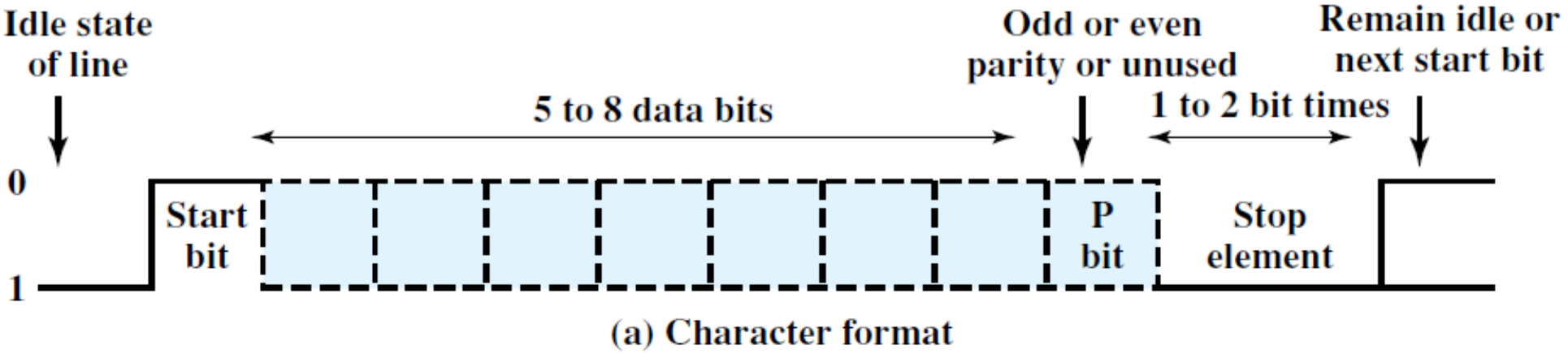
Synchronization

- The transmission of a stream of bits from one device to another across a transmission link involves cooperation and agreement between the two sides. This is the most fundamental requirement and is called synchronization.
- The receiver must know the rate at which bits are being received so that it can sample the line at appropriate intervals to determine the value of each received bit.
- The timing (rate, duration, spacing) of these bits must be the same for transmitter and receiver.
- Two techniques are in common use for this purpose.
  - Asynchronous transmission
  - Synchronous transmission

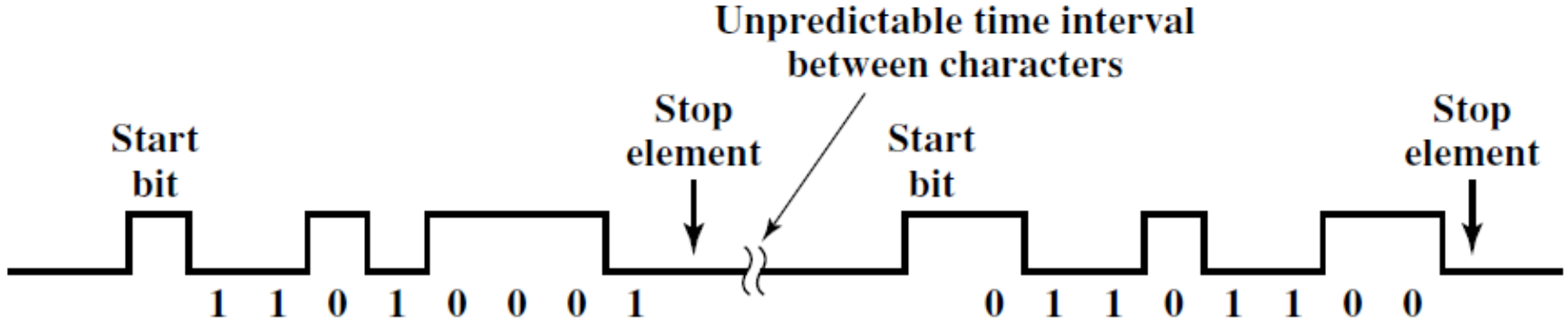
Asynchronous Transmission

- Data are transmitted one character at a time , where each character is 5 – 8 bits in length.
- Timing or synchronization is maintained within each character.
- When no character is being transmitted, the line between transmitter and receiver is in an idle state. The definition of idle is equivalent to the signaling element for binary 1.
- The beginning of a character is signaled by a start bit with a value of binary 0.
- This is followed by the 5 to 8 bits that actually make up the character. The bits of the character are transmitted beginning with the least significant bit.
- The parity bit is set by the transmitter such that the total number of ones in the character, including the parity bit, is even (even parity) or odd (odd parity), depending on the convention being used. The receiver uses this bit for error detection.
- The final element is a stop element, which is a binary 1.
- A minimum length for the stop element is specified, and this is usually 1, 1.5, or 2 times the duration of an ordinary bit.
- Because the stop element is the same as the idle state, the transmitter will continue to transmit the stop element until it is ready to send the next character.

Asynchronous Transmission



(a) Character format



(b) 8-bit asynchronous character stream

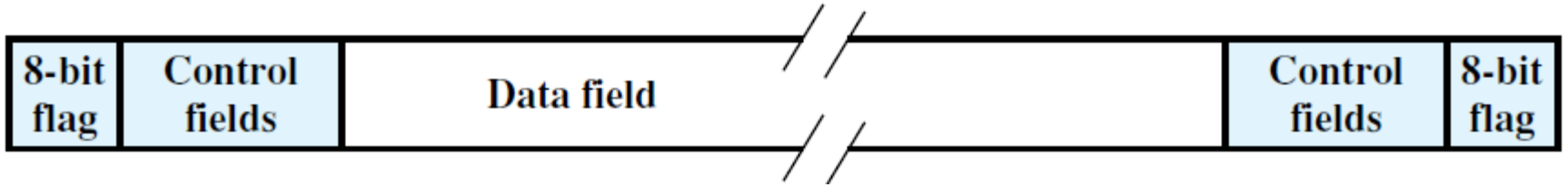
Asynchronous Transmission

- Asynchronous transmission is simple and cheap but requires an overhead of two to three bits per character.
- For example, for an 8 bit character with no parity bit, using a 1 bit long stop element, two out of every ten bits convey no information but are there merely for synchronization; thus the overhead is 20%.

Synchronous Transmission

- With synchronous transmission, a block of bits is transmitted in a steady stream without start and stop codes. The block may be many bits in length.
- With synchronous transmission, the receiver is required to determine the beginning and end of a block of data.
- To achieve this, each block begins with a preamble bit pattern and generally ends with a postamble bit pattern.
- In addition, other bits are added to the block that convey control information used in the data link control procedures.
- The data plus preamble, postamble, and control information are called a frame.

Synchronous Transmission

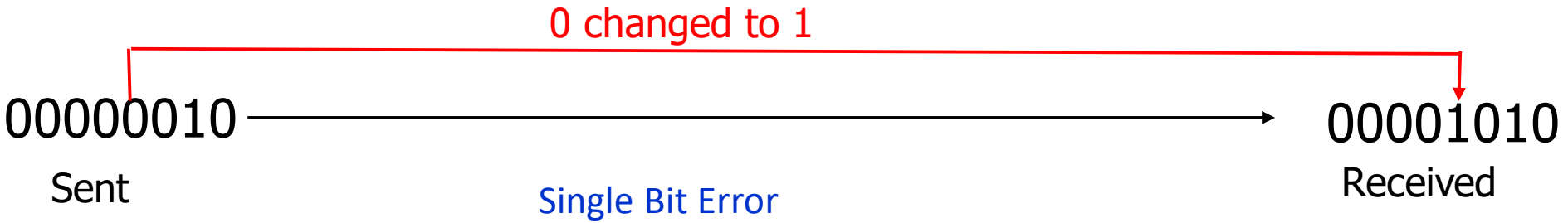


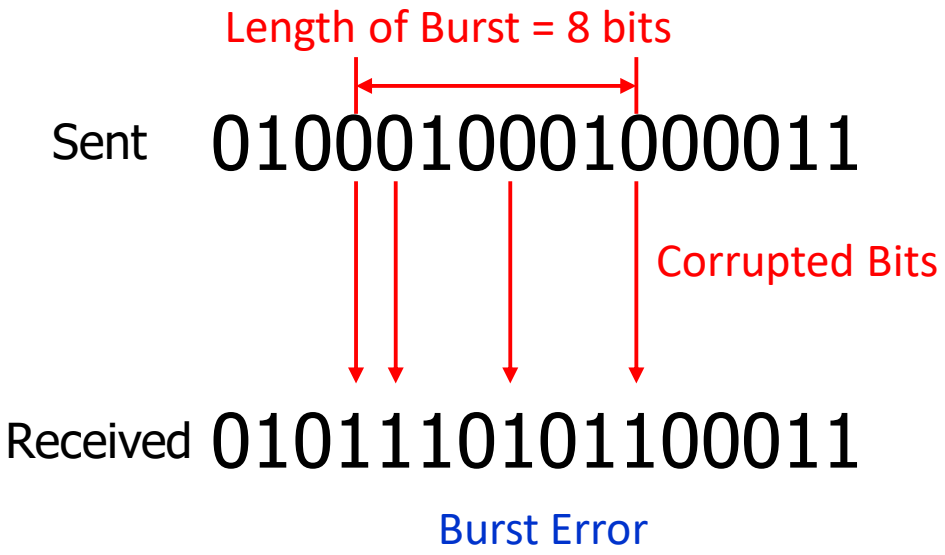
- The frame starts with a preamble called a flag, which is 8 bits long. The same flag is used as a postamble.
- The receiver looks for the occurrence of the flag pattern to signal the start of a frame.
- This is followed by some number of control fields (containing data link control protocol information), then a data field (variable length for most protocols), more control fields, and finally the flag is repeated.
- Synchronous transmission is far more efficient than asynchronous.

Types of Errors

- In digital transmission systems, an error occurs when a bit is altered between transmission and reception; that is, a binary 1 is transmitted and a binary 0 is received, or a binary 0 is transmitted and a binary 1 is received.
- Two general types of errors can occur:
  - Single bit errors - A single bit error is an isolated error condition that alters one bit but does not affect nearby bits.
  - Burst errors - A burst error of length B is a contiguous sequence of B bits in which the first and last bits and any number of intermediate bits are received in error.

Types of Errors





Types of Errors

- Single bit errors are the least likely type of error in serial data transmission.
- A burst error does not necessarily mean that the errors occur in consecutive bits.
- The length of the burst is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not have been corrupted.
- A burst error is more likely to occur than a single bit error. The duration of noise is normally longer than the duration of 1 bit, which means that when noise affects data, it affects a set of bits.
- The number of bits affected depends on the data rate and duration of noise.
- Example: when a wireless transmitter transmits at 11 Mbps and an interference burst of 200 μs occurs, 2200 bits are affected by the burst.

Detection Vs Correction

- In error detection, we are looking only to see if any error has occurred. The answer is a simple yes or no. We are not interested in the number of errors. A single bit error is the same for us as a burst error.
- In error correction, we need to know the exact number of bits that are corrupted and more importantly, their location in the message. The number of the errors and the size of the message are important factors.

Redundancy

- The central concept in detecting or correcting errors is redundancy.
- To be able to detect or correct errors, we need to send some extra(redundant) bits with our data.
- These redundant bits are added by the sender and removed by the receiver.
- Their presence allows the receiver to detect or correct corrupted bits.

Coding

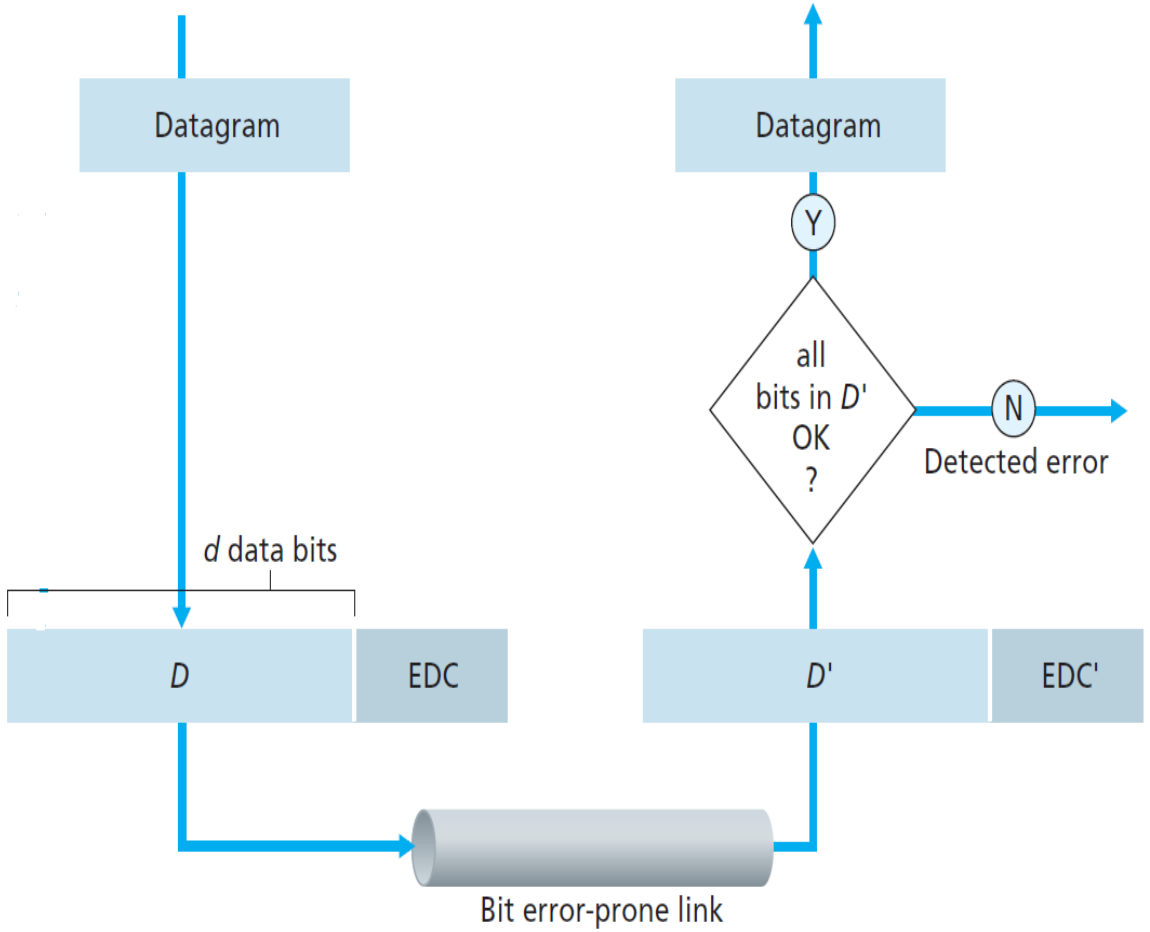
- Redundancy is achieved through various coding schemes.
- The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits.
- The receiver checks the relationships between the two sets of bits to detect or correct the errors.
- The ratio of redundant bits to the data bits is important factor in any coding scheme.

Forward Error Correction Vs Retransmission

- Two main methods of error correction are:
  - Forward error correction(FEC) is the process in which the receiver tries to guess the message by using redundant bits.
  - Correction by retransmission is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message. Resending is repeated until a message arrives that the receiver believes is error free.

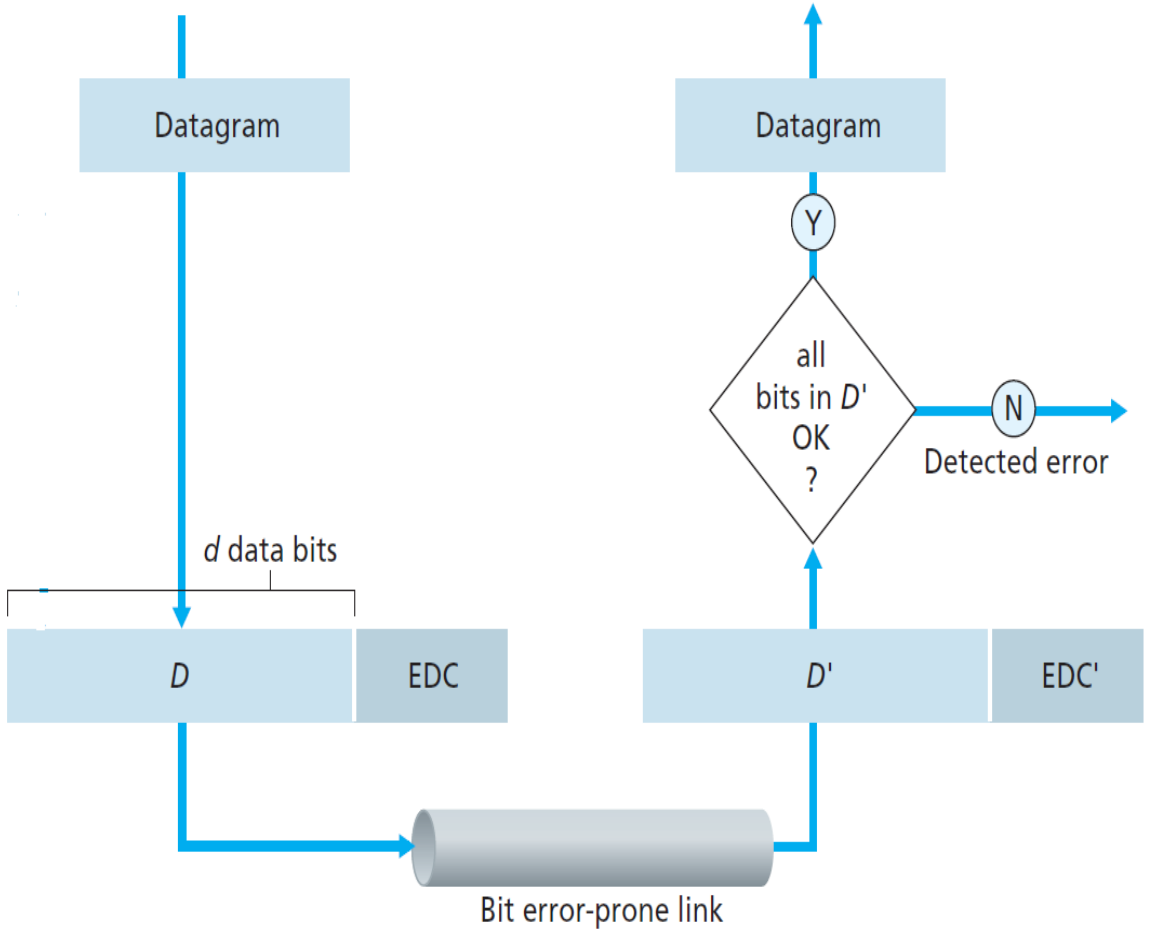
How it is done?

- At the sending node, data, D, to be protected against bit errors is augmented with error detection and correction bits (EDC).
- EDC is the function of transmitted bits.
- Typically, for a data block of k bits, the error detecting algorithm yields an error detecting code of $n-k$ bits, where $(n-k) < k$.
- The error detecting code, also referred to as the check bits, is appended to the data block to produce a frame of n bits, which is then transmitted.



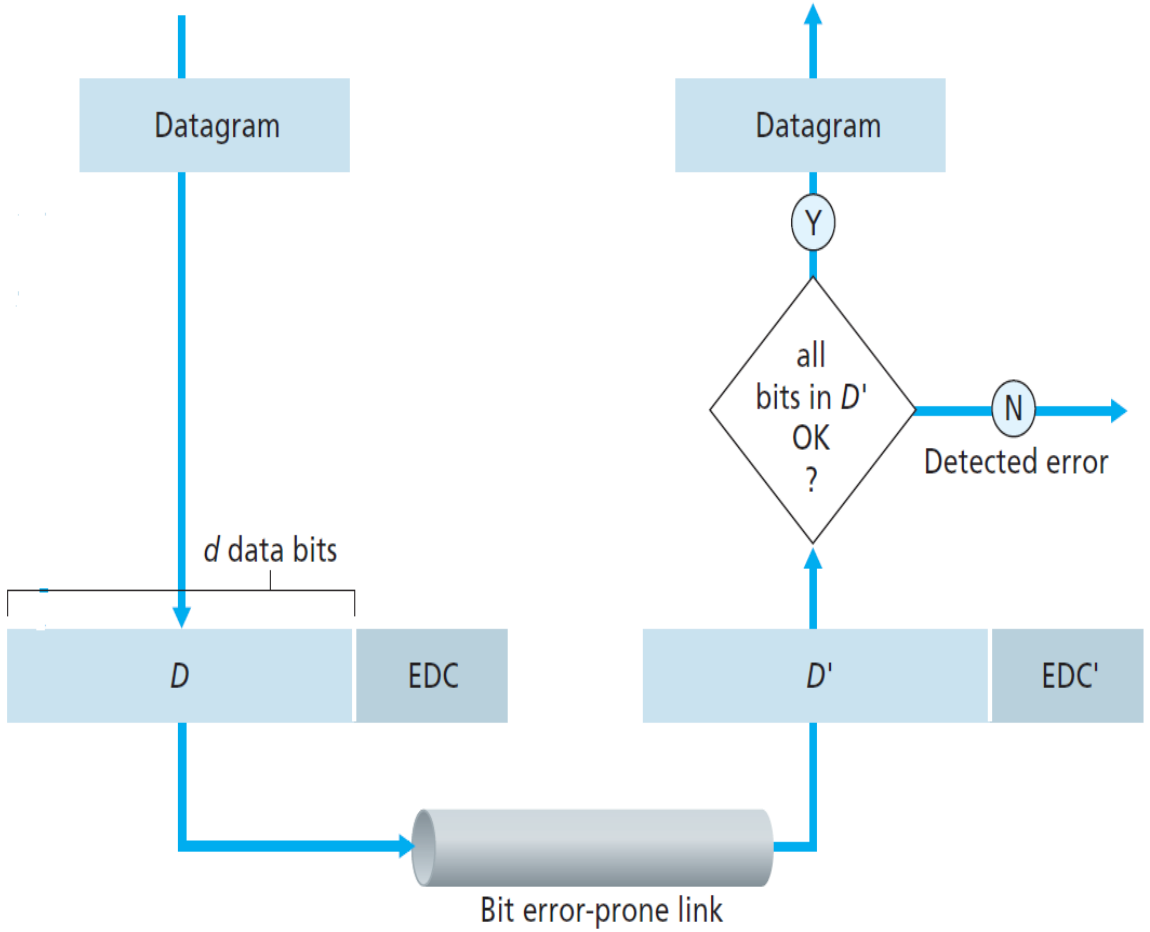
How it is done?

- Both $D(k$ bits) and $EDC(n-k$ bits) are sent to the receiving node in a link level frame.
- At the receiving node, a sequence of bits, $\acute{D}$ ($\acute{k}$ bits) and $E\acute{D}C$ ($(n-\acute{k})$ bits)is received.
- $\acute{D}$ and $E\acute{D}C$ may differ from the original D and EDC as a result of in transit bit flips.



How it is done?

- The receiver performs the same error detecting calculation on the data bits and compares this value with the value of the incoming error detecting code.
- A detected error occurs if and only if there is a mismatch.



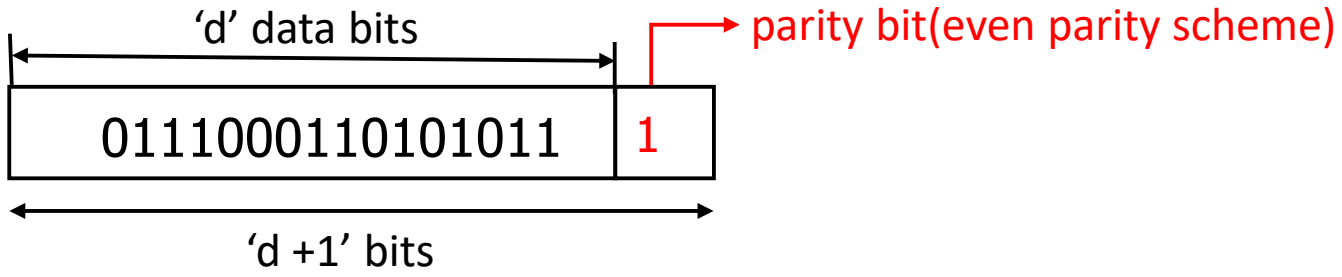
Error Detection Techniques

- Three techniques for detecting errors in the transmitted data are:
  - Parity checks
  - Checksum methods
  - Cyclic redundancy checks (CRC)

Parity Checks

- The simplest error detecting scheme is to append a single parity bit to the end of a block of data.
- In an even parity scheme, the sender simply includes one additional bit and chooses its value such that the total number of 1s in the $d + 1$ bits (the original information plus a parity bit) is even.
- For odd parity schemes, the parity bit value is chosen such that there is an odd number of 1s.

Parity Checks



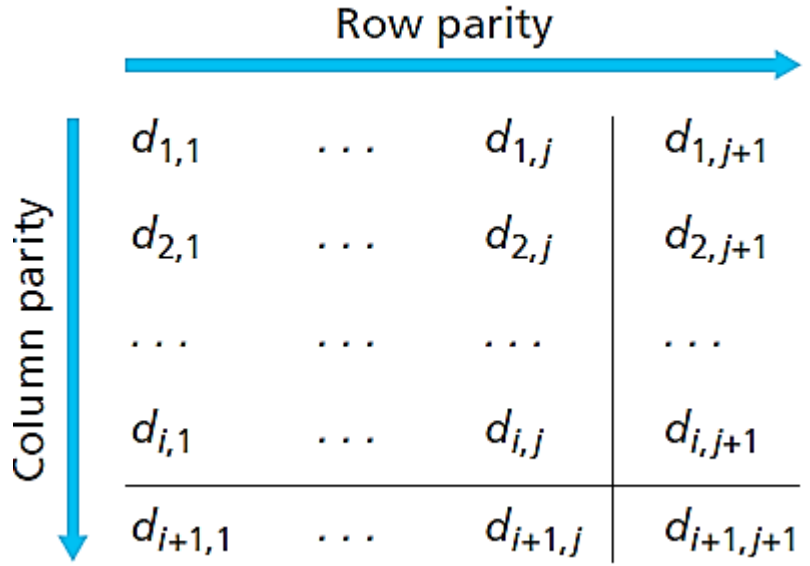
- Receiver operation is also simple with a single parity bit.
- The receiver need only count the number of 1s in the received $d + 1$ bits.
- If an odd number of 1 valued bits are found with an even parity scheme, the receiver knows that at least one bit error has occurred. More precisely, it knows that some odd number of bit errors have occurred.
- But when even no. of bits error occurred, it is undetectable.

Properties of Parity Checks

- Properties of parity check codes are:
  - All odd numbers of bit errors are detected.
  - All even numbers of bit errors are not detected.
- Parity check codes have traditionally been used on serial interfaces, e.g. RS-232, where a parity bit has been appended to each byte.

Two Dimensional Parity Checks

- The d bits in D are divided into i rows and j columns.
- A parity value is computed for each row and for each column.
- The resulting $i + j + 1$ parity bits comprise the link layer frame's error detection bits.



Two Dimensional Parity Checks

1100111 1011101 0111001 0101001 ← Original data

1100111	1
1011101	1
0111001	0
0101001	1
0101010	1

Sender Side

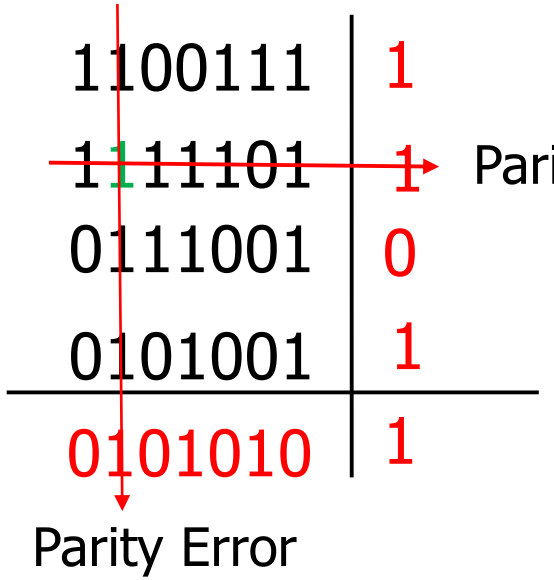
Even Parity scheme used

Transmitted Frame

11001111 10111011 01110010 01010011 01010101

Two Dimensional Parity Checks

Receiver Side

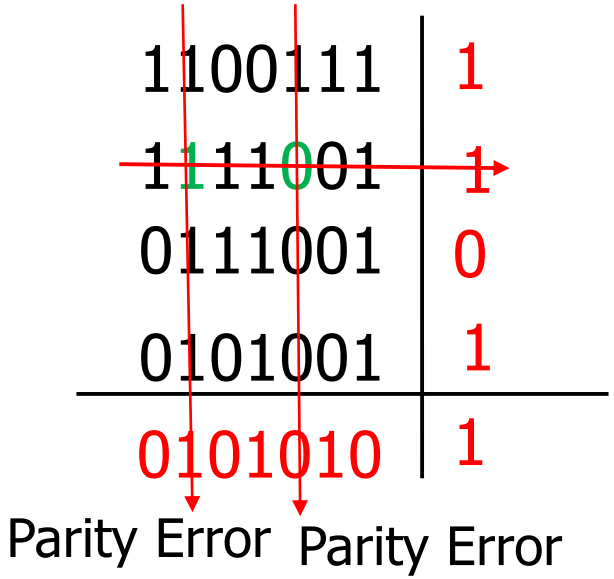


*Bit in position (2,2) is switched to 1

*A single error in the parity bits is also detectable and correctable.

Two Dimensional Parity Checks

Receiver Side



*Bit in position (2,2) is switched to 1 and bit in position (2,5)is switched to 0

*Two-dimensional parity can also detect (but not correct!) any combination of two errors in a packet.

Checksum Methods

- On the sender side: The checksum generator sub divides the data unit into equal segments of 'n' bits(usually 16).
- These segments are added using 1's complement arithmetic in such a way that total is also 'n' bits.
- The total or sum is then complemented and appended to the end of the data as redundancy bits.
- These redundancy bits are called checksum field.
- The extended data unit is transmitted across the network.

Checksum Methods

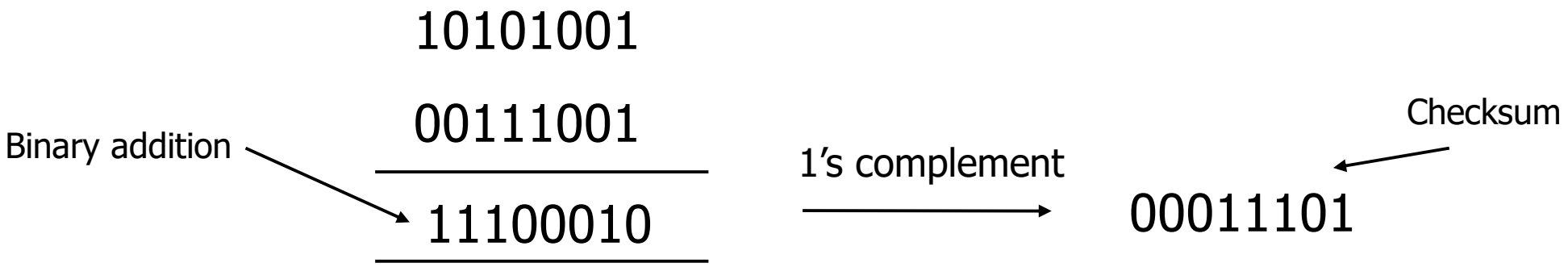
- The receiver sub divides the data unit adds all the segments and complements the results.
- If the encoded data is intact, the total values found by adding the data segments and the checksum field should be zero.
- If the result is not zero , the packet contains error and receiver rejects it.
- The Internet checksum is based on this approach.

Checksum Methods

10101001 00111001 ← Original data

Length of checksum = 8 bits

Sender Side:

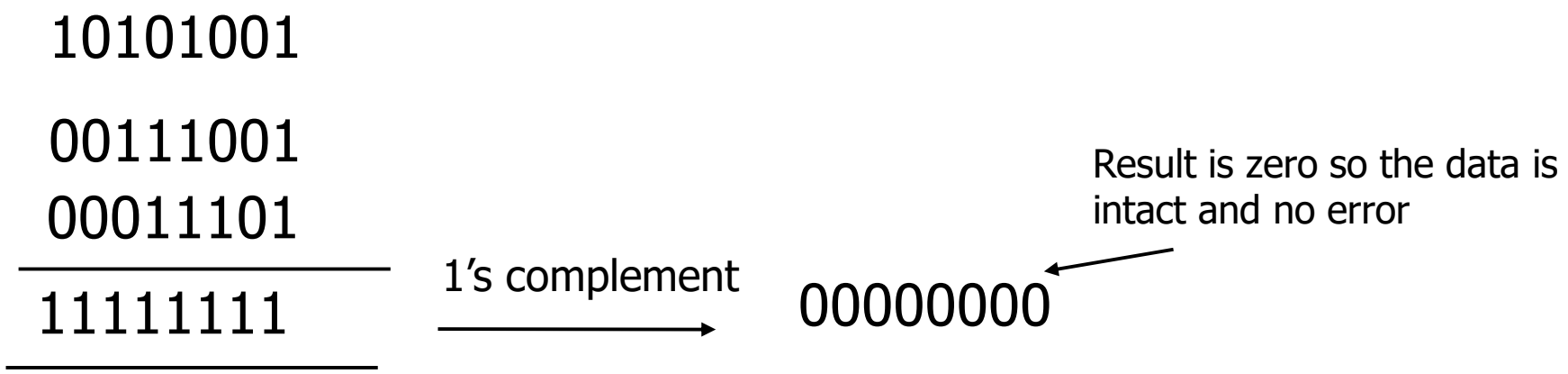


Encoded Word 10101001 00111001 00011101

Checksum Methods

Received Word 10101001 00111001 00011101

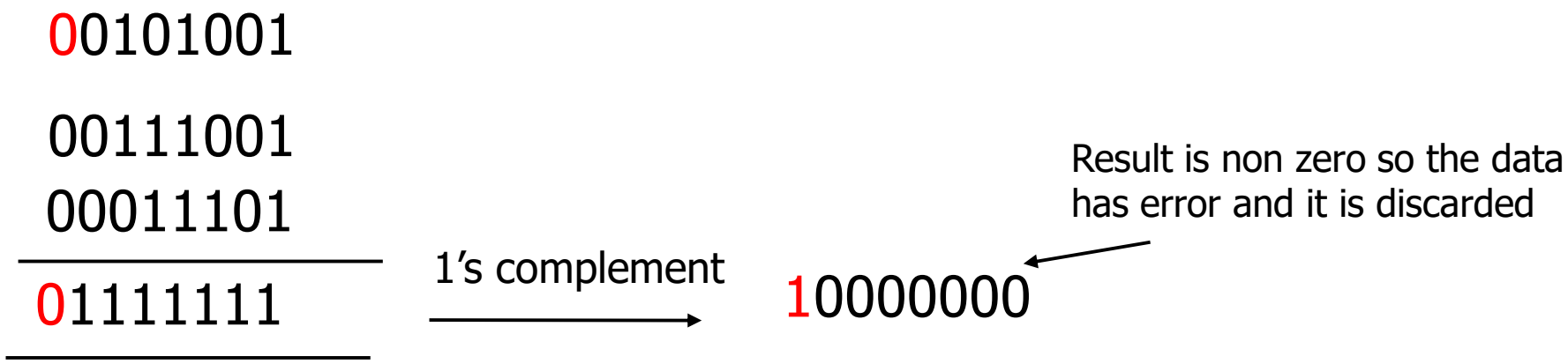
Receiver Side:



Checksum Methods

Now let us introduce error in first bit

Receiver Side:



Checksum Method - Properties

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

Transmitted Word	10101001	00111001	00011101
Received Word	00101001	10111001	00011101

Receiver Side:

00101001
10111001
00011101

11111111

1's complement →

00000000

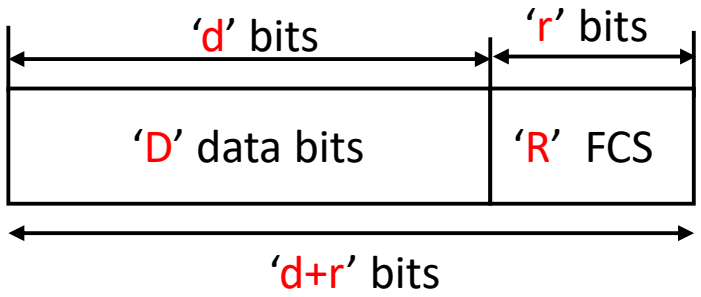
Result is zero no error is detected though error was present

Cyclic Redundancy Check

- Most powerful and widely used error detection technique and it is based on binary division.
- CRC codes are known as polynomials.
- Consider the d bit piece of data, D, that the sending node wants to send to the receiving node.
- The sender and receiver must first agree on an $r + 1$ bit pattern, known as a generator, which we will denote as G.
- For a given piece of data, D, the sender will choose r additional bits(also known as frame check sequence), R, and append them to D such that the resulting $d + r$ bit pattern is exactly divisible by G (i.e., has no remainder) using modulo 2 arithmetic.
- All CRC calculations are done in modulo 2 arithmetic without carries in addition or borrows in subtraction. This means that addition and subtraction are identical, and both are equivalent to the bitwise exclusive or (XOR) of the operands.

Cyclic Redundancy Check

- The process of error checking with CRCs is thus simple: The receiver divides the $d + r$ received bits by G.
- If the remainder is nonzero, the receiver knows that an error has occurred; otherwise the data is accepted as being correct.



- Mathematical formula: $D.2^r \text{ XOR R}$

$D.2^r$ → Left Shift 'D' by 'r' bits to yield 'd+r' bits

- In order to calculate R , we divide $D.2^r$ by G

Cyclic Redundancy Check

- Given:

  Message ‘D’ = 1010001101 (10 Bits) = ‘d’ bits

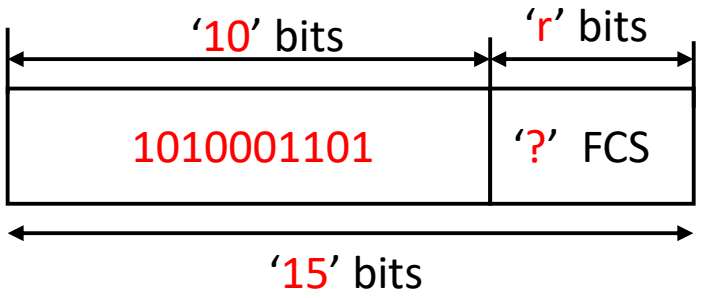
  Generator ‘G’ = 110101 (6 Bits)

  Degree of G = $1.X^5 + 1.X^4 + 0.X^3 + 1.X^2 + 0.X^1 + 1.X^0$ = 5 = ‘r’ bits

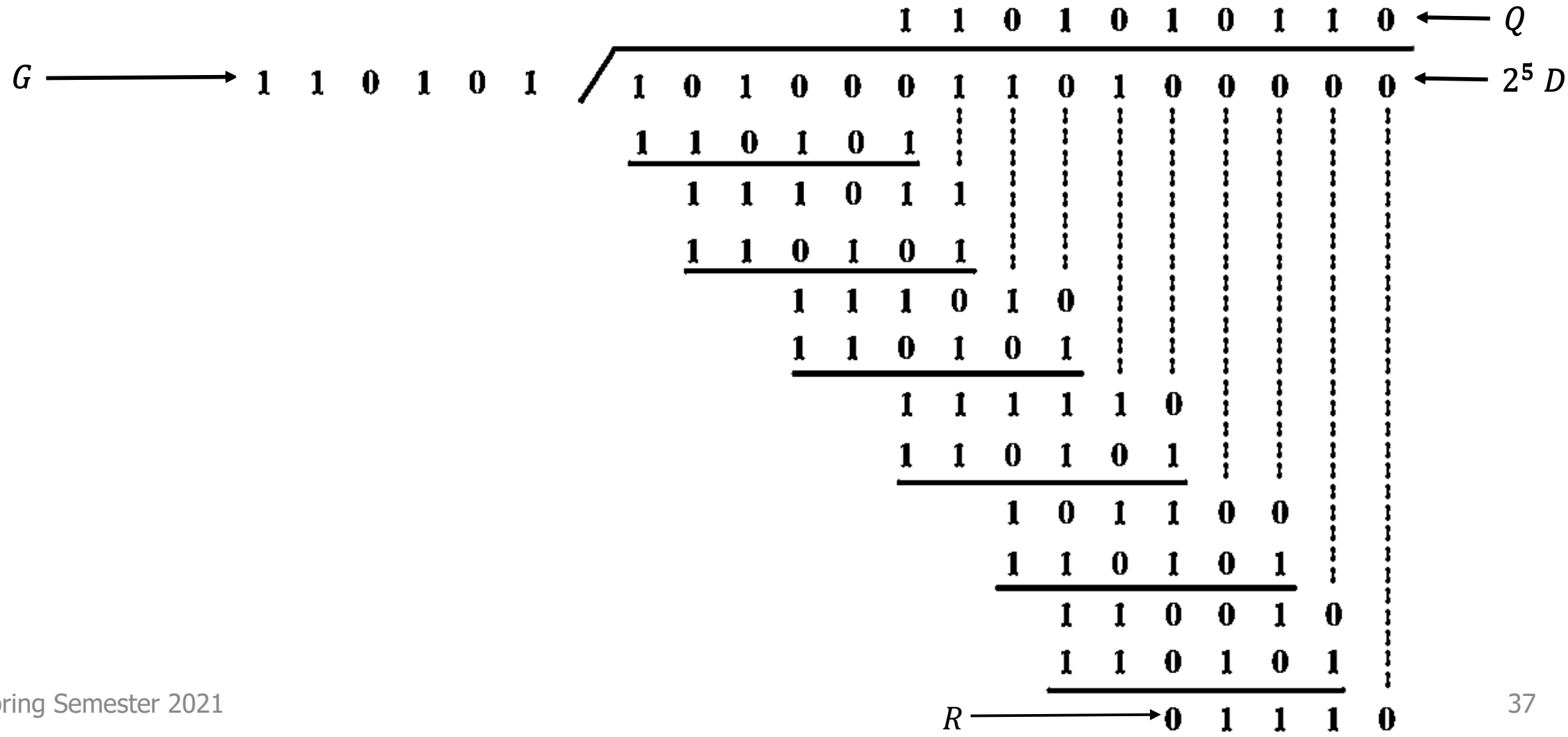
  FCS to be calculated? = ‘R’ = ?

  ‘d + r’ = 10 + 5 = 15 bits

  The message is multiplied 2^5 by yielding 1010001101**00000**

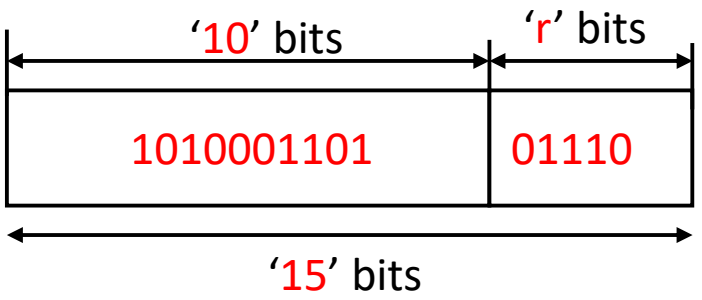


Cyclic Redundancy Check



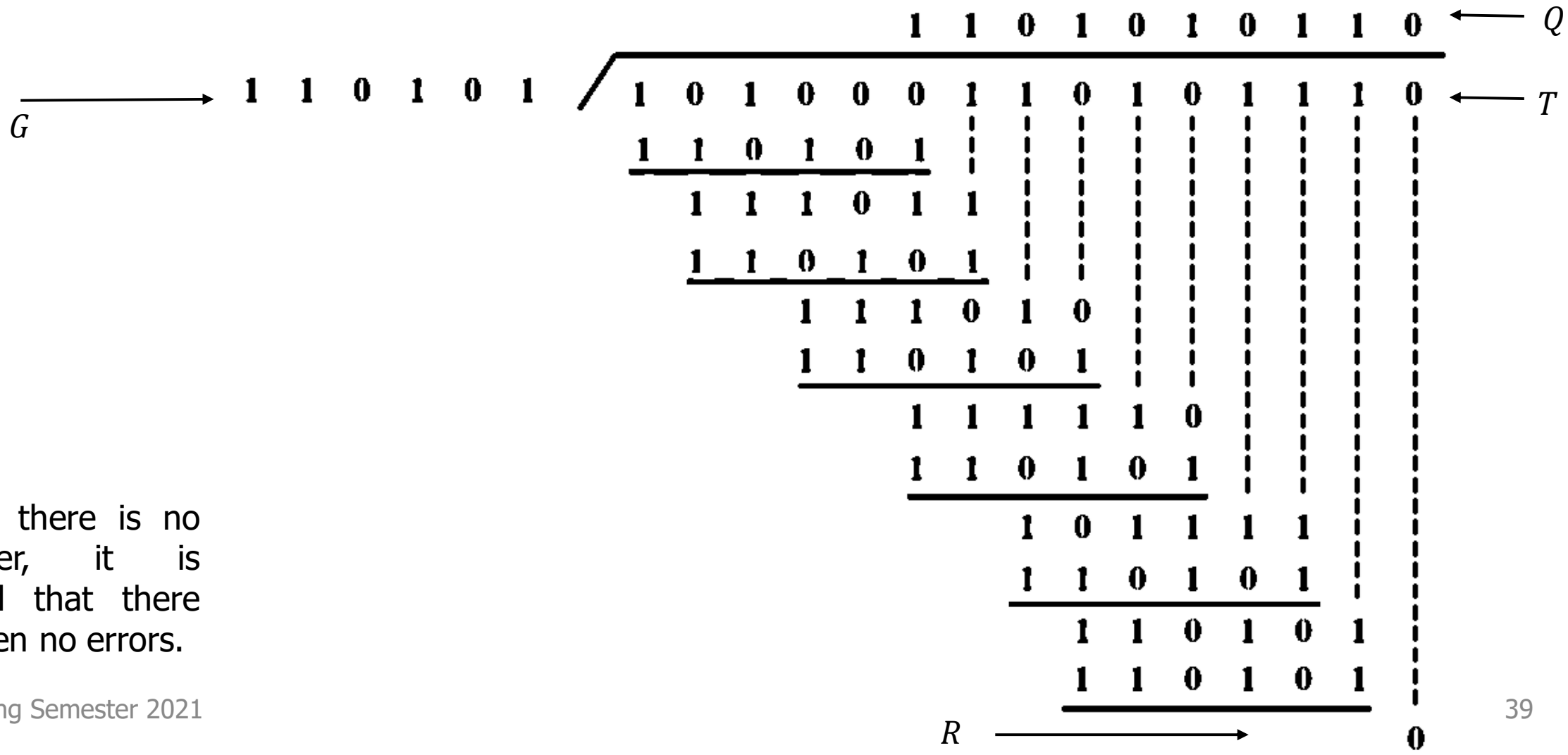
Cyclic Redundancy Check

- The remainder is added to $2^5 D$ to give $T = 1010001101\mathbf{01110}$ which is transmitted.



- If there are no errors, the receiver receives T intact. The received frame is divided by G.

Cyclic Redundancy Check



Because there is no remainder, it is assumed that there have been no errors.

CRC – Some Standard Polynomials

- A second way of viewing the CRC process is to express all values as polynomials in a dummy variable X, with binary coefficients. The coefficients correspond to the bits in the binary number.
- Thus for D = 110011 we have $\mathrm{D}(X) = X^5 + X^4 + X + 1$, and for G= 11001 we $\mathrm{G}(X) = X^4 + X^3 + 1$
- Arithmetic operations are again modulo 2.
- International standards have been defined for 8-, 12-, 16-, and 32-bit generators, *G*.
- The CRC-32 32 bit standard, which has been adopted in a number of link level IEEE protocols, uses a generator of:

$$CRC - 32 = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

Cyclic Redundancy Check - Properties

- Errors are not detected when their bit pattern (taken as a polynomial) is evenly divisible by $G(X)$
- All single bit errors, if $G(X)$ has more than one nonzero term.
- Any odd number of errors, as long as $G(X)$ contains a factor $(X + 1)$
- Any burst error for which the length of the burst is less than or equal to 'r' that is, less than or equal to the length of the FCS.