



Indian Institute of Information Technology,
Nagpur

Course: Computer Networks
(CSL 302, Core)
5th Semester



Topics Covered

- Error Control

Dr. Aishwarya Ukey
Assistant Professor
Dept. of CSE, IIIT Nagpur

Error Control



- Bit errors are sometimes introduced into frames owing to transmission impairments i.e. electrical interference or noise
- Although errors are rare, some mechanism is needed to detect these errors so that corrective action can be taken
- Detecting errors - one part of the problem
- Other part - correcting errors once detected

Cont...

- Two basic approaches can be taken when the recipient of a message detects an error
 - ▣ Notify the sender that the message was corrupted so that the sender can retransmit a copy of the message
 - ▣ Some types of error detection algorithms allow the recipient to reconstruct the correct message even after it has been corrupted

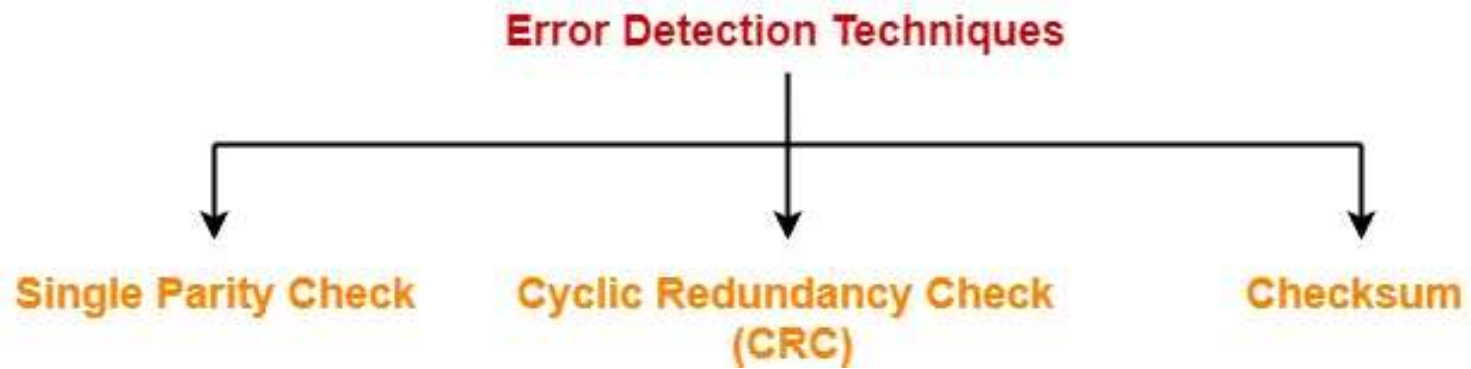
Cont...



- The basic idea behind any error detection scheme
 - ▣ To add redundant information to a frame that can be used to determine if errors have been introduced
- In the extreme case
 - ▣ Imagine transmitting two complete copies of the data. If the two copies are identical at the receiver, then it is probably the case that both are correct
 - ▣ If they differ, then an error was introduced into one (or both) of them, and they must be discarded

Cont...

□ Error Detection Methods



Single Parity

□ Single Parity

- ▣ One extra bit called as parity bit is sent along with the original data bits
- ▣ Parity bit helps to check if any error occurred in the data during the transmission

□ Steps 1

- ▣ At sender side,
 - Total number of 1's in the data unit to be transmitted is counted
 - The total number of 1's in the data unit is made even in case of even parity
 - The total number of 1's in the data unit is made odd in case of odd parity
 - This is done by adding an extra bit called as parity bit

Cont...

□ Step-02:

- ▣ The newly formed code word (Original data + parity bit) is transmitted to the receiver

□ Step-03:

- ▣ At receiver side,

- Receiver receives the transmitted code word
- The total number of 1's in the received code word is counted


Cont...

- Then, following cases are possible
 - ▣ If total number of 1's is even and even parity is used, then receiver assumes that no error occurred
 - ▣ If total number of 1's is even and odd parity is used, then receiver assumes that error occurred
 - ▣ If total number of 1's is odd and odd parity is used, then receiver assumes that no error occurred
 - ▣ If total number of 1's is odd and even parity is used, then receiver assumes that error occurred

Cont...

- Example: Consider the data unit to be transmitted is 1001001 and even parity is used
- At Sender Side
 - ▣ Total number of 1's in the data unit is counted.
 - ▣ Total number of 1's in the data unit = 3.
 - ▣ Clearly, even parity is used and total number of 1's is odd.
 - ▣ So, parity bit = 1 is added to the data unit to make total number of 1's even.
 - ▣ Then, the code word 10010011 is transmitted to the receiver.

Cont...



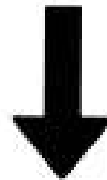
1	0	0	1	0	0	1
---	---	---	---	---	---	---

Original data unit

Cont...

1	0	0	1	0	0	1
---	---	---	---	---	---	---

Original data unit



Parity bit



1	0	0	1	0	0	1	1
---	---	---	---	---	---	---	---

Transmitted data unit

Cont...

□ At Receiver Side

- After receiving the code word, total number of 1's in the code word is counted.
 - Consider receiver receives the correct code word = 10010011.
 - Even parity is used and total number of 1's is even.
 - So, receiver assumes that no error occurred in the data during the transmission.

Cont...

□ Advantage

- ▣ This technique is guaranteed to detect an odd number of bit errors (one, three, five and so on)
- ▣ If odd number of bits flip during transmission, then receiver can detect by counting the number of 1's

□ Limitation

- ▣ This technique can not detect an even number of bit errors (two, four, six and so on)
- ▣ If even number of bits flip during transmission, then receiver can not catch the error

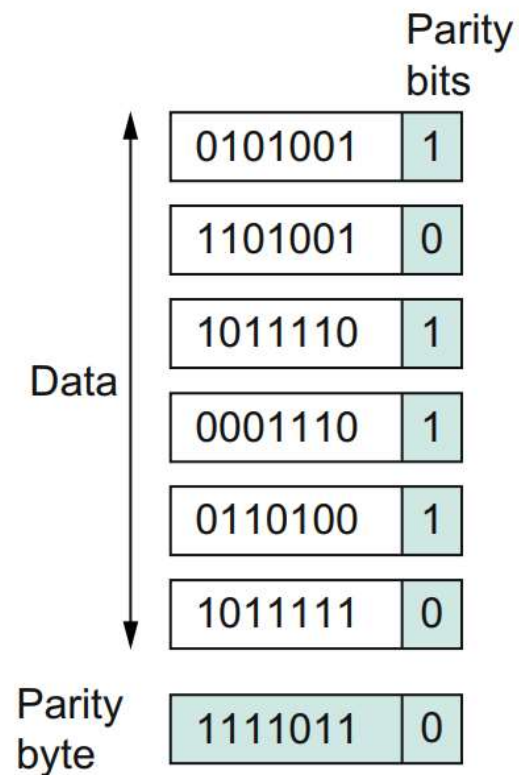
Two-Dimensional Parity



- Two-dimensional parity does a similar calculation for each bit position across each of the bytes contained in the frame
- This results in an extra parity byte for the entire frame, in addition to a parity bit for each byte

Cont...

- Figure illustrates how two-dimensional even parity works for an example frame containing 6 bytes of data



Two-dimensional parity catches all 1-, 2-, and 3-bit errors, and most 4-bit errors.

In this case, we have added 14 bits of redundant information to a 42-bit message

Cyclic Redundancy Check

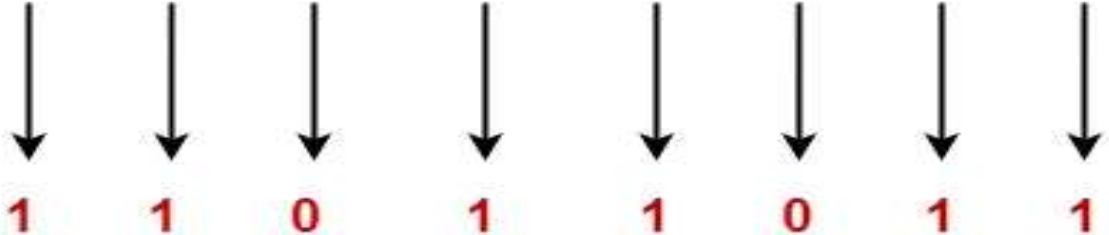


- Cyclic Redundancy Check (CRC)
 - ▣ An error detection method
 - ▣ Based on binary division
- CRC Generator
 - ▣ An algebraic polynomial represented as a bit pattern
 - ▣ Bit pattern is obtained from the CRC generator using the following rule
 - The power of each term gives the position of the bit and the coefficient gives the value of the bit

Cont...

□ Example

- Consider the CRC generator is $x^7 + x^6 + x^4 + x^3 + x + 1$
- The corresponding binary pattern is obtained as

$$1x^7 + 1x^6 + 0x^5 + 1x^4 + 1x^3 + 0x^2 + 1x^1 + 1x^0$$


1 1 0 1 1 0 1 1

- IEEE 802.3 uses 32 bits polynomial

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

Cont...

□ Properties of CRC Generator

- ▣ The algebraic polynomial chosen as a CRC generator should satisfy at least the following rules
- ▣ Rule-01: It should not be divisible by x
 - This condition guarantees that all the burst errors of length equal to the length of polynomial are detected
- ▣ Rule-02: It should be divisible by $x+1$
 - This condition guarantees that all the burst errors affecting an odd number of bits are detected

Cont...

□ Steps Involved

▣ Step-01: Calculation of CRC At Sender Side

- A string of n 0's is appended to the data unit to be transmitted
 - Here, n is one less than the number of bits in CRC generator
- Binary division is performed of the resultant string with the CRC generator
- After division, the remainder so obtained is called as CRC
- It may be noted that CRC also consists of n bits

Cont...

- Step-02: Appending CRC To Data Unit At sender side
 - The CRC is obtained after the binary division
 - The string of n 0's appended to the data unit earlier is replaced by the CRC remainder
- Step-03: Transmission To Receiver
 - The newly formed code word (Original data + CRC) is transmitted to the receiver

Cont...

□ Step-04: Checking at Receiver Side

- The transmitted code word is received
- The received code word is divided with the same CRC generator
- On division, the remainder so obtained is checked

□ The following two cases are possible

- Case-01: Remainder = 0
 - Receiver assumes that no error occurred in the data during the transmission.
 - Receiver accepts the data.

Cont...



□ Case-02: Remainder $\neq 0$

- Receiver assumes that some error occurred in the data during the transmission
- Receiver rejects the data and asks the sender for retransmission

Cont...

□ Example:

original message
1010000

1 0 1 0 0 0 0

Cont...

□ Example:

original message
1 0 1 0 0 0 0

Generator polynomial
 x^3+1
 $1 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$
CRC generator
1 0 0 1 4-bit

Cont...

□ Example:

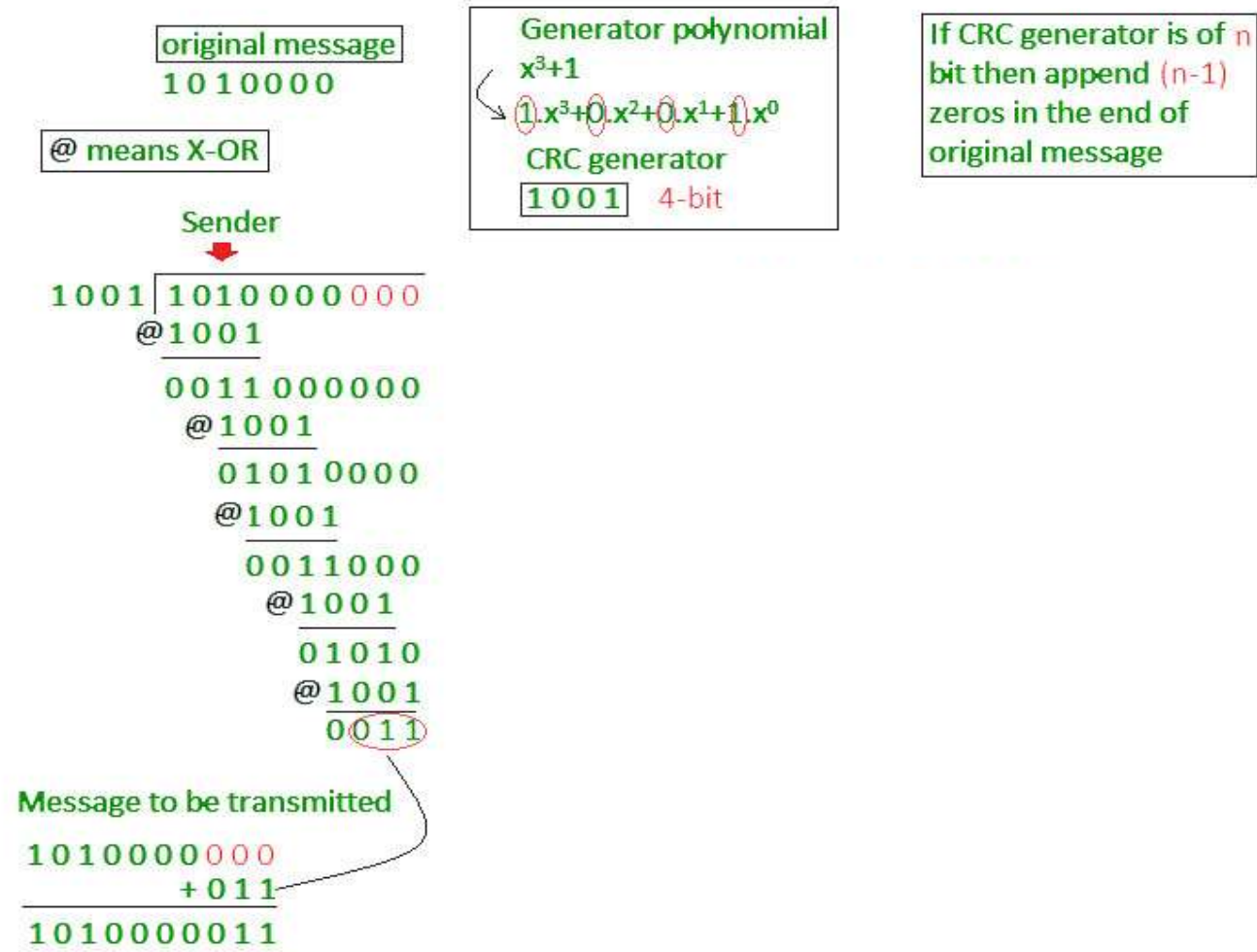
original message
1 0 1 0 0 0

Generator polynomial
 x^3+1
 $1 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$
CRC generator
1 0 0 1 4-bit

If CRC generator is of n bit then append $(n-1)$ zeros in the end of original message

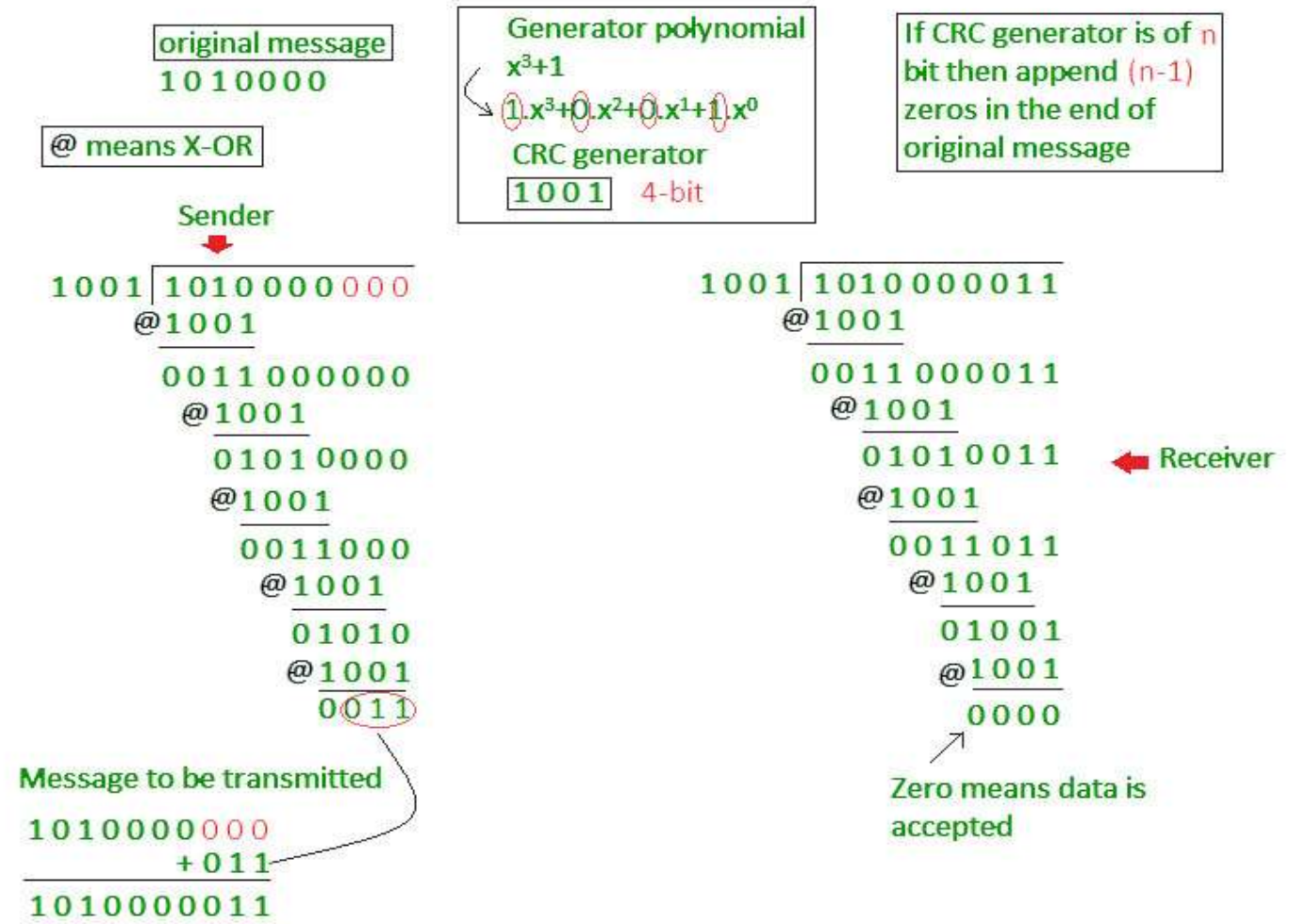
Cont...

□ Example:



Cont...

□ Example:

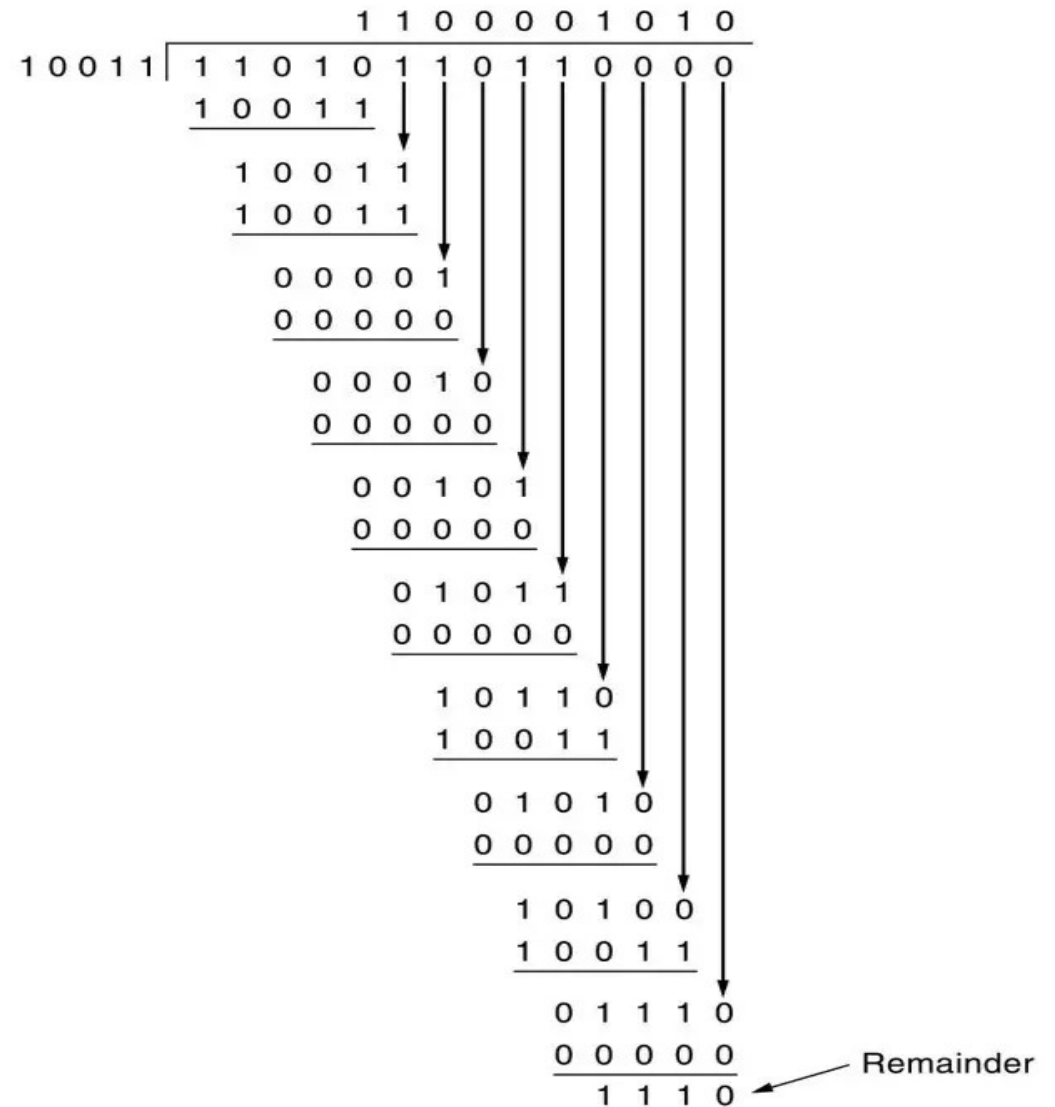


Cont...

- Problem 1: A bit stream 1101011011 is transmitted using the standard CRC method. The generator polynomial is $x^4 + x + 1$. What is the actual bit string transmitted?
- Solution:
 - ▣ The generator polynomial $G(x) = x^4 + x + 1$ is encoded as 10011
 - ▣ Clearly, the generator polynomial consists of 5 bits
 - ▣ So, a string of 4 zeroes is appended to the bit stream to be transmitted
 - ▣ The resulting bit stream is 11010110110000

Cont...

- Now, the binary division is performed as:



Cont...

- Problem-02: A bit stream 10011101 is transmitted using the standard CRC method. The generator polynomial is x^3+1 . What is the actual bit string transmitted? Suppose the third bit from the left is inverted during transmission. How will receiver detect this error?
- Solution:
 - ▣ The generator polynomial $G(x) = x^3+1$ is encoded as 1001.
 - ▣ Clearly, the generator polynomial consists of 4 bits.
 - ▣ So, a string of 3 zeroes is appended to the bit stream to be transmitted.
 - ▣ The resulting bit stream is 10011101000.

Cont...

- From here, CRC = 100.
- The code word to be transmitted is obtained by replacing the last 3 zeroes of 10011101000 with the CRC.
- Thus, the code word transmitted to the receiver = 10011101100.

$$\begin{array}{r} 10001100 \\ 1001 \overline{) 10011101000} \\ \underline{1001} \\ 00001 \\ \underline{0000} \\ 00011 \\ \underline{0000} \\ 00110 \\ \underline{0000} \\ 01101 \\ \underline{1001} \\ 01000 \\ \underline{1001} \\ 00010 \\ \underline{0000} \\ 00100 \\ \underline{0000} \\ 0100 \end{array} \leftarrow \text{CRC}$$

Cont...

- According to the question,
 - ▣ Third bit from the left gets inverted during transmission.
 - ▣ So, the bit stream received by the receiver = 10111101100.
- Receiver receives the bit stream = 10111101100.
- Receiver performs the binary division with the same generator polynomial as:

$$\begin{array}{r}
 1001 \overline{) 10101000} \\
 \underline{1001} \\
 00101 \\
 \underline{0000} \\
 01011 \\
 \underline{1001} \\
 00100 \\
 \underline{0000} \\
 01001 \\
 \underline{1001} \\
 00001 \\
 \underline{0000} \\
 00010 \\
 \underline{0000} \\
 00100 \\
 \underline{0000} \\
 0100 \leftarrow \text{Remainder}
 \end{array}$$

Checksum

- Checksum is an error detection method
- Basic Idea
 - ▣ Add up all the words that are transmitted and then transmit the result of that sum. The result is the checksum.
 - ▣ The receiver performs the same calculation on the received data and compares the result with the received checksum.
 - ▣ If any transmitted data, including the checksum itself, is corrupted, then the results will not match, so the receiver knows that an error occurred

Cont...

- Error detection using checksum method involves the following steps
 - ▣ Step-01: At sender side,
 - If m bit checksum is used, the data unit to be transmitted is divided into segments of m bits.
 - All the m bit segments are added.
 - The result of the sum is then complemented using 1's complement arithmetic.
 - The value so obtained is called as checksum.

Cont...

- At receiver side,
 - ▣ If m bit checksum is being used, the received data unit is divided into segments of m bits.
 - ▣ All the m bit segments are added along with the checksum value.
 - ▣ The value so obtained is complemented and the result is checked.
- Then, following two cases are possible
 - ▣ Case-01: Result = 0
 - If the result is zero,
 - Receiver assumes that no error occurred in the data during the transmission.
 - Receiver accepts the data.

Cont...

□ Case-02: Result $\neq 0$

- If the result is non-zero,
- Receiver assumes that error occurred in the data during the transmission.
- Receiver discards the data and asks the sender for retransmission.

Cont...

- Example: Consider the data unit to be transmitted is 10011001111000100010010010000100. Consider 8 bit checksum is used. Show how receiver will detect the error.

- Solution:

- ▣ Step-01: At sender side,

- The given data unit is divided into segments of 8 bits as

10011001	11100010	00100100	10000100
----------	----------	----------	----------

Cont...

- Now, all the segments are added and the result is obtained as-
- $10011001 + 11100010 + 00100100 + 10000100 = 1000100011$
- Since the result consists of 10 bits, so extra 2 bits are wrapped around.
- $00100011 + 10 = 00100101$ (8 bits)
- Now, 1's complement is taken which is 11011010.
- Thus, checksum value = 11011010

Cont...

□ Step-02:

- The data along with the checksum value is transmitted to the receiver

□ Step-03: At receiver side,

- The received data unit is divided into segments of 8 bits
- All the segments along with the checksum value are added
- Sum of all segments + Checksum value = $00100101 + 11011010 = 11111111$
- Complemented value = 00000000
- Since the result is 0, receiver assumes no error occurred in the data and therefore accepts it

Cont...

□ Important Notes

- Consider while adding the m bit segments, the result obtained consists of more than m bits. Then, wrap around the extra bits and add to the result so that checksum value consists of m bits.
- While calculating the checksum, if checksum value is needed, then assume it to be zero. After calculating the checksum value, substitute the checksum value in the checksum field. This will be required during checksum calculation of IP Header, TCP Header and UDP Header.
- The checksum is used in the internet by several protocols although not at the data link layer.

Cont...

□ Problem: Calculate the checksum value of 10010011100100111001100001001101 of 16 bit segment is. Consider 16 bit checksum.

□ 1010101000011111

□ 1011111000100101

□ 1101010000011110

□ 1101010000111111