d	r_4	d	r_2	r_1

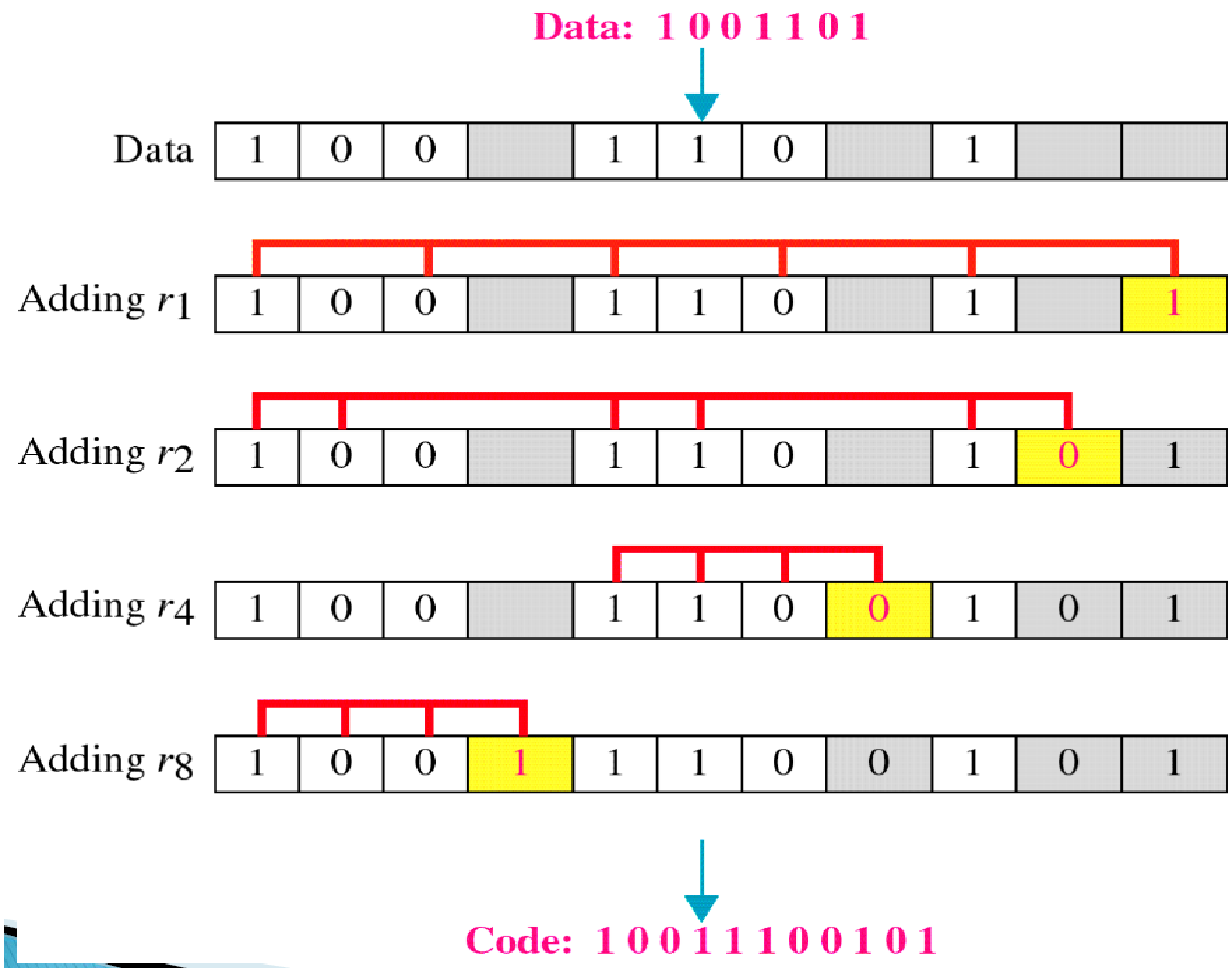
Contd…


r4 will take care of these bits
0111 1011 1001 01 0100
7 6 5 4
d d d r8 d d d r4 d r2 r1


r8 will take care of these bits
1011 1010 1001 1000
11 10 9 8
d d d r8 d d d r4 d r2 r1

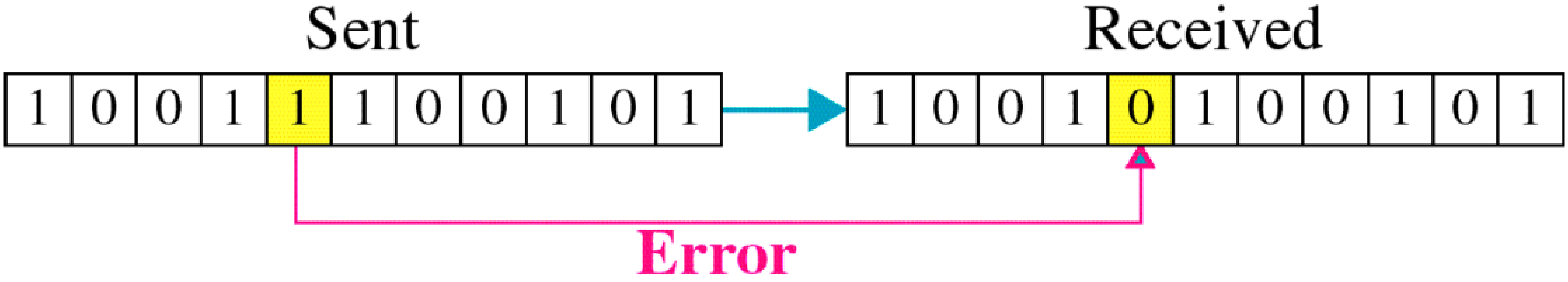
Example of Hamming Code

Contd...



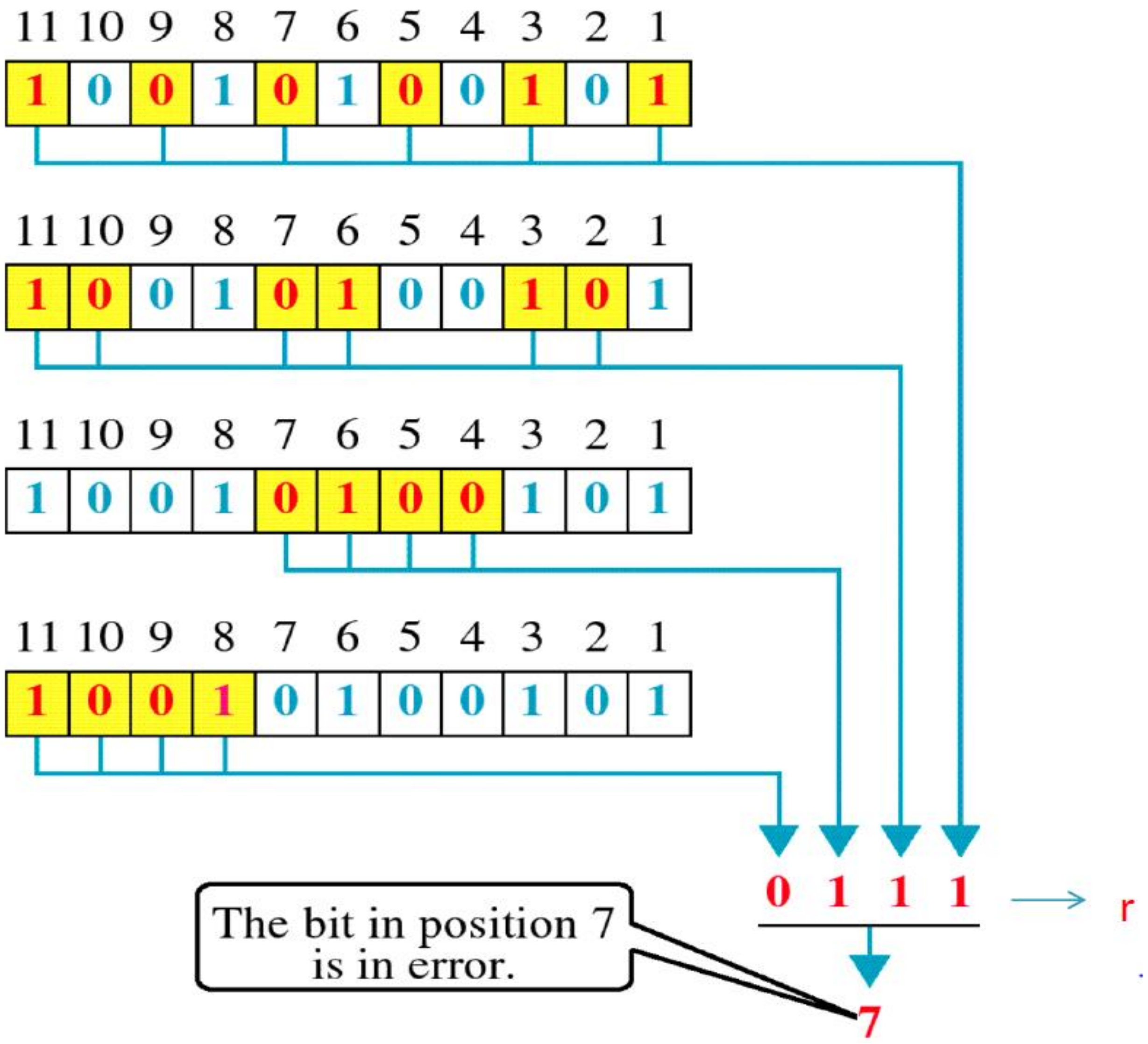
Contd…

- **Single-bit error**



Contd…

Error Detection



Contd...

- Eg: 2
- ▸Suppose the data to be transmitted is 1011001, the bits will be placed as follows:
- Thus, the data transferred is:

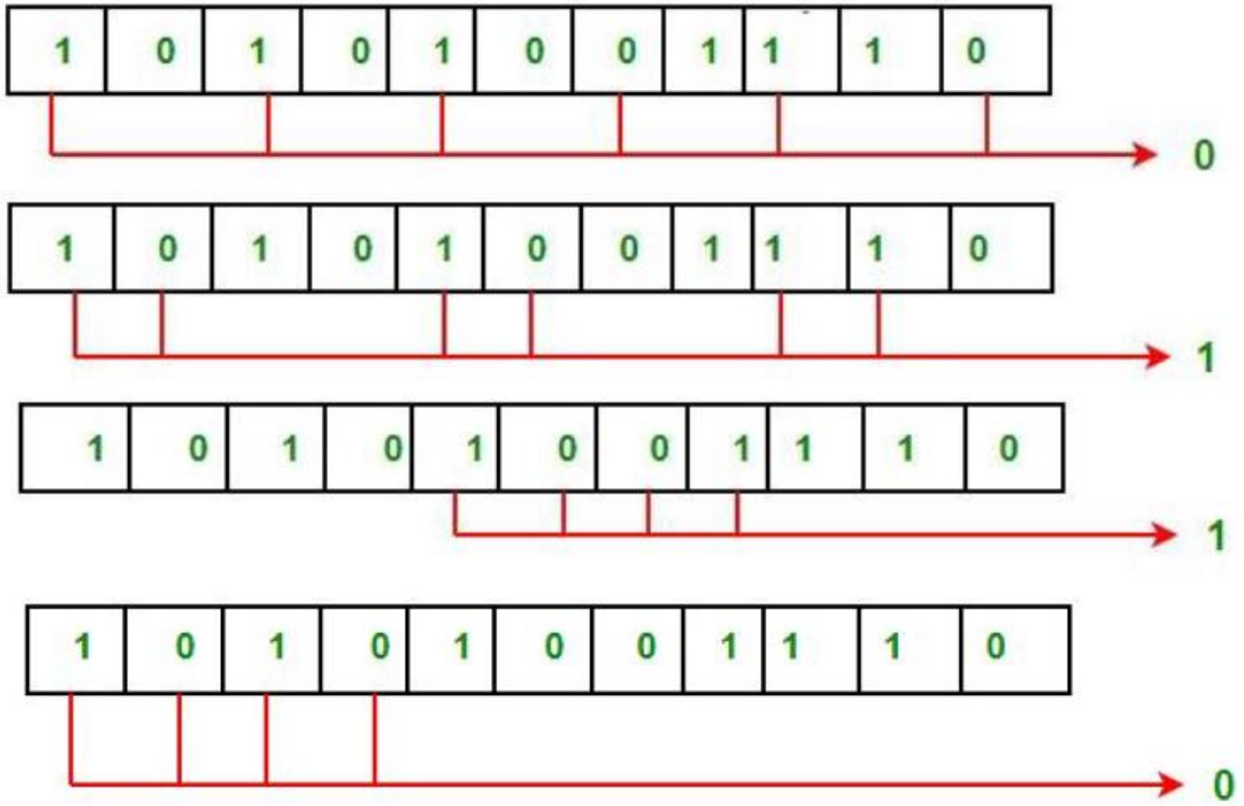
11	10	9	8	7	6	5	4	3	2	1
1	0	1	R8	1	0	0	R4	1	R2	R1

- Thus, the data transferred is:

11	10	9	8	7	6	5	4	3	2	1
1	0	1	0	1	0	0	1	1	1	0

Contd…

Suppose in the above example the 6th bit is changed from 0 to 1 during data transmission, then it gives new parity values in the binary number



- The bits give the binary number as 0110 whose decimal representation is 6. Thus, the bit 6 contains an error. To correct the error the 6th bit is changed from 1 to 0.

Contd…

OR Example :

- Suppose that the message **11000100** needs to be encoded using even parity Hamming code.
- Hence, the message sent will be, 11000101100 (after adding the redundant or parity bits).
- After adding P = XOR(1,1,0,0,0,1,0,1,1,0,0) = 1, the new codeword to be sent will be 11000101100P13. i.e 110001011001

Assuming 2 bits are changed(last 2 lsb bits changed to 11)

- Received code word –**1**11000101111
- New parity -1
- $r \neq 0$
- Hence Double bit error detected

Contd...

Alternatively can be Calculated:

- At the receiver's end, error detection is done as shown in the following table .
- Note : r-value(Check values-C) is the binary value obtained after checking the redundant bit positions

r-value	P value	Conclusion
$r=0$	$P(sender)=P(receiver)$	No error
$r=0$	$P(sender)\neq P(receiver)$	Error in P bit, Data can be sent to the uppers layers after removing all check bits.
$r\neq 0$	$P(sender)\neq P(receiver)$	Single bit error occurred that can be corrected by reversing the bit value at the bit position given by value r
$r\neq 0$	$P(sender)=P(receiver)$	Double bit error detected that cannot be corrected.