CNL
Communications & Networking Laboratory

ECE 462 – Data and Computer Communications

Lecture 9/10: Digital Data Communications Techniques

Bijan Jabbari, PhD
Dept. of Electrical and Computer Eng.
George Mason University
bjabbari@gmu.edu

October 3 and 9, 2007

Outline

- Asynchronous and Synchronous Communications
- Error Detection
  - Parity check
  - CRC
- Error Correction
  - Block codes

Note: Some material adapted from various textbook. In particular, the sequences of slides have been sorted to match closely that of the textbook Data and Computer Communications by W. Stallings, 7th Edition, Prentice Hall, 2007

Asynchronous and Synchronous Transmission

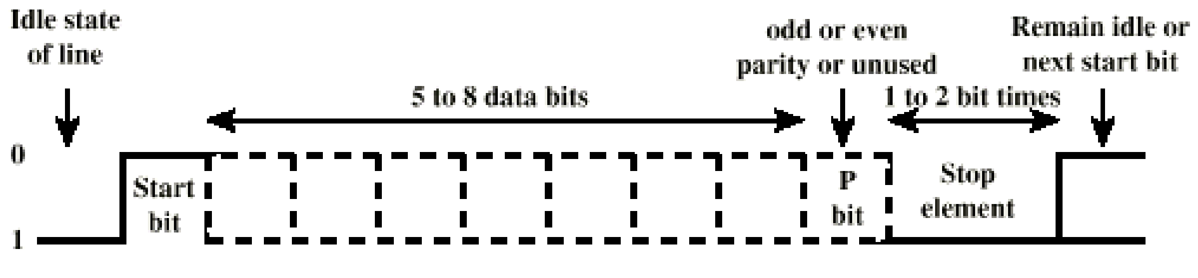
- Timing problems require a mechanism to synchronize the transmitter and receiver
- Two solutions
  - Asynchronous
  - Synchronous

CNL
Communications & Networking Laboratory

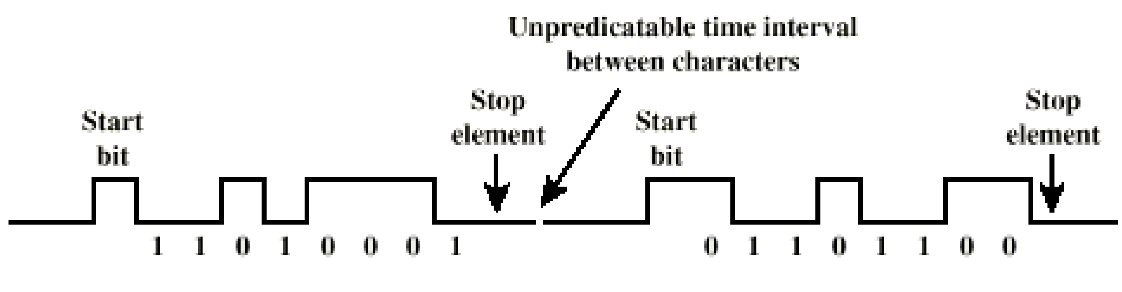
Asynchronous

- Data transmitted on character at a time
  - 5 to 8 bits
- Timing to be maintained within each character
- Resynchronize with each character

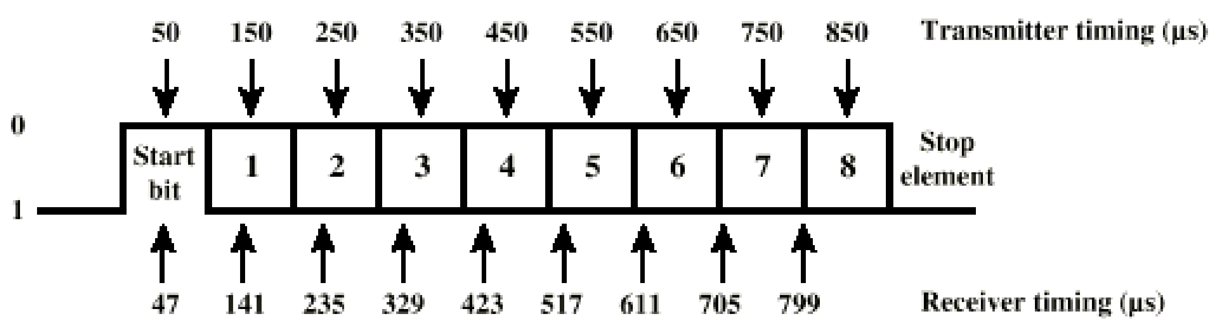
Asynchronous (diagram)



(a) Character format



(b) 8-bit asynchronous character stream



(c) Effect of timing error

Asynchronous - Behavior

- In a steady stream, interval between characters is uniform (length of stop element)
- In idle state, receiver looks for transition 1 to 0
- Then samples next seven intervals (char length)
- Then looks for next 1 to 0 for next char

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

Synchronous - Bit Level

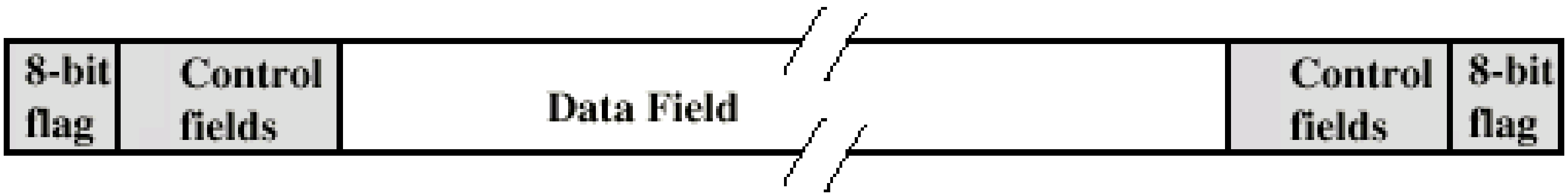
- Block of data transmitted without start or stop bits
- Clocks must be synchronized
- Can use separate clock line
  - Good over short distances
  - Subject to impairments
- Embed clock signal in data
  - Manchester encoding
  - Carrier frequency (analog)

Synchronous - Block Level

- Need to indicate start and end of block
- Use preamble and postamble
  - e.g. series of SYN characters
  - e.g. block of 11111111 patterns ending in 11111110

- More efficient (lower overhead) than async

Synchronous (diagram)



Flag (F)

- The bit pattern 01111110
- The opening flag indicates the start of a frame
- The opening flag of one frame is normally the closing flag of the preceding signal unit
- The closing flag indicates the end of a fame
- ***Zero Insertion*** to prevent flag code imitation (zero insertion and deletion after every five consecutive 1s)

Example 1:

Data bitstream ....011011111111001111100...

Transmitted Bitstream:

$\boxed{01111110}$.........0110111110110011111000....$\boxed{01111110}$

Received bitstream: ....011011111111001111100...

Misalignments

Example 2:

0111111011001100011110001111010111000010011011010 1111110

An error in the message will cause:

011111101100110001111000111**1**10111000010011011010 1111110

F110011000111100F11100001001101101F

Or, two messages:

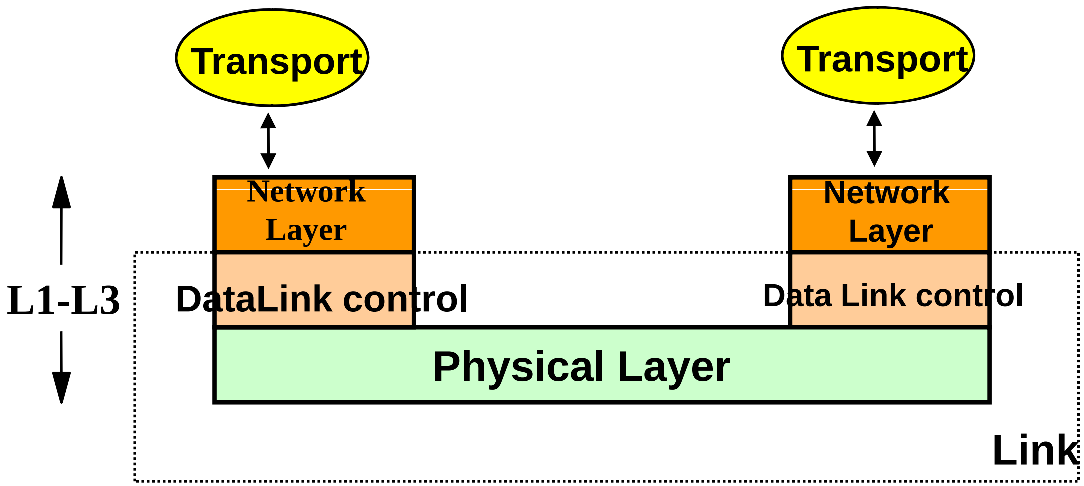
F00110110011100110111**1F**00111101001101011 01F

will become one message:

F0011011001110011011111**01101110**00111101001101011 01F

Layer 2

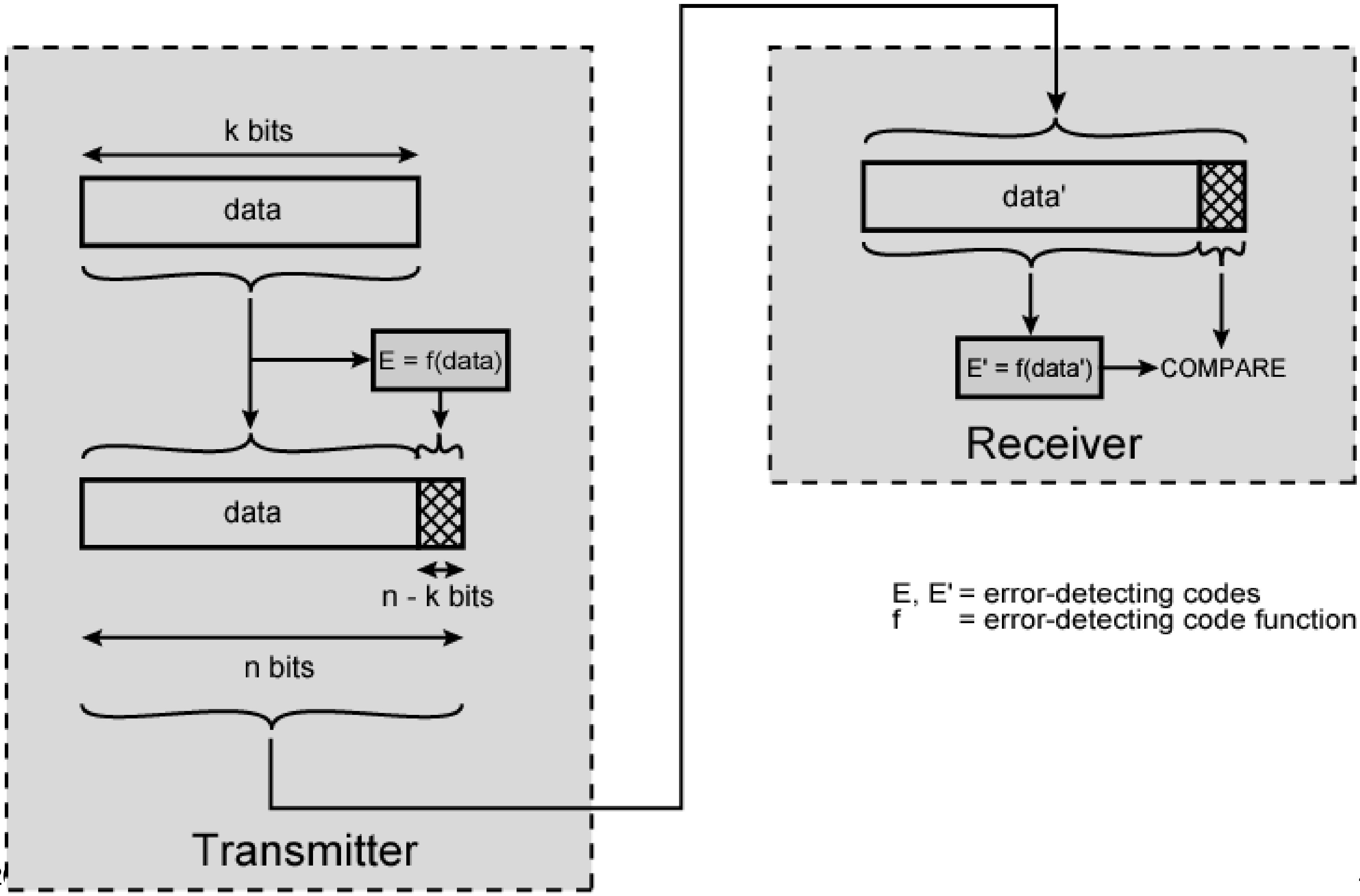
- Layer two functions include Error Detection



Types of Error

- An error occurs when a bit is altered between transmission and reception
- Single bit errors
  - One bit altered
  - Adjacent bits not affected
  - White noise
- Burst errors
  - Length *B*
  - Contiguous sequence of *B* bits in which first, last, and any number of intermediate bits in error
  - Impulse noise
  - Fading in wireless
  - Effect greater at higher data rates

Error Detection Process



E, E' = error-detecting codes
f = error-detecting code function

Error Detection

- Additional bits added by transmitter for error detection code
- Parity
  - Value of parity bit is such that character has even (even parity) or odd (odd parity) number of ones
  - Even number of bit errors goes undetected

Cyclic Redundancy Check

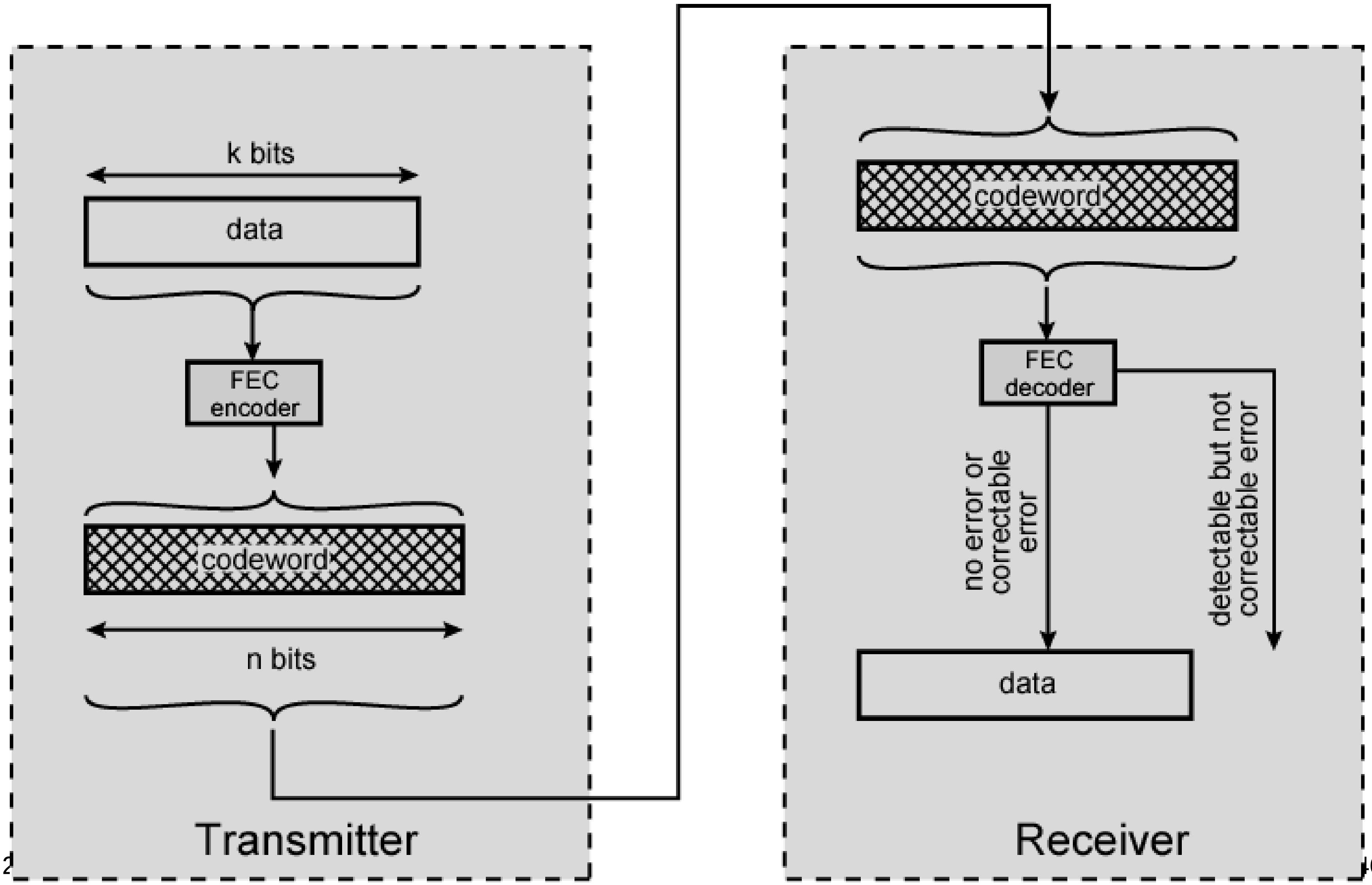
- For a block of *k* bits transmitter generates *n* bit sequence
- Transmit *k+n* bits which is exactly divisible by some number
- Receive divides frame by that number
  - If no remainder, assume no error

CNL Communications & Networking Laboratory

Error Correction

- Correction of detected errors usually requires data block to be retransmitted (see chapter 7)
- Not appropriate for wireless applications
  - Bit error rate is high
    - Lots of retransmissions
  - Propagation delay can be long (satellite) compared with frame transmission time
    - Would result in retransmission of frame in error plus many subsequent frames
- Need to correct errors on basis of bits received

Error Correction Process Diagram



Error Correction Process

- Each k bit block mapped to an n bit block ($n > k$)
  - Codeword
  - Forward error correction (FEC) encoder
- Codeword sent
- Received bit string similar to transmitted but may contain errors
- Received code word passed to FEC decoder
  - If no errors, original data block output
  - Some error patterns can be detected and corrected
  - Some error patterns can be detected but not corrected
  - Some (rare) error patterns are not detected
    - Results in incorrect data output from FEC

Working of Error Correction

- Add redundancy to transmitted message
- Can deduce original in face of certain level of error rate
- E.g. block error correction code
  - In general, add $(n - k)$ bits to end of block
    - Gives n bit block (codeword)
    - All of original k bits included in codeword
  - Some FEC map k bit input onto n bit codeword such that original k bits do not appear

Error Detection

- 16 check bits at the end of each frame perform the error detection function
- The check bits are generated by the transmitting link terminal by operating on the preceding bits of the frame, using a specified algorithm
- At the receiving link terminal, the received check bits are used to examine the preceding bits of the frame
- If complete correspondence is not found, the frame is discarded

Generating Check Bits

- The transmitting signalling link terminal generates the check bits by taking the ones complement of the sum (modulo 2) of *a* and *b* when
- *a* is the remainder of $x^k (x^{15} + x^{14} + x^{13} + ... + x^2 + x + 1)$ divided (modulo 2) by the generator polynomial $x^{16} + x^{12} \; x^5 + 1$
- *b* is the remainder after multiplication by x^{16} and then division (modulo 2) by the generator polynomial $x^{16} + x^{12} + x^5 + 1$ of the contents of the *k* bits of the frame
- The *k* bits of the frame start after the final bit of the opening flag up to the beginning of the check bits, excluding bits inserted for transparency

Generators

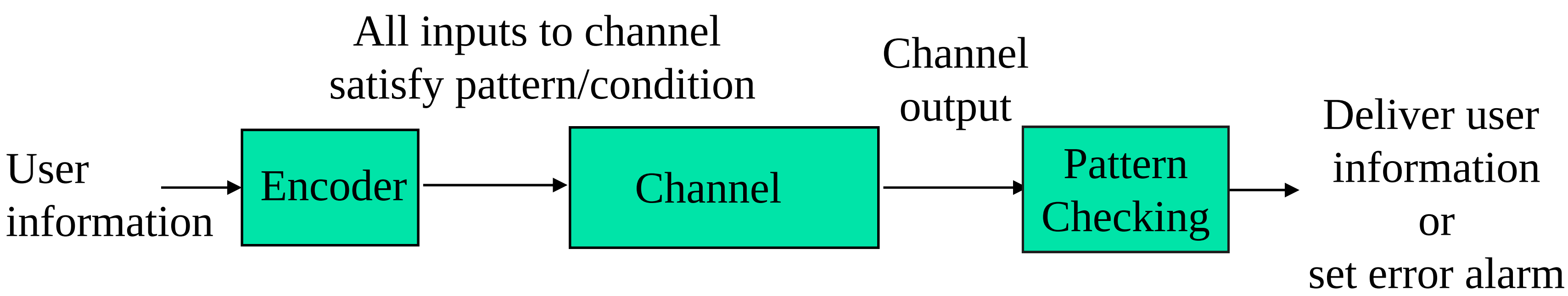
CRC-12 $x^{12} + x^{11} + x^3 + x^2 + x + 1$

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

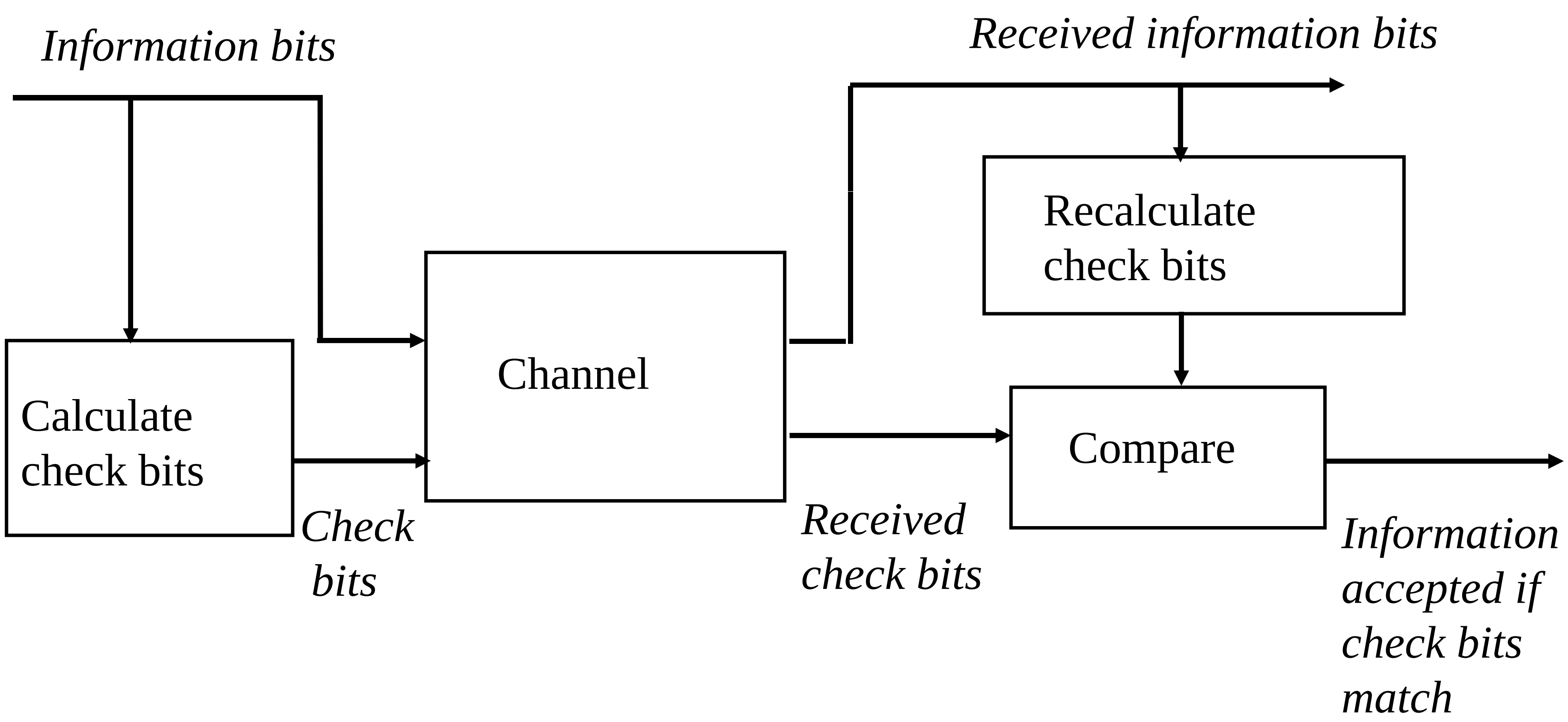
CRC-CCITT $x^{16} + x^{12} + x^5 + 1$

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

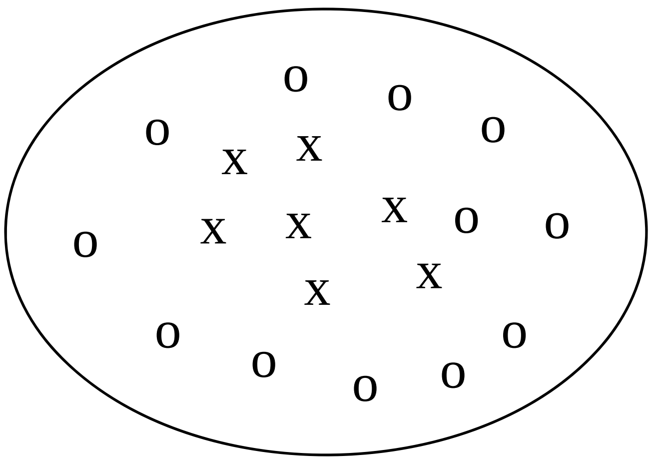
CRC-ATM $X^8 + X^2 + X + 1$ (ATM Header Error Control -HEC)



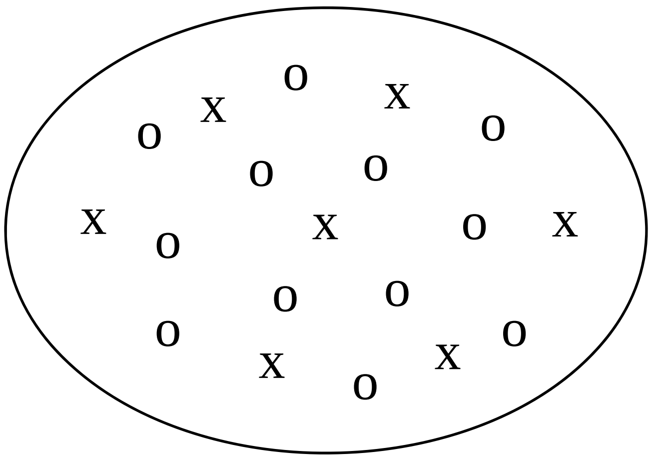
All inputs to channel
satisfy pattern/condition
Channel
output
User
information
Encoder
Channel
Pattern
Checking
Deliver user
information
or
set error alarm


Information bits
Received information bits
Recalculate check bits
Channel
Calculate check bits
Compare
Check bits
Received check bits
Information accepted if check bits match

(a) A code with poor distance properties



(b) A code with good distance properties



x = codewords o = **non-codewords**

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

Last column consists of check bits for each row

Bottom row consists of check bit for each column

One error

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

Two errors

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

Three errors

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

Four errors

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

Arrows indicate failed checks bits

Addition: $(x^7 + x^6 + 1) + (x^6 + x^5) = x^7 + (1+1)x^6 + x^5 + 1$

$$= x^7 + x^5 + 1$$

Multiplication: $(x+1)(x^2 + x + 1) = x^3 + x^2 + x + x^2 + x + 1 = x^3 + 1$

Division:

$$
\begin{array}{r l}
 & x^3 + x^2 + x \;=\; q(x) \text{ quotient} \\
\cline{2-2}
x^3 + x + 1 \,\big) & x^6 + x^5 \\
 & x^6 + \quad x^4 + x^3 \\
\cline{2-2}
 & \quad x^5 + x^4 + x^3 \\
 & \quad x^5 + \quad x^3 + x^2 \\
\cline{2-2}
 & \qquad x^4 + \quad x^2 \\
 & \qquad x^4 + \quad x^2 + x \\
\cline{2-2}
 & \qquad\qquad x \;=\; r(x) \text{ remainder}
\end{array}
$$

divisor

dividend

$$
\begin{array}{r l}
 & 3 \\
\cline{2-2}
35\,\big) & 122 \\
 & \underline{105} \\
 & \;17
\end{array}
$$

Steps:

1) Multiply $i(x)$ by x^{n-k} (puts zeros in $(n-k)$ low order positions)

2) Divide $x^{n-k}\ i(x)$ by $g(x)$

$$x^{n-k}i(x) = \underbrace{g(x)\ q(x)}_{\text{quotient}} + \underbrace{r(x)}_{\text{remainder}}$$

3) Add remainder $r(x)$ to $x^{n-k}\ i(x)$
(puts check bits in the $n-k$ low order positions):

$$b(x) = x^{n-k}i(x) + r(x) \longleftarrow \text{transmitted codeword}$$

Operation

Generator polynomial: $g(x) = x^3 + x + 1$

Information: (1,1,0,0) $\Longrightarrow$ $i(x) = x^3 + x^2$

Encoding: $x^3 i(x) = x^6 + x^5$

$$
\begin{array}{r l}
 & x^3 + x^2 + x \\
\cline{2-2}
x^3 + x + 1 \,\big) & x^6 + x^5 \\
 & x^6 + x^4 + x^3 \\
\cline{2-2}
 & x^5 + x^4 + x^3 \\
 & x^5 + x^3 + x^2 \\
\cline{2-2}
 & x^4 + x^2 \\
 & x^4 + x^2 + x \\
\cline{2-2}
 & x
\end{array}
$$

$$
\begin{array}{r l}
 & 1110 \\
\cline{2-2}
1011 \,\big) & 1100000 \\
 & 1011 \\
\cline{2-2}
 & 1110 \\
 & 1011 \\
\cline{2-2}
 & 1010 \\
 & 1011 \\
\cline{2-2}
 & 010
\end{array}
$$

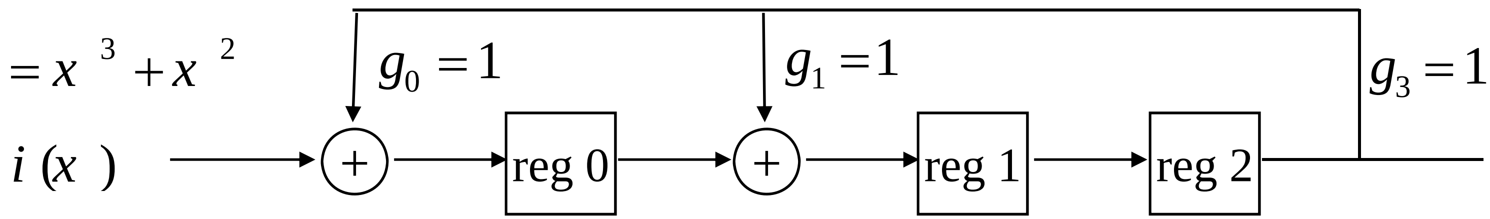
Transmitted codeword:

$b(x) = x^6 + x^5 + x$

$\Longrightarrow$ $\underline{b} = (1,1,0,0,0,1,0)$

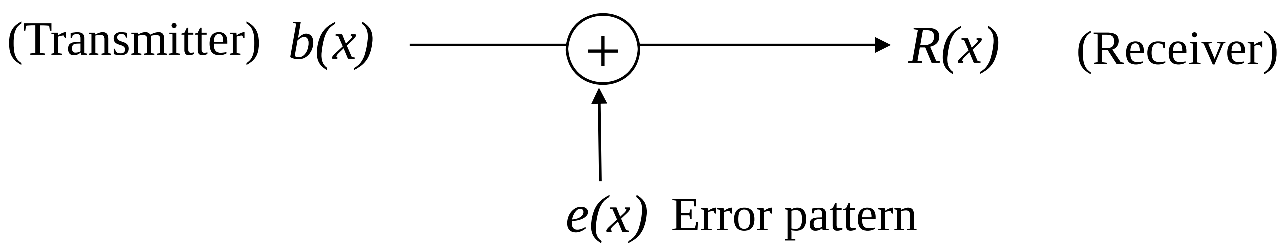
Encoder for $g(x) = x^3 + x + 1$

$i(x) = x^3 + x^2$



clock	input	reg 0	reg 1	reg 2
0	-	0	0	0
1	$1=i_3$	1	0	0
2	$1=i_2$	1	1	0
3	$0=i_1$	0	1	1
4	$0=i_0$	1	1	1
5	0	1	0	1
6	0	1	0	0
7	0	**0**	**1**	**0**
	check bits:	$r_0 = 0$	$r_1 = 1$	$r_2 = 0$

$\Longrightarrow$ $r(x) = x$


(Transmitter) b(x)
+
R(x)
(Receiver)
e(x) Error pattern

1. **Single errors:** $e(x) = x^i$ $\quad 0 \le i \le n$-1

   If $g(x)$ has more than one term, it cannot divide $e(x)$

2. **Double errors:** $e(x) = x^i + x^j \quad 0 \le i < j \le n-1$

   $= x^i (1 + x^{j-i})$

   If $g(x)$ is primitive, it will not divide $(1 + x^{j-i})$ for $j-i \le 2^{n-k}-1$

3. **Odd number of errors:** $e(1) = 1$ If number of errors is odd

   If $g(x)$ has $(x+1)$ as a factor, then $g(1) = 0$ and all codewords have an even number of 1s.

4. Error bursts of length b:

0001101100 • • • 0

*i*th position

L

0000**110** • • • •

error pattern *d(x)*

$e(x) = x^i\ d(x)$ where deg$(d(x)) = L\text{-}1$

$g(x)$ has degree $n\text{-}k$;

$g(x)$ cannot divide $d(x)$ if deg$(g(x))$> deg$(d(x))$

- ***L = (n-k)*** or less: all will be detected
- ***L = (n-k+1)***: deg$(d(x))$ = deg$(g(x))$

  i.e. d(x) = g(x) is the only undetectable error pattern,

  fraction of bursts which are undetectable = $1/2^{L-2}$
- ***L > (n-k+1)***: fraction of bursts which are undetectable = $1/2^{n-k}$