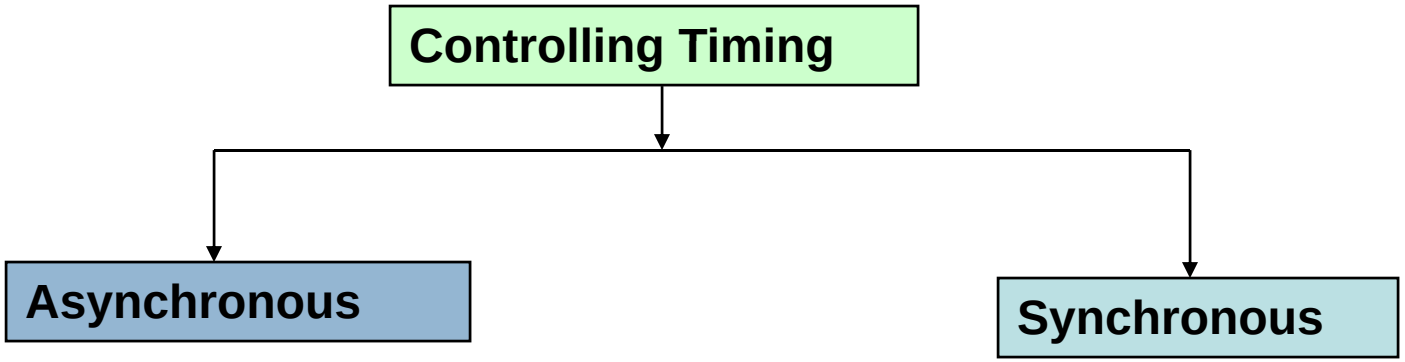
CSE 350
DATA COMMUNICATIONS

Lecture 5: Digital Data Communication Techniques

Maheen Islam

Asynchronous and Synchronous Transmission

- Timing problems require a mechanism to synchronize the transmitter and receiver
- Receiver must know—arrival time, data rate, duration of each bit
- Two solutions for controlling timing:



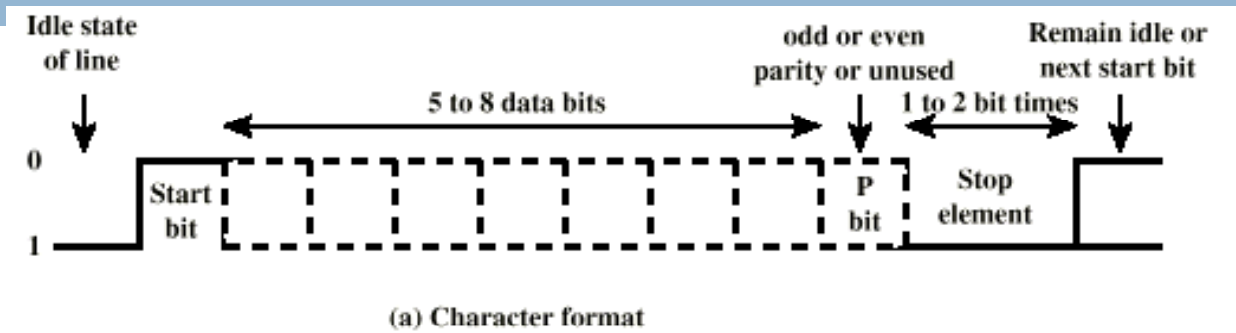
Asynchronous and Synchronous Transmission

- With serial transmission signaling elements are sent down the signal path one at a time.
- Reception of digital data involves sampling the signal once per bit time to determine the binary value.
- In order for the receiver to sample the incoming bits properly, it must know the arrival time and duration of each bit it receives.

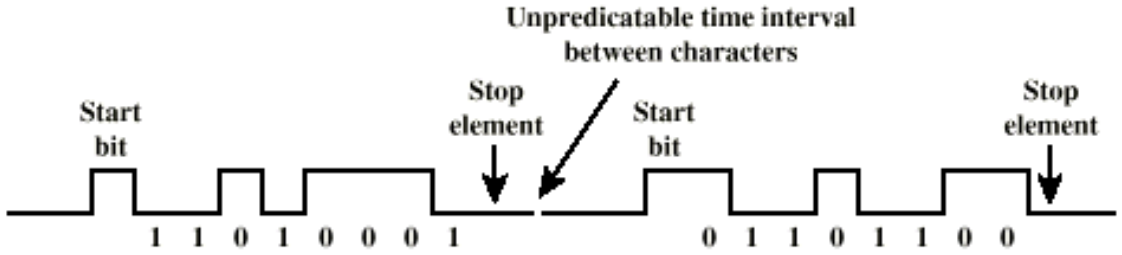
Asynchronous

- Data transmitted one character at a time
  - Each char is 5 to 8 bits in length
- Timing only needs maintaining within each character
- The receiver has the opportunity to resynchronize at the beginning of each new character

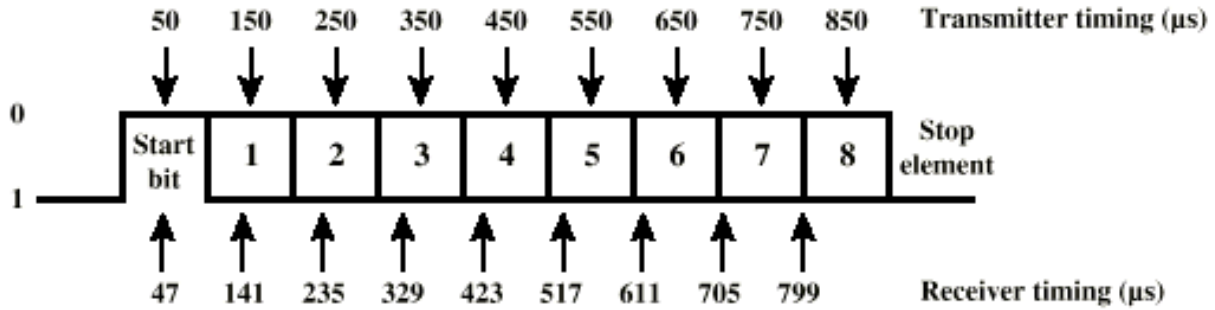
Asynchronous (diagram)



(a) Character format



(b) 8-bit asynchronous character stream



(c) Effect of timing error

Asynchronous - Behavior

- **Idle state:** When no character is being transmitted
  - Signaling element for binary 1
- The beginning of a character is signaled by a ***start bit*** with a value of binary 0.
- The bits of the character are transmitted beginning with the least significant bit.
- The receiver set a **parity bit**
- **Stop element**
  - binary 1.
  - Minimum length for the stop element is:1,1.5,2 bit duration.
  - No maximum value is specified
  - The transmitter will continue to transmit the stop element until it is ready to send the next character.

Timing Error

- Data rate:10 kbps; each bit 0.1ms or 100µs duration
- Receiver fast :6% or 6µs per bit time
- So, the receiver samples the incoming character every 94µs—error
- Results two errors—
  - The last sampled bit is incorrectly received
  - The bit count may be out of alignment (Framing error). If bit 7 is 1 and bit 8 is a 0, bit 8 could be mistaken for a start bit.
  - Frame = character + start bit +stop bit.

Pros and Cons

- Simple
- Cheap
- Overhead of 2 or 3 bits per char (~20%)
- Good for data with large gaps

Synchronous - Bit Level

- Block of data is transmitted without start or stop bits
- The block may be many bits in length
- Clocks must be synchronized
- **Techniques**

1. Provide separate clock line between transmitter and receiver
   - One side pulses the line regularly with one short pulse per bit time
   - The other side uses these pulses as a clock.

Synchronous - Bit Level

- Good over short distances
- Subject to impairments for longer distances—timing error

2. Embed clock signal in data
   - Manchester encoding
   - Carrier frequency (analog)
     - Based on the phase of the carrier

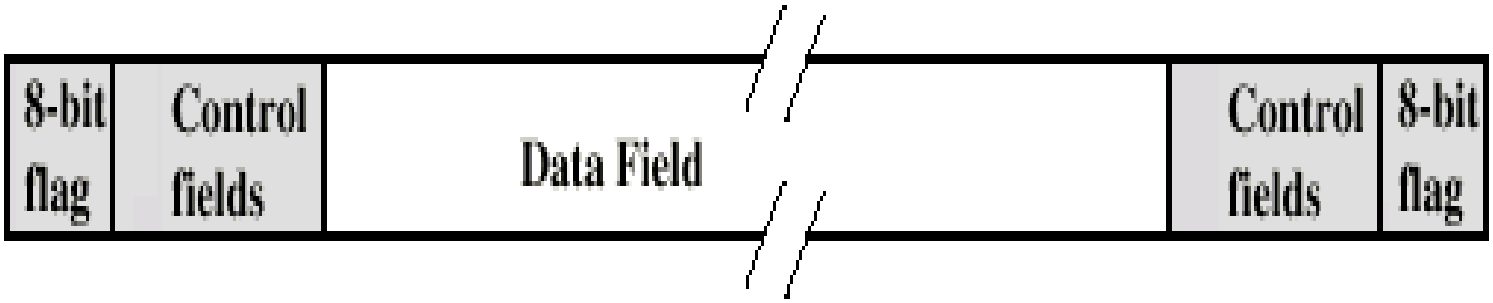
Synchronous - Block Level

- To allow the receiver to determine the beginning and end of a block of data
- Use **preamble** (called flag, 8 bits long) bit pattern – beginning of blocks
- **Postamble** (called flag, 8 bits long) bit pattern – ending of blocks.
- Control information – other bits are added to convey control information.
- data+ preamble+ postamble +control info = **frame**

Synchronous - Block Level

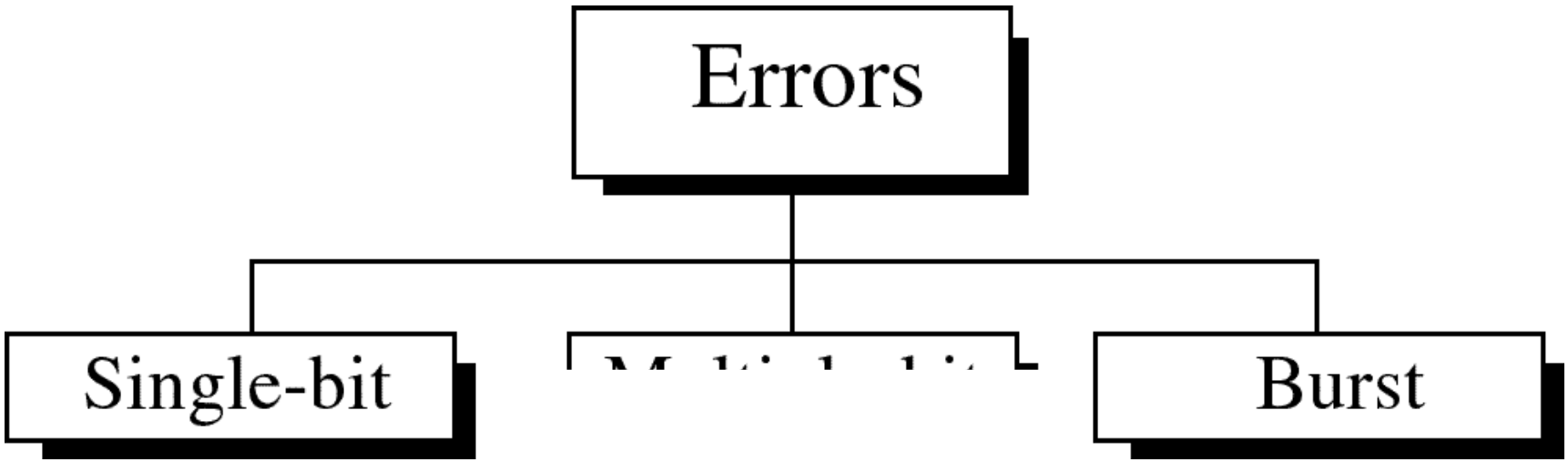
- More efficient (lower overhead) than asynchronous
  - For example one scheme contains 48 bits of control, preamble and postamble. Thus for a 1000 character block of data, each frame contains 8000 bits of data and 48bits of control information, for a percentage overhead of only 0.6%.

Synchronous - Block Diagram



Types of Errors

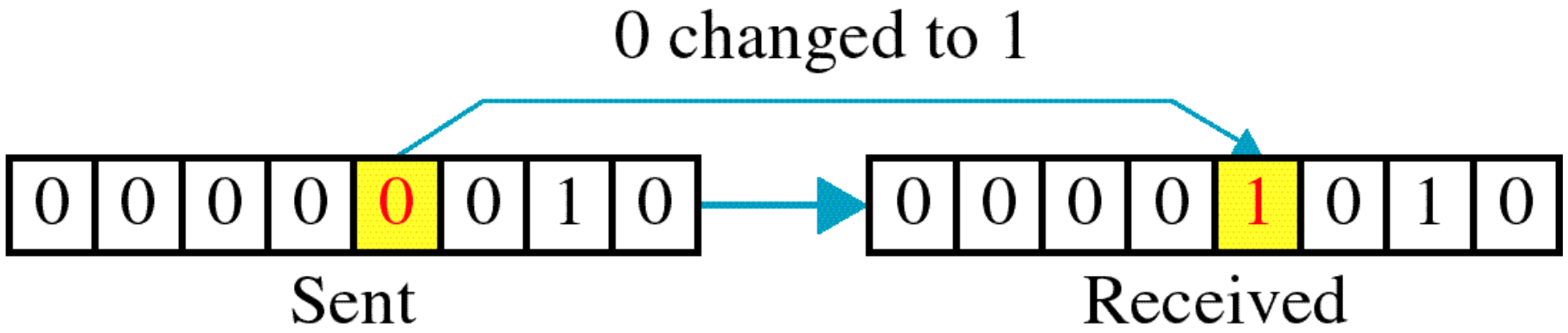
- An error occurs when a bit is altered between transmission and reception



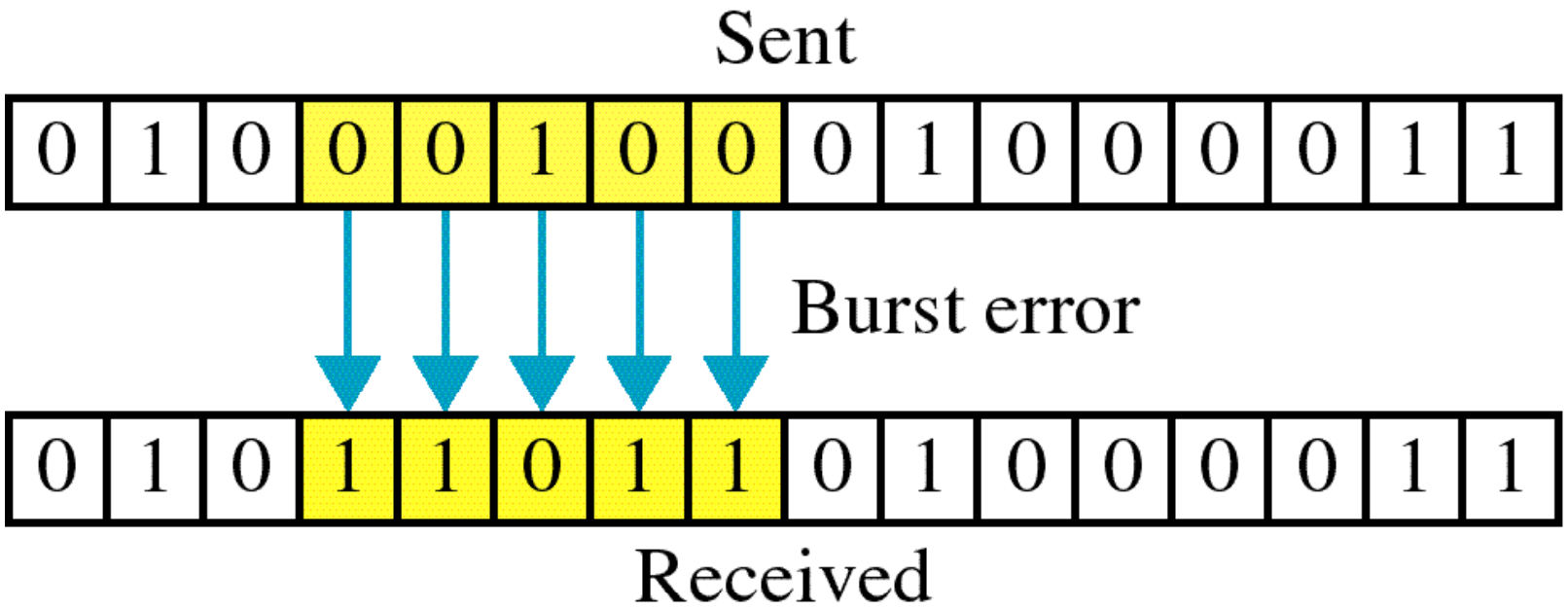
Types of Errors

- Single bit errors
  - Only one bit is altered
  - Adjacent bits are not affected
  - White noise
- Burst errors
  - Length *B*
  - Contiguous sequence of *B* bits in which first, last and any number of intermediate bits are in error
  - Or two or more bits in the data unit have changed
  - Impulse noise
  - Effect is greater at higher data rates
  - More difficult to deal with than Single bit errror.

Single-bit error



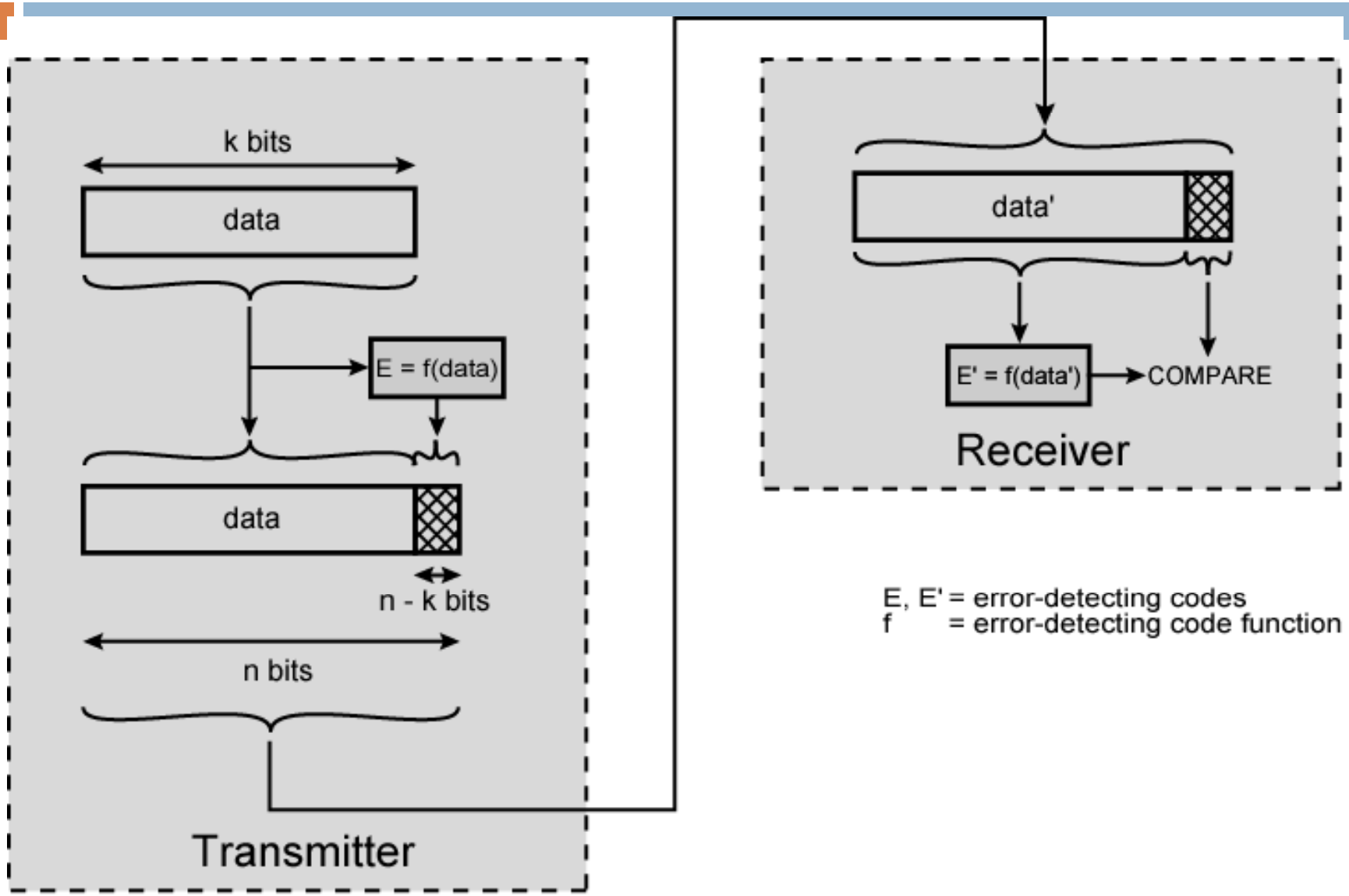
Burst error


Sent
0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1
Burst error
0 1 0 1 1 0 1 1 0 1 0 0 0 0 1 1
Received

Error Detection

- Principle error detection technique.
- Additional bits added by transmitter for error detection code with a given frame of bits.
- This code is calculated on data bits

Error Detection Process



Parity Check

- Simplest error detection scheme
- Append a parity-bit to the end of a block of data
- Value of parity bit is such that character has even (even parity) or odd (odd parity) number of ones
- Problem: Two or even number of bit errors goes undetected

Parity Example

- Character to be transmitted → "S"
  - "S" → 1010011
- Even Parity → 0 1010011
- Odd Parity → 1 1010011
- Transmission w/ one error (odd) → 11011011
  - Error detected!!
- Transmission w/ two errors (odd) → 11011111
  - No error detected!!
  - Received character is "_"

Cyclic Redundancy Check(CRC)

- The most powerful
- A sequence of redundant bits, CRC or CRC remainder is added to the end of a data unit
- The resulting data unit becomes exactly divisible by a predetermined binary number
- At the destination, the incoming data is divided by the same number
- If there is no reminder
- The data unit is assumed to be intact and accepted
- If there is reminder, data has been damaged and rejected.

Cyclic Redundancy Check (CRC)

- For block of *k* data bits, transmitter generates frame check sequence (FCS) of length *n-k*
- Transmit *n*-bit frame which is built to be exactly divisible by some number (i.e. predetermined divisor)
- Receiver divides frame by that number
  - If no remainder, assume no error

CRC described—in 3 ways

- Modulo 2 Arithmetic
- Polynomial
- Digital logic

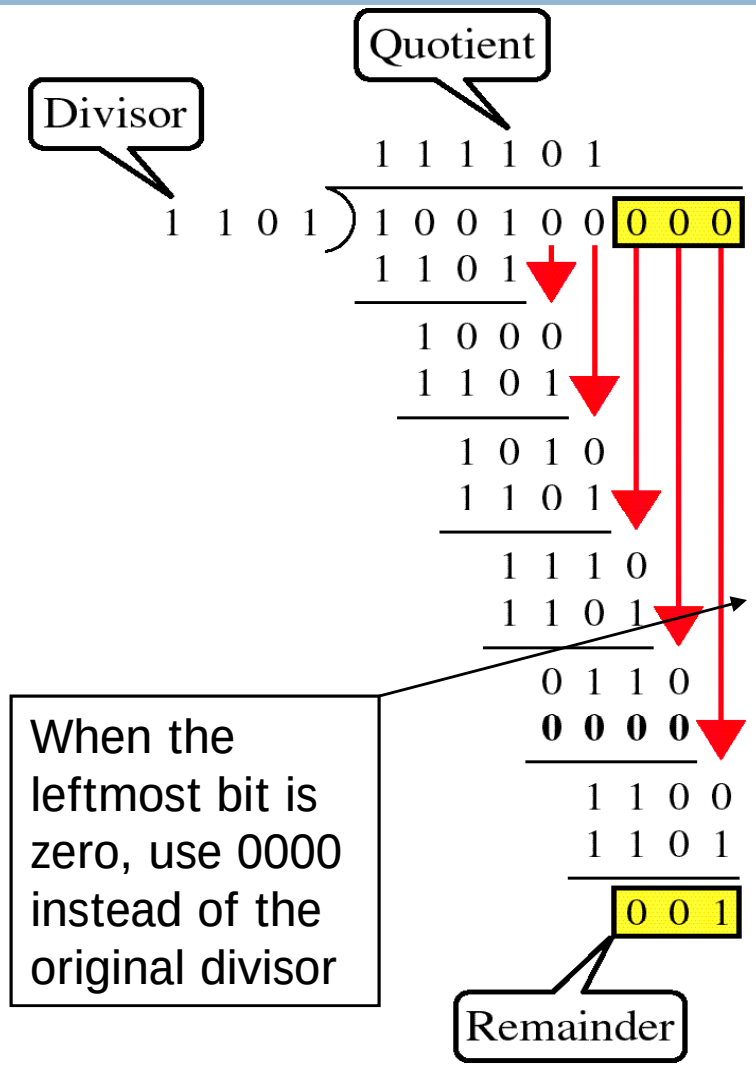
Modulo 2 Arithmetic

- Uses binary addition with no carries
- Just the XOR operation

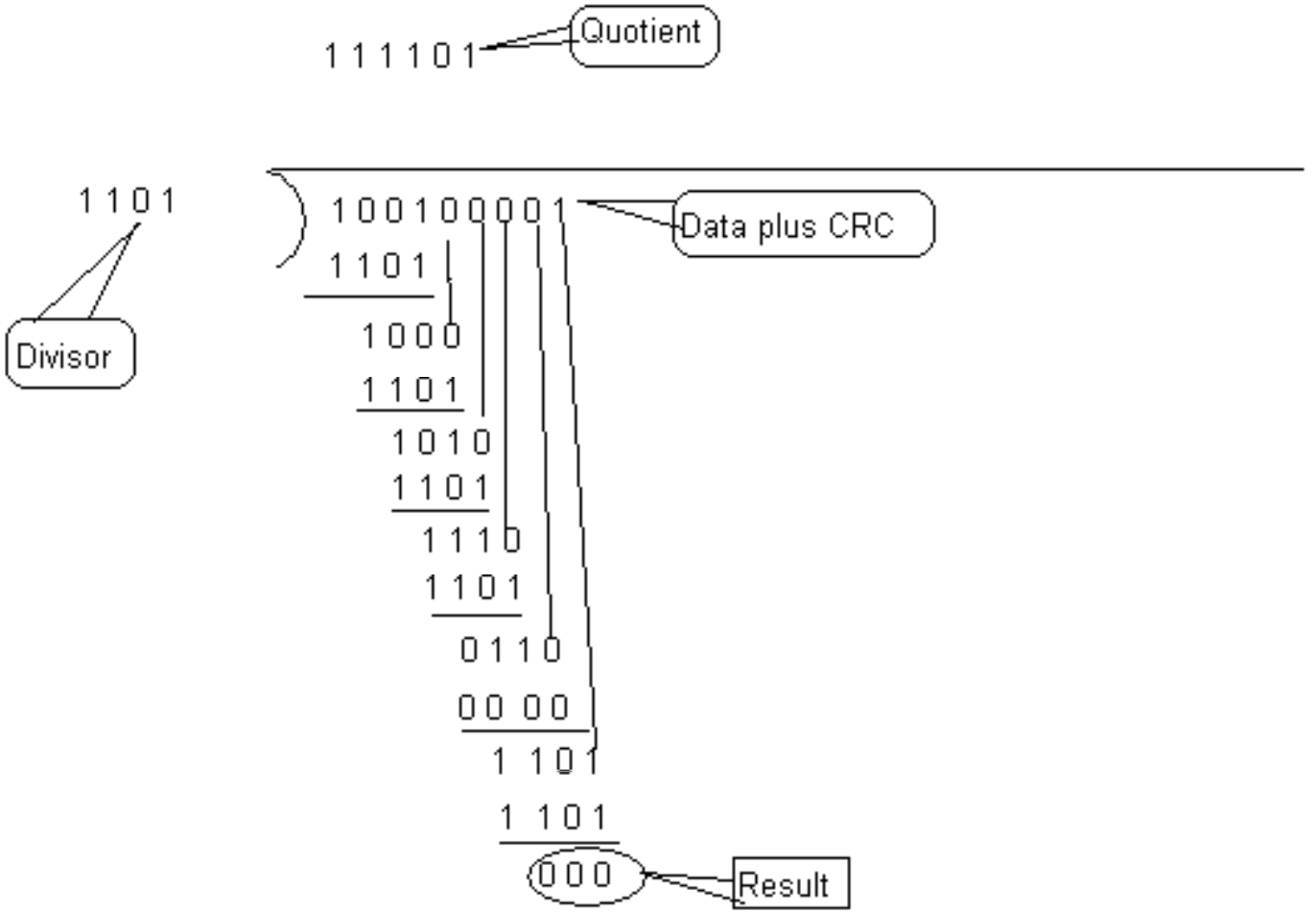
Modulo 2 Arithmetic

$$\begin{array}{r} 1111 \\ +1010 \\ \hline 0101 \end{array} \qquad \begin{array}{r} 1111 \\ -0101 \\ \hline 1010 \end{array} \qquad \begin{array}{r} 11001 \\ \times\ 11 \\ \hline 11001 \\ 11001 \\ \hline 101011 \end{array}$$

Binary Division In CRC generator



Binary Division in CRC Checker

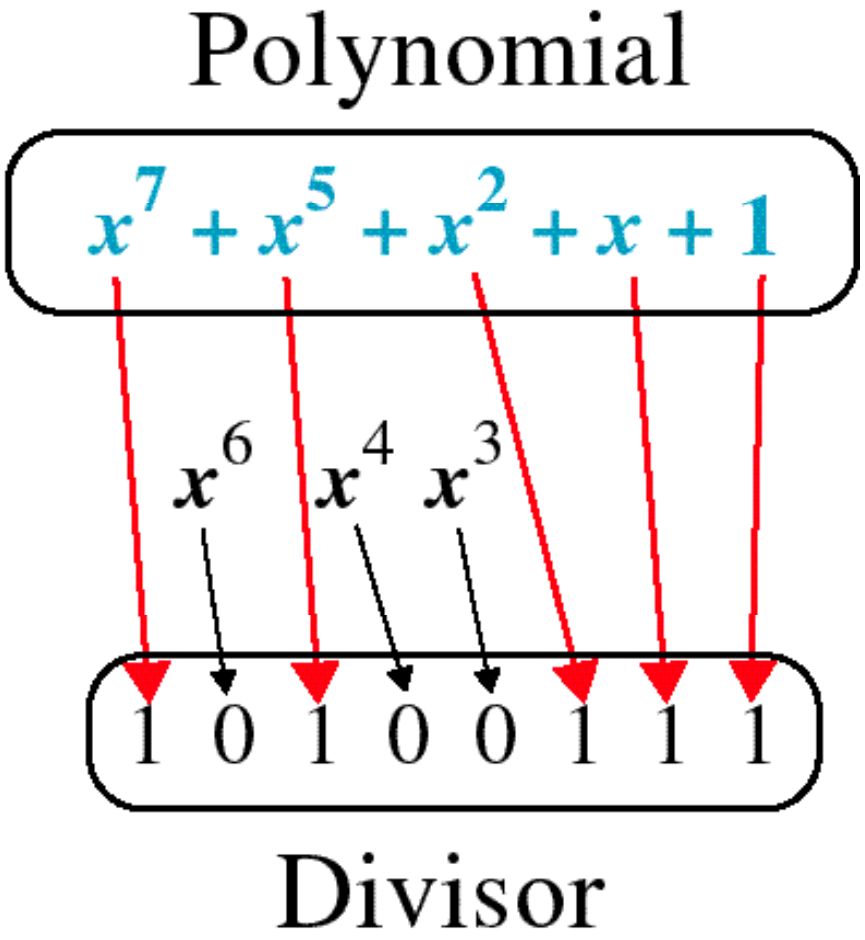


Polynomials

- To express all values as polynomials in a dummy variable x, with binary coefficients
- The coefficients correspond to the bits in the binary number.

$$x^7 + x^5 + x^2 + x + 1$$

Polynomial and Divisor



Polynomial Division

110101

1010001101

6.3

$$Q(X) = X^9 + X^8 + X^6 + X^4 + X^2 + X$$

$$P(X) \rightarrow X^5 + X^4 + X^2 + 1 \quad \big) \quad X^{14} + X^{12} + X^8 + X^7 + X^5 \leftarrow X^5 D(X)$$

$$X^{14} + X^{13} + X^{11} + X^9$$

$$X^{13} + X^{12} + X^{11} + X^9 + X^8$$

$$X^{13} + X^{12} + X^{10} + X^8$$

$$X^{11} + X^{10} + X^9 + X^7$$

$$X^{11} + X^{10} + X^8 + X^6$$

$$X^9 + X^8 + X^7 + X^6 + X^5$$

$$X^9 + X^8 + X^6 + X^4$$

$$X^7 + X^5 + X^4$$

$$X^7 + X^6 + X^4 + X^2$$

$$X^6 + X^5 + X^2$$

$$X^6 + X^5 + X^3 + X$$

01110

$$X^3 + X^2 + X \leftarrow R(X)$$

CRC Standards—P(X)

- CRC-12
  - Used when character length is 6 bits
- CRC-16 and CRC-CCITT
  - Used when character length is 8 bits
  - Used in WANs
- CRC-32
  - Used in LANs

Standard Polynomials

CRC-12

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

CRC-16

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

CRC-ITU

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

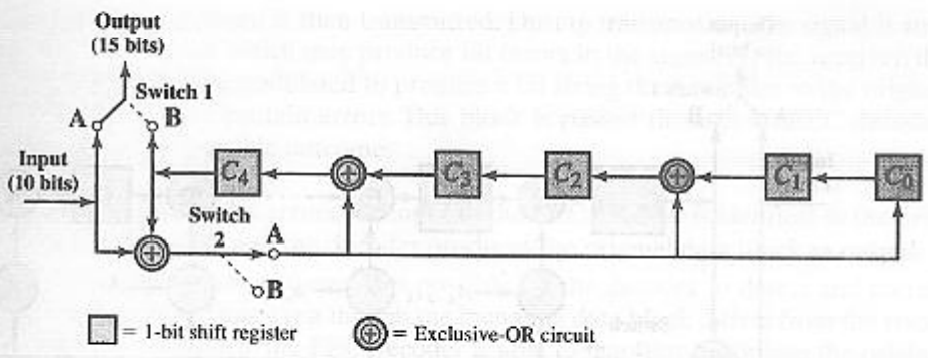
CRC-32

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

Digital Logic

- CRC can be implemented using XOR gates and a shift register
- Shift register
  - Is a string of 1 bit storage devices
  - Each device has an output line, indicates the value currently stored, and an input line
  - At discrete time instants (clock times) the value in the storage device is replaced by the value indicated by its input line

Example with P = 110101



(a) Shift-register implementation

	C_4	C_3	C_2	C_1	C_0	$C_4 \oplus C_3 \oplus I$	$C_4 \oplus C_1 \oplus I$	$C_4 \oplus I$	I = input	
Initial	0	0	0	0	0	1	1	1	1	
Step 1	1	0	1	0	1	1	1	1	0	
Step 2	1	1	1	1	1	1	1	0	1	
Step 3	1	1	1	1	0	0	0	1	0	
Step 4	0	1	0	0	1	1	0	0	0	Message to be sent
Step 5	1	0	0	1	0	1	0	1	0	
Step 6	1	0	0	0	1	0	0	0	1	
Step 7	0	0	0	1	0	1	0	1	1	
Step 8	1	0	0	0	1	1	1	1	0	
Step 9	1	0	1	1	1	0	1	0	1	
Step 10	0	1	1	1	0					

(b) Example with input of 1010001101

Figure 6.5 Circuit with Shift Registers for Dividing by the Polynomial $X^5 + X^4 + X^2 + 1$