EEN 1043/EE452 Wireless and Mobile Communication

Channel Coding

Sobia Jangsher

Assistant Professor

School of Electronic Engineering

Channel Coding

- Different properties of the communication channel result in errors made by the receivers
- Channel coding is performed to detect and/or correct those errors
- Most systems use additional bits that are added to the useful information
- Detected errors are dealt with in one of two ways
  - ARQ: automatic request repeat – message is resent
  - FEC: forward error correction -- errors are corrected based on coded information
- Channel coding is an essential technique in a noisy channel

Channel Coding

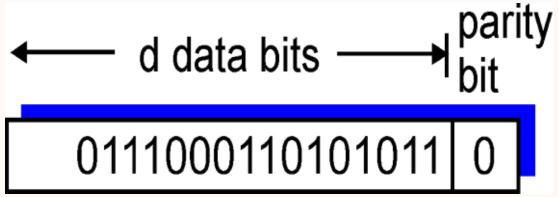
- Error Detection
- Error Correction

- Coding rate: It is the proportion of the data-stream that is useful
  - r=k/n
- Hamming Distance: It is the number of bit positions in which the two bits are different
- 11011001
- 10011101
- d(11011001, 10011101)
- 11011001 XOR 10011101 = 01000100 (count number of 1s)

Channel Coding: Parity checking

Single bit parity:

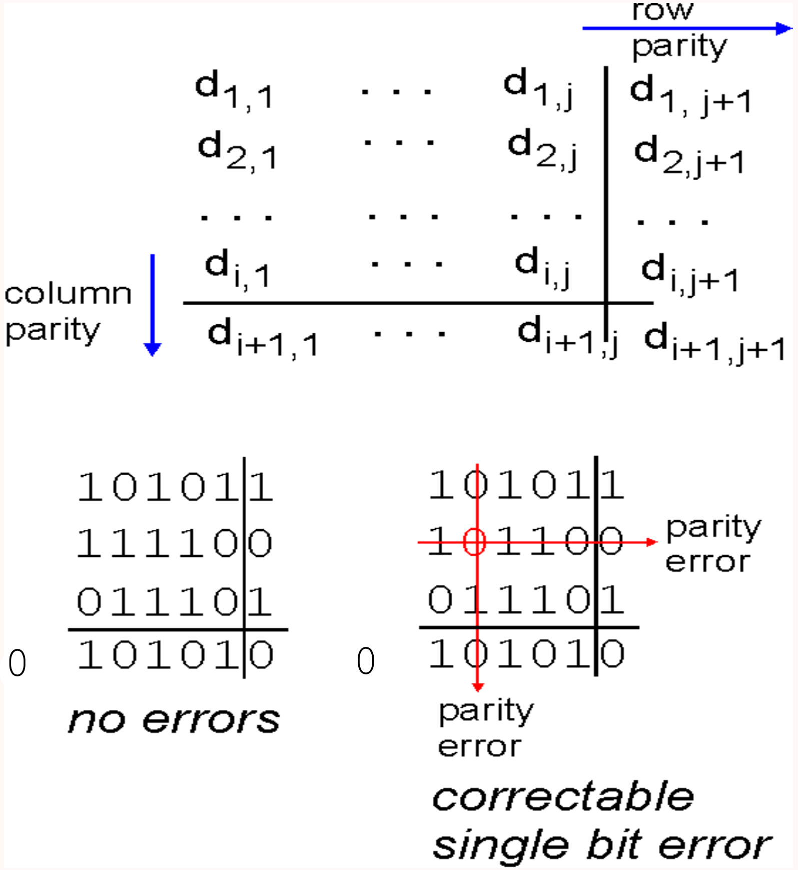
- detect single bit errors



Add an extra bit (parity bit) to the data, such that the total number of 1 bits is even (even parity) or odd (odd parity)
Even parity – Make the number of 1's in a bit string an even number
Odd parity– Make the number of 1's in a bit string an odd number

Two-dimensional bit parity:

- detect and correct single bit errors



Channel Coding: Convolution Coding

- Convolution codes are used in applications that require good performance (forward error correction) with low implementation cost.
- Operate on data streams, not blocks.
- Have memory

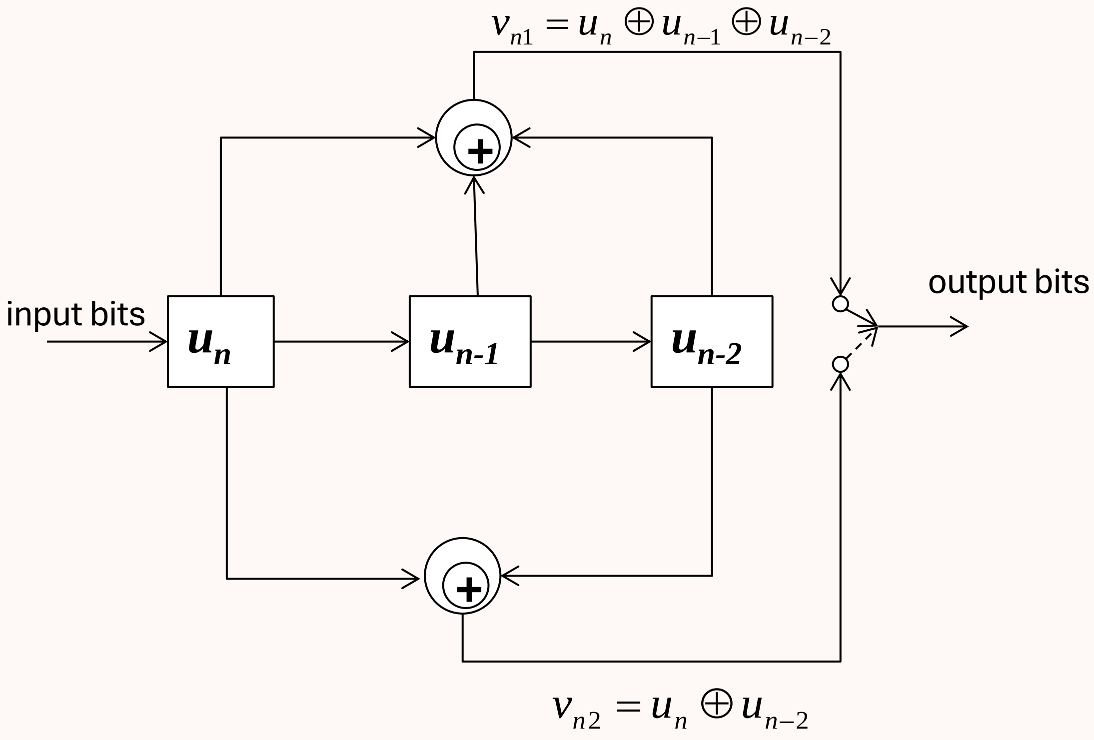
  Coding is done on a small block of k data bits

  - Produces a coded block of n bits
  - Uses K-1 previous blocks of k data bits to produce coded block
    - K referred to as constraint factor or memory depth
  - Typically n, k, and K are small.
    - Code characterised by the numbers (n, k, K).
      - E.g. $(n, k, K) = (2, 1, 3)$
      - Coding rate k/n (e.g. 1/2)

Channel Coding: Convolution Coding

- Producing a good convolution code is complex.
- We will concentrate on how coding and decoding is done.
  - Shift register implementation (memory)
  - Finite state representation
  - Trellis diagram
    - Error correction

Channel Coding: Convolution Coding Encoding example

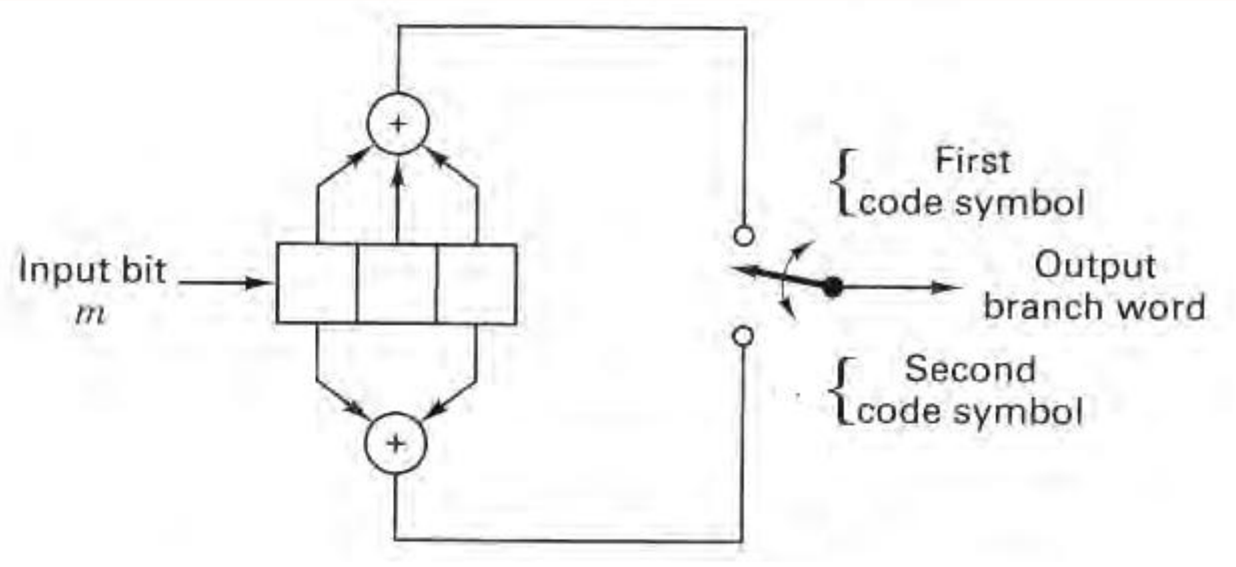


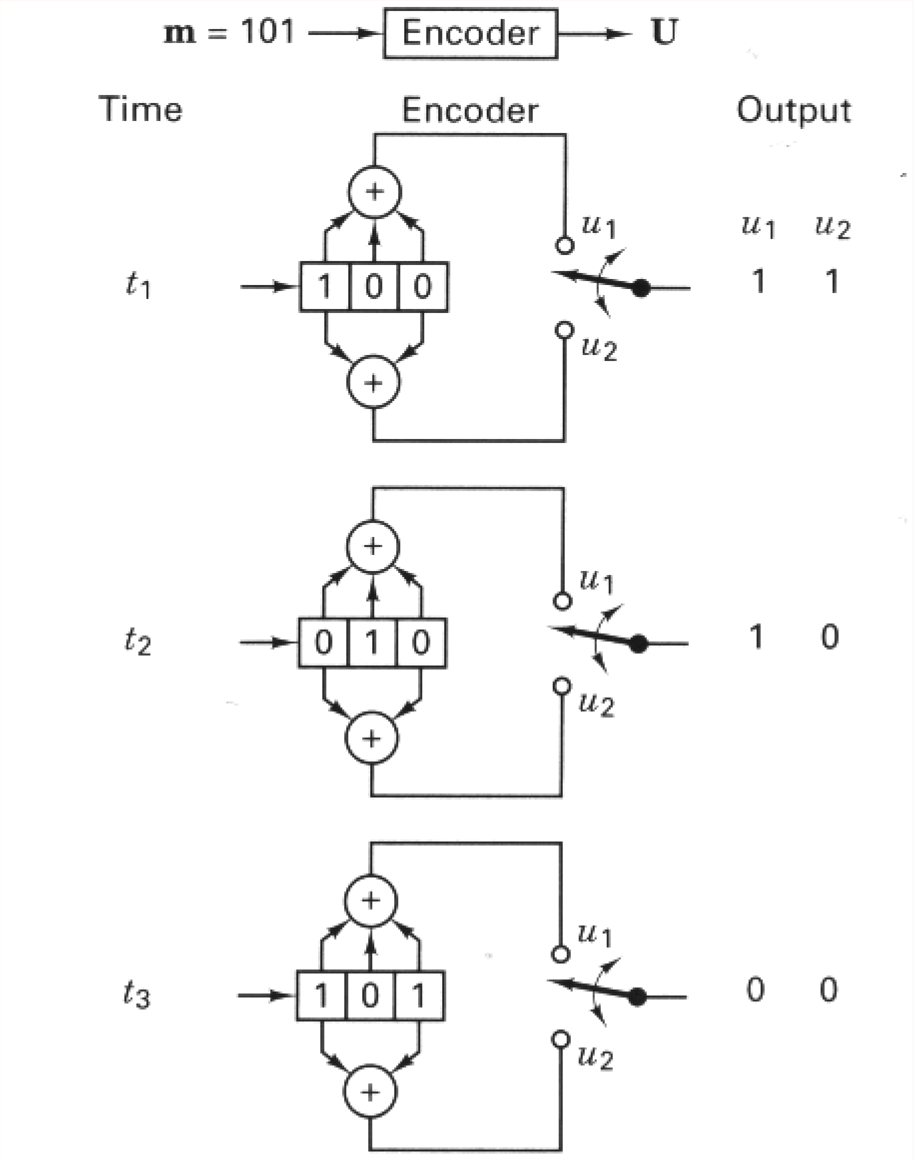
- Example
  - (2,1,3) encoder
  - Input stream *u*
  - Output *v*

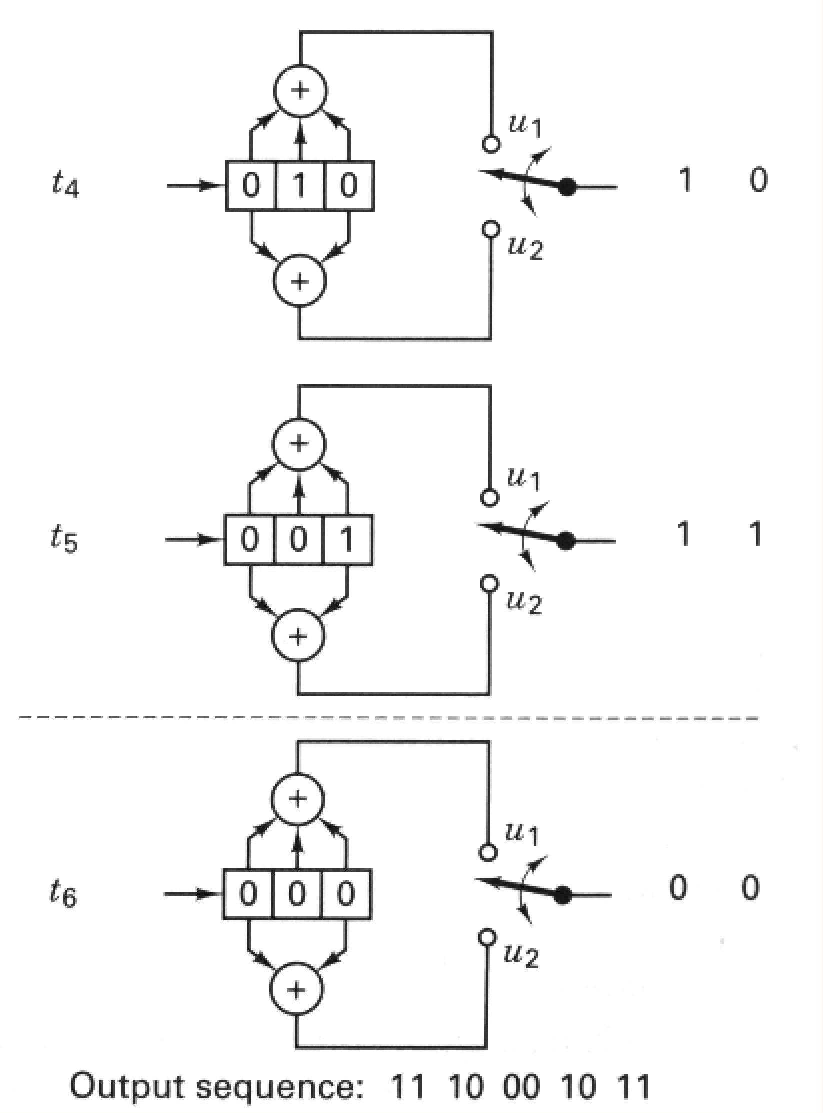
$$v_{n1} = u_n \oplus u_{n-1} \oplus u_{n-2}$$
$$v_{n2} = u_n \oplus u_{n-2}$$

Convolutional Coding

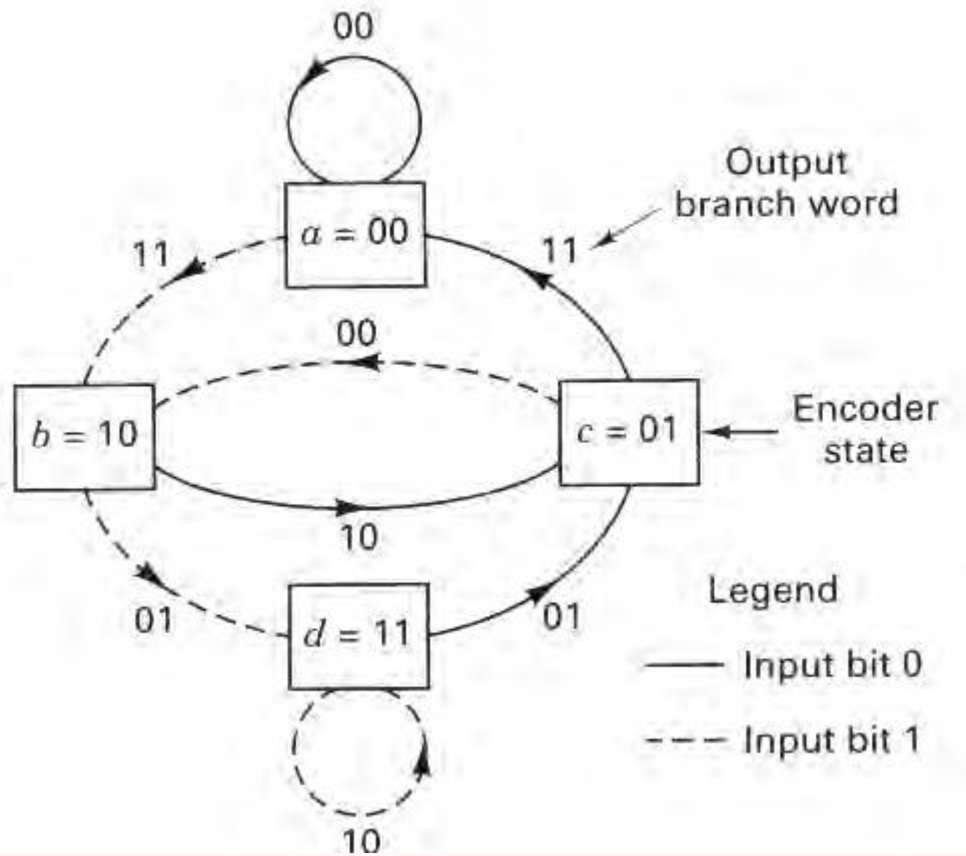
- (n,k,K)=(2,1,3)
- K is the constraint length



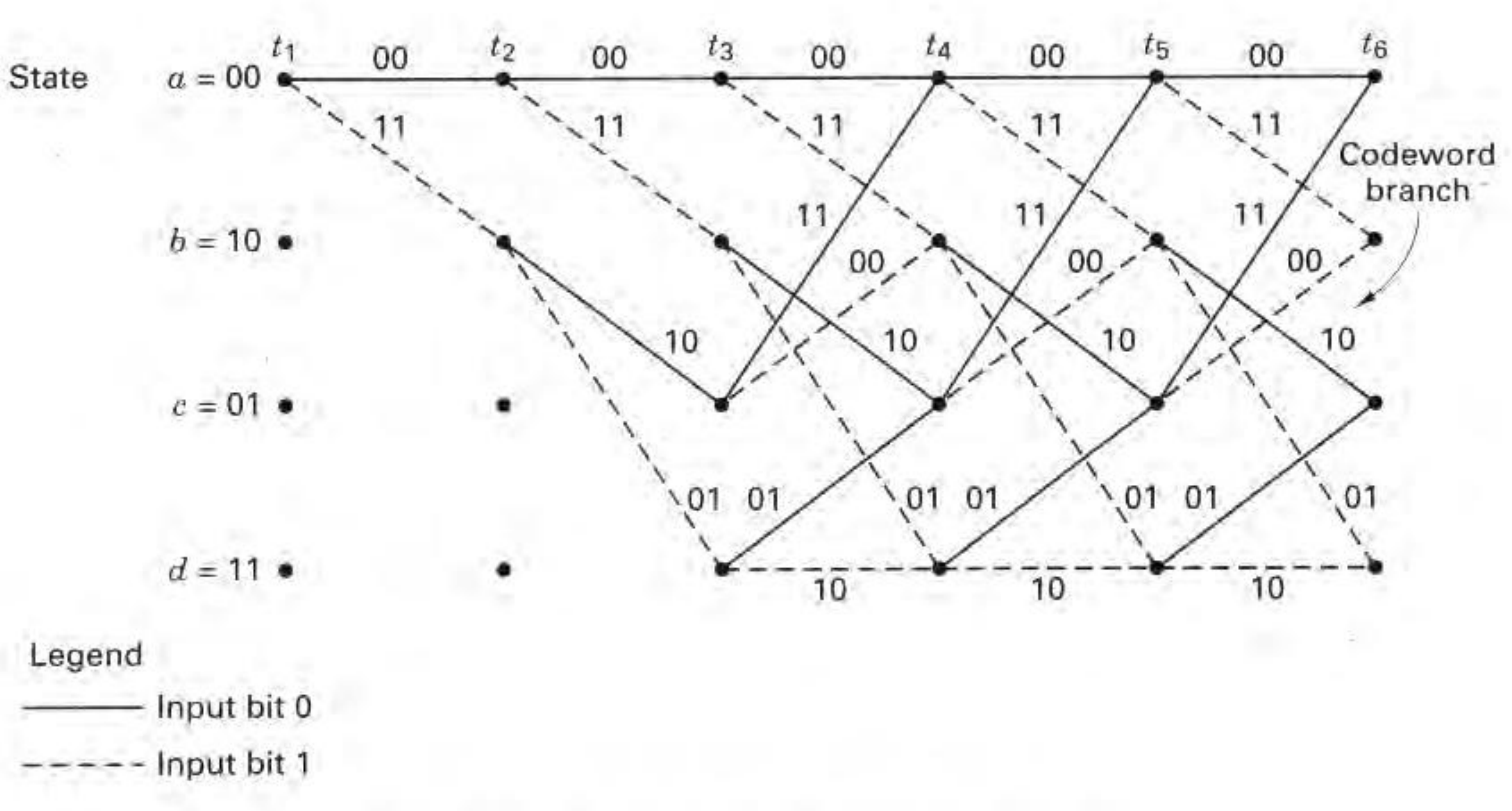

m = 101
Encoder
U
Time
Encoder
Output
u_1 u_2
t_1
1 0 0
1 1
t_2
0 1 0
1 0
t_3
1 0 1
0 0


t_4
0 1 0
1 0
t_5
0 0 1
1 1
t_6
0 0 0
0 0
Output sequence: 11 10 00 10 11

State Diagram



Encoder Trellis



Channel Coding

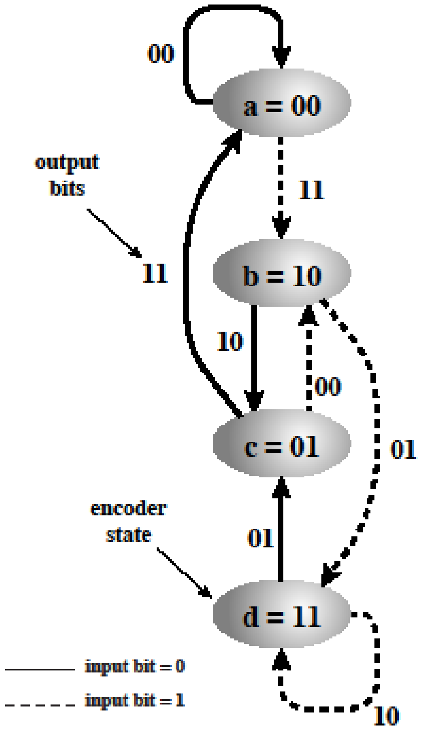
Convolution coding decoding example

- Example
  - (2,1,3) encoder

$$v_{n1} = u_n \oplus u_{n-1} \oplus u_{n-2}$$

$$v_{n2} = u_n \oplus u_{n-2}$$

- So, if there are no error, the input can be decoded from the sequence of states.
  - u = 0110
  - v=00110101
- A trellis diagram is used to represent this visually



(b) Encoder state diagram

Figure 8.9 Convolutional Encoder with $(n, k, K) = (2, 1, 3)$

Channel Coding

Coding gain

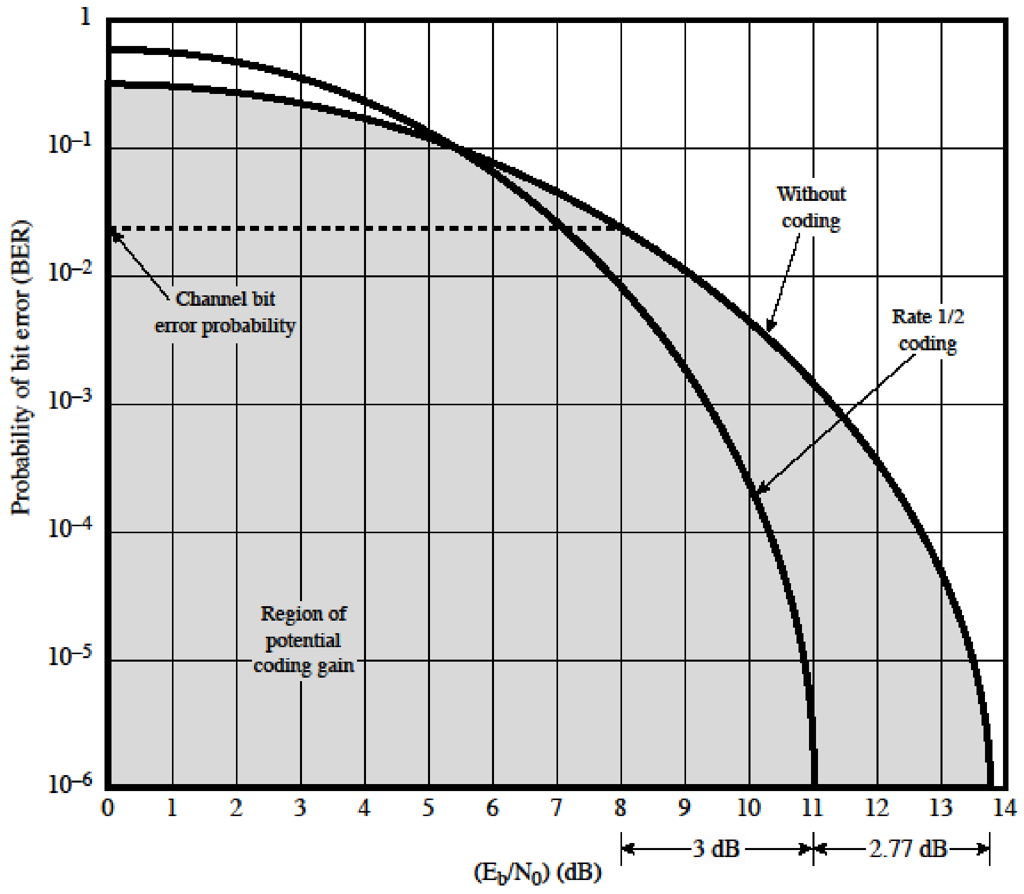


Figure 8.6 How Coding Improves System Performance