Chapter 1
Coding for Reliable Digital Transmission and Storage
Introduction

- A major concern of designing digital data transmission and storage systems is the control of errors so that reliable reproduction of data can be obtained.

- In 1948, Shannon demonstrated that, by proper encoding of the information, errors induced by a noisy channel or storage medium can be reduced to any desired level without sacrificing the rate of information transmission or storage, as long as the information rate is less than the capacity of the channel.

- A great deal of effort has been expended on the problem of devising efficient encoding and decoding methods for error control in a noisy environment.
Typical Digital Communications Systems

- Block diagram of a typical data transmission or storage system
Channel Code: Error-Correcting Codes

- **Message**
- **Transmitted Codeword**
  - (Encoding)
  - (Redundancy Bits)
- **Received Codeword**
  - Bad Design
  - Good Design
Example: A (3,1) Repetition Code

\[(n,k)\] error correction code \(\Rightarrow k\) bits message, \(n\) bit codeword
Parity check bit \(\Rightarrow n-k\) bits
1.2 Types of Major Codes

- There are four types of codes in common use today:
  - Block codes
  - Convolutional codes
  - Turbo codes
  - Low-Density Parity-Check (LDPC) Codes
Block Codes

- **Block codes (cont.)**
  - The encoder for a block code divides the information sequence into message blocks of \( k \) information bits each.
  - A message block is represented by the binary \( k \)-tuple \( \mathbf{u} = (u_1, u_2, \cdots, u_k) \) called a *message*.
  - There are a total of \( 2^k \) different possible messages. The encoder transforms each message \( \mathbf{u} \) into an \( n \)-tuple \( \mathbf{v} = (v_1, v_2, \cdots, v_n) \) of discrete symbols called a *codeword*.
  - This set of \( 2^k \) code words of length \( n \) is called an \((n,k)\) *block code*.
  - The ratio \( R = k/n \) is called the *code rate*.
  - \( n-k \) *redundant* bits can be added to each message.
  - Since the \( n \)-symbol output codeword depends only on the corresponding \( k \)-bit input message, the encoder is *memoryless*, and can be implemented with a combinational logic circuit.
Finite Field (Galois Field)

- Much of the theory of linear block code is highly mathematical in nature, and requires an extensive background in modern algebra.

- Finite field was invented by the early 19th century mathematician, Evariste Galois.

- Galois was a young French math whiz who developed a theory of finite fields, now know as Galois fields, before being killed in a duel at the age of 21.

- For well over 100 years, mathematicians looked upon Galois fields as elegant mathematics but of no practical value.
Convolutional Codes

- Convolutional code
  - The encoder for a convolutional code also accepts $k$-bit blocks of the information sequence $u$ and produces an encoded sequence (code word) $v$ of $n$-symbol blocks.
  
  - Each encoded block depends not only on the corresponding $k$-bit message block at the same time unit, but also on $m$ previous message blocks. Hence, the encoder has a memory order of $m$.
  
  - The set of encoded sequences produced by a $k$-input, $n$-output encoder of memory order $m$ is called an $(n, k, m)$ convolutional code.
  
  - The ratio $R = k/n$ is called the code rate.
  
  - Since the encoder contains memory, it must be implemented with a sequential logic circuit.
Convolutional Codes

- Binary convolutional encoder with $k=1$, $n=2$, and $m=2$
Turbo Codes Basic Concepts

- Turbo coding uses **parallel** or **serial** concatenation of two recursive systematic convolutional codes joined through an interleaver.
- Information bits are encoded **block by block**.
- Turbo codes uses iterative decoding techniques.
- Soft-output decoder is necessary for iterative decoding.
- Turbo codes can approach to Shannon limit.
Turbo Codes Encoder – An Example

When the switch is placed on the low position, the tail bits are feedback and the trellis will be terminated.
1.5 Types of Errors

- On memoryless channels, the noise affects each transmitted symbol independently.
- Memoryless channels are called random-error channels.

![Transition probability diagrams for binary symmetric channel (BSC).](image)
1.5 Types of Errors

- On channels with memory, the noise is not independent from transmission to transmission.
- Channel with memory are called *burst-error channels*.

Simplified model of a channel with memory.
1.6 Error Control Strategies

- Error control for a one-way system must be accomplished using forward error correction (FEC), that is, by employing error-correcting codes that automatically correct errors detected at the receiver.

- Error control for a two-way system can be accomplished using error detection and retransmission, called automatic repeat request (ARQ). This is also known as the backward error correction (BEC).
  
  - In an ARQ system, when errors are detected at the receiver, a request is sent for the transmitter to repeat the message, and this continues until the message is received correctly.

- The major advantage of ARQ over FEC is that error detection requires much simpler decoding equipment than does error correction.
1.6 Error Control Strategies

- ARQ is adaptive in the sense that information is retransmitted only when errors occur.

- When the channel error rate is high, retransmissions must be sent too frequently, and the system throughput, the rate at which newly generated messages are correctly received, is lowered by ARQ.

- In general, wire-line communications (more reliable) adopts BEC scheme, while wireless communications (relatively unreliable) adopts FEC scheme.
Error Detecting Codes

- Cyclic Redundancy Code (CRC Code) – also know as the polynomial code.

- Polynomial codes are based upon treating bit strings as representations of polynomials with coefficients of 0 and 1 only.

- For example, 110001 represents a six-term polynomial: $x^5+x^4+x^0$

- When the polynomial code method is employed, the sender and receiver must agree upon a generator polynomial, $G(x)$, in advance.

- To compute the checksum for some frame with $m$ bits, corresponding to the polynomial $M(x)$, the frame must be longer than the generator polynomial.
Error Detecting Codes

- The idea is to append a checksum to the end of the frame in such a way that the polynomial represented by the checksummed frame is divisible by $G(x)$.
- When the receiver gets the checksummed frame, it tries dividing it by $G(x)$. If there is a remainder, there has been a transmission error.
- The algorithm for computing the checksum is as follows:

1. Let $r$ be the degree of $G(x)$. Append $r$ zero bits to the low-order end of the frame, so it now contains $m + r$ bits and corresponds to the polynomial $x^rM(x)$.
2. Divide the bit string corresponding to $G(x)$ into the bit string corresponding to $x^rM(x)$ using modulo 2 division.
3. Subtract the remainder (which is always $r$ or fewer bits) from the bit string corresponding to $x^rM(x)$ using modulo 2 subtraction. The result is the checksummed frame to be transmitted. Call its polynomial $T(x)$. 

Calculation of the polynomial code checksum

(a)

Frame contents: 11100110
With appended zeros: 11100110 0000
Generator polynomial: 11001

Transmitted frame: 11100110 0110

\[
\begin{array}{cccc}
1 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 \\
+ & 1 & 1 & 0 \\
0 & 0 & 1 & 1 \\
+ & 0 & 0 & 0 \\
0 & 1 & 0 & 1 \\
+ & 1 & 1 & 0 \\
0 & 1 & 1 & 0 \\
+ & 1 & 1 & 0 \\
0 & 0 & 1 & 0 \\
+ & 0 & 0 & 0 \\
0 & 1 & 0 & 1 \\
+ & 1 & 1 & 0 \\
0 & 1 & 1 & 0 \\
+ & 1 & 1 & 0 \\
0 & 0 & 1 & 1 \\
+ & 0 & 0 & 0 \\
0 & 1 & 1 & 0 \\
\end{array}
\]

= Quotient (ignored)

= Remainder (FCS/CRC)
Calculation of the polynomial code checksum

Remainder = 0: no errors
Remainder ≠ 0: error detected
Cyclic Redundancy Code (CRC)

• Examples of CRCs used in practice:

\[
\begin{align*}
\text{CRC – 16} & = x^{16} + x^{15} + x^{2} + 1 \\
\text{CRC – CCITT} & = x^{16} + x^{12} + x^{5} + 1 \\
\text{CRC – 32} & = x^{32} + x^{26} + x^{23} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} \\
& \quad + x^{5} + x^{4} + x^{22} + x + 1
\end{align*}
\]

• A 16-bit checksum catches all single and double errors, all errors with an odd number of bits, all burst errors of length 16 or less, 99.997% of 17-bit error bursts, and 99.998% of 18-bit and longer bursts.