

Convolutional Coding

In telecommunication, a **convolutional code** is a type of error-correcting code in which m -bit information symbol to be encoded is transformed into n -bit symbol. Convolutional codes are used extensively in numerous applications in order to achieve reliable data transfer, including digital video, radio, mobile communication, and satellite communication. These codes are often implemented in concatenation with a hard-decision code, particularly Reed Solomon.

Course Name: Error Correcting Codes

Level : UG

Author

Phani Swathi Chitta

Mentor

Prof. Saravanan Vijayakumaran

Learning Objectives

After interacting with this Learning Object, the learner will be able to:

- Explain the convolutional encoding and Viterbi (decoding) Algorithms

Definitions of the components/Keywords:

1

- A convolutional encoder is a finite state machine. An encoder with n registers will have 2^n states.

2

- Convolutional codes have memory that uses previous bits to encode or decode following bits

3

- The most commonly known graphical representation of a code is the trellis representation. A code trellis diagram is simply an edge labeled directed graph in which every path represents a code sequence.

4

- This representation has resulted in a wide range of applications of convolutional codes for error control in digital communications.

5

- Viterbi algorithm is used for decoding a bit stream that has been encoded using forward error correction based on a convolutional code.

- Viterbi decoding compares the hamming distance between the branch code and the received code.

- Path producing larger hamming distance is eliminated.

- In information theory, the **Hamming distance** between two strings of equal length is the number of positions at which the corresponding symbols are different.

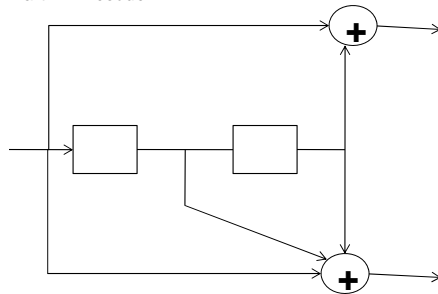
Master Layout 1

1

Part 1 – Encoder
Part 2 – Decoder

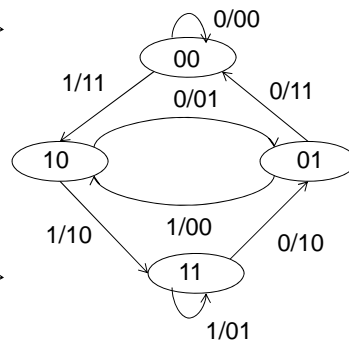
Place a dropdown box for encoder and decoder

2



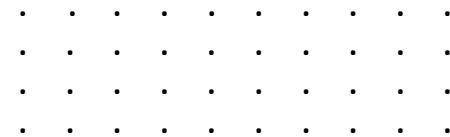
Encoder

3



State diagram

4



Trellis Diagram

5

Step 1:

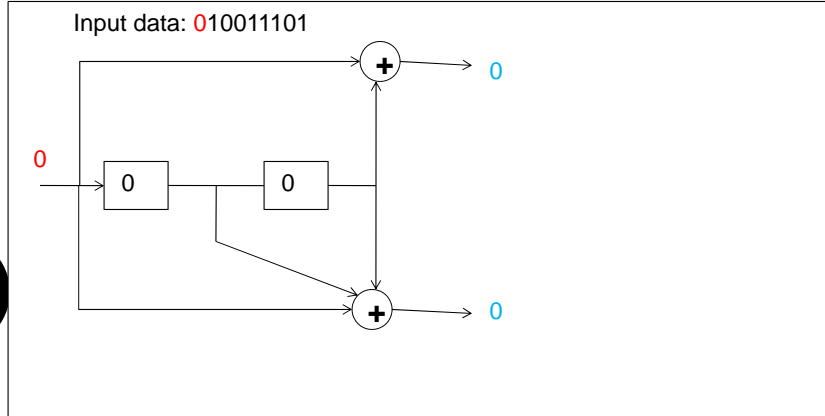
1

2

3

4

5



Instruction for the animator

- The encoder figure in the master layout is shown first.
- Along with that show the state diagram without arrows (show only the ovals)
- After first sentence in DT, red 0 must appear.
- Then the circles should blink and blue 0s must appear.

Text to be displayed in the working area (DT)

- The initial state of encoder is 00 which represents the contents of the shift register in the encoder.
- Input 0s given to the encoder and the corresponding outputs of modulo 2 adders are 00.

Step 2:

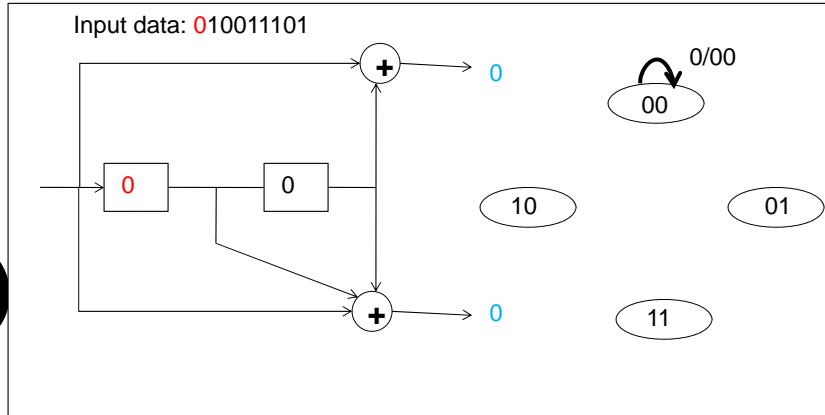
1

2

3

4

5



Instruction for the animator

- Then right shift the bits so that red 0 appears in first box.
- Then the second figure should appear.
- The text in DT should be displayed with the second figure.

Text to be displayed in the working area (DT)

- This is the state diagram.
- 0/00 represents input/output

Step 3:

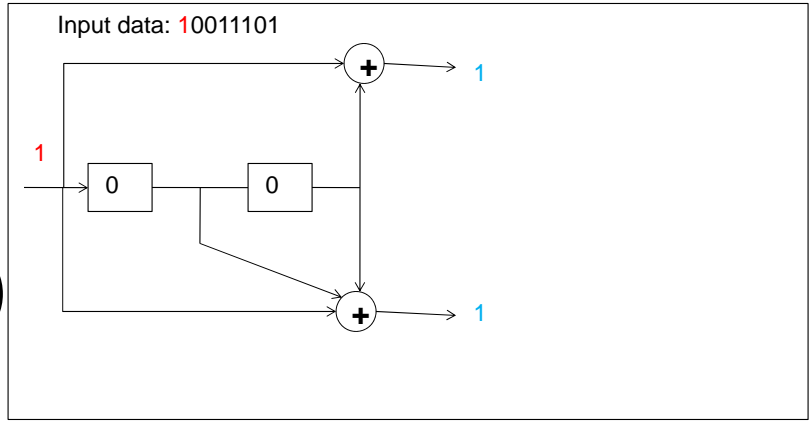
1

2

3

4

5



Instruction for the animator

- After first sentence in DT, red 1 must appear .
- Then the circles should blink and blue 1s must appear.

Text to be displayed in the working area (DT)

- Input 1s given to the encoder and the corresponding outputs of modulo 2 adders are 11.

Step 4:

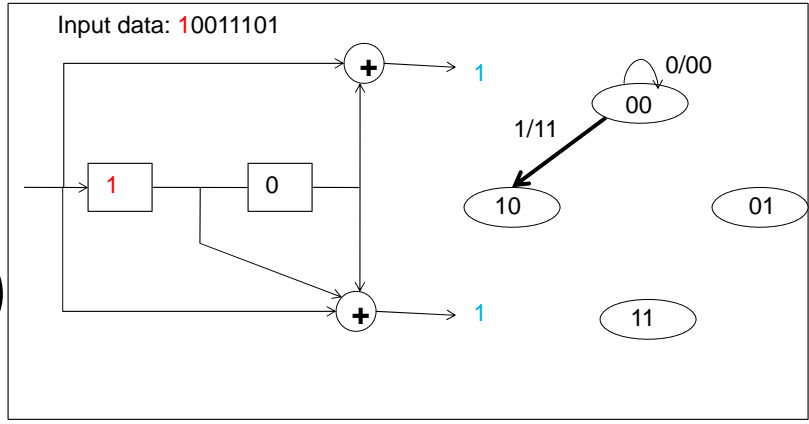
1

2

3

4

5



Instruction for the animator

- Then right shift the bits so that red 1 appears in first box.
- Then the second figure should appear.
- The text in DT should be displayed with the second figure.

Text to be displayed in the working area (DT)

- This is the state diagram.

Step 5:

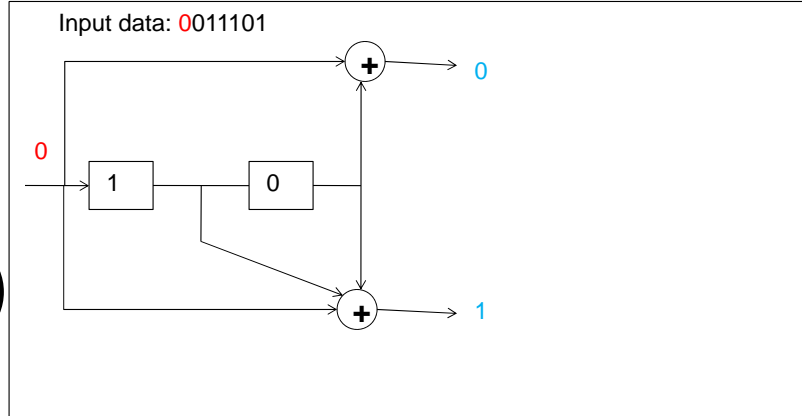
1

2

3

4

5



Instruction for the animator

- After first sentence in DT, red 0 must appear .
- Then the circles should blink and blue 01 must appear.

Text to be displayed in the working area (DT)

- Input 0is given to the encoder and the corresponding outputs of modulo 2 adders are 01.

Step 6:

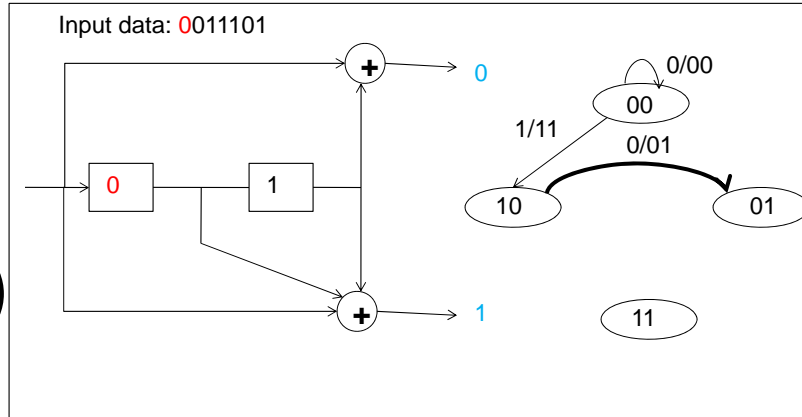
1

2

3

4

5



Instruction for the animator

- Then right shift the bits so that red 0 appears in first box.
- Then the second figure should appear.
- The text in DT should be displayed with the second figure.

Text to be displayed in the working area (DT)

- This is the state diagram.

Step 7:

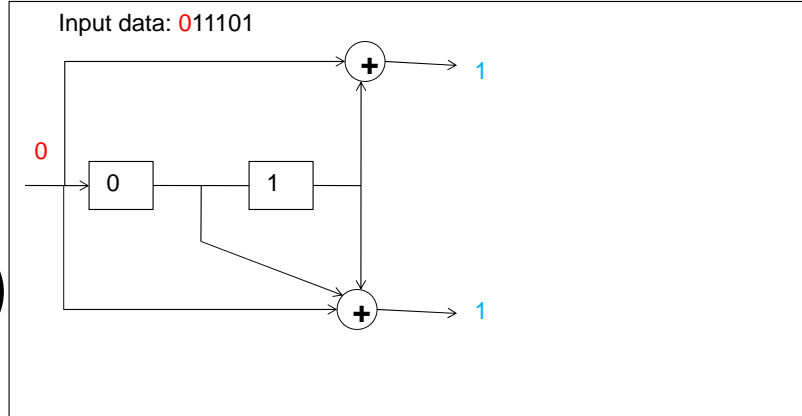
1

2

3

4

5



Instruction for the animator

- After first sentence in DT, red 0 must appear .
- Then the circles should blink and blue 1s must appear.

Text to be displayed in the working area (DT)

- Input 0is given to the encoder and the corresponding outputs of modulo 2 adders are 11.

Step 8:

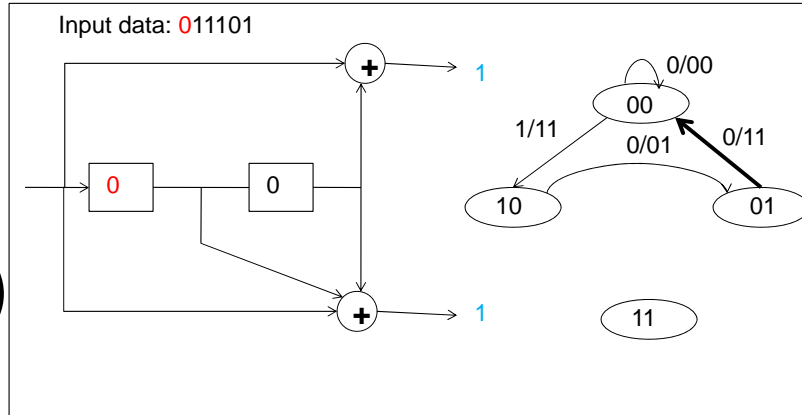
1

2

3

4

5



Instruction for the animator

- Then right shift the bits so that red 0 appears in first box.
- Then the second figure should appear.
- The text in DT should be displayed with the second figure.

Text to be displayed in the working area (DT)

- This is the state diagram.

Step 9:

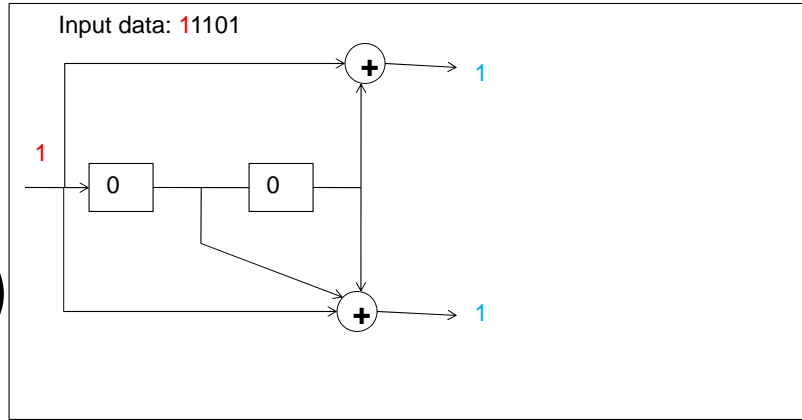
1

2

3

4

5



Instruction for the animator

- After first sentence in DT, red 1 must appear .
- Then the circles should blink and blue 1s must appear.

Text to be displayed in the working area (DT)

- Input 1s given to the encoder and the corresponding outputs of modulo 2 adders are 11.

Step 10:

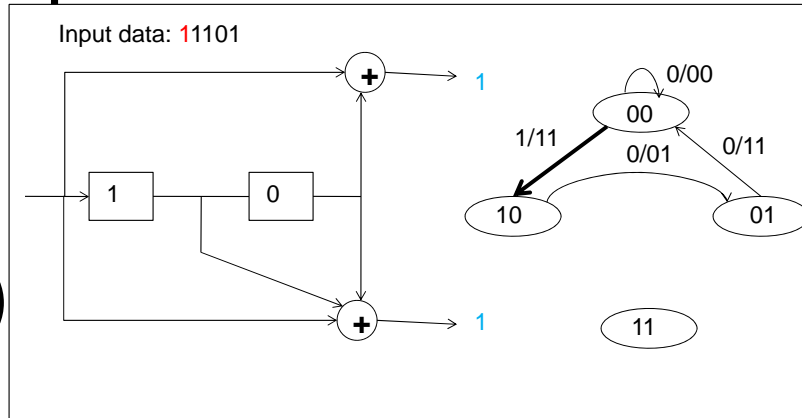
1

2

3

4

5



Instruction for the animator

- Then right shift the bits so that red 0 appears in first box.
- Then the second figure should appear.
- The text in DT should be displayed with the second figure.

Text to be displayed in the working area (DT)

- This is the state diagram.

Step 11:

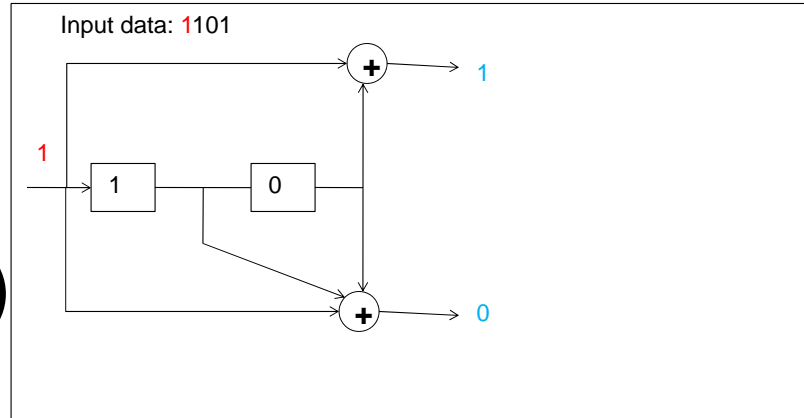
1

2

3

4

5



Instruction for the animator

- After first sentence in DT, red 1 must appear .
- Then the circles should blink and blue 10 must appear.

Text to be displayed in the working area (DT)

- Input 1 is given to the encoder and the corresponding outputs of modulo 2 adders are 10.

Step 12:

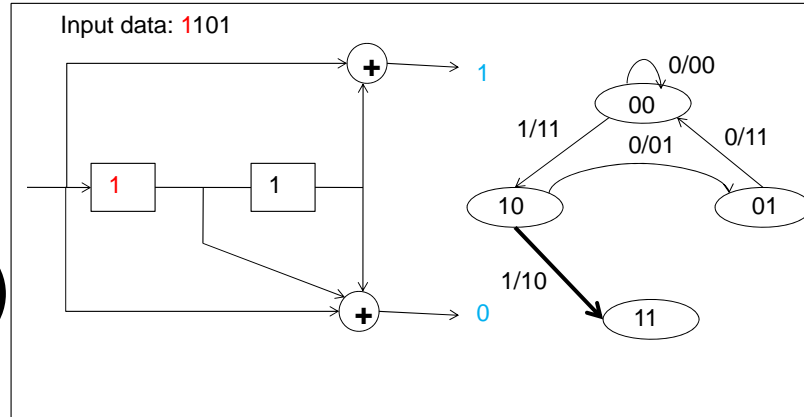
1

2

3

4

5



Instruction for the animator

- Then right shift the bits so that red 1 appears in first box.
- Then the second figure should appear.
- The text in DT should be displayed with the second figure.

Text to be displayed in the working area (DT)

- This is the state diagram.

Step 13:

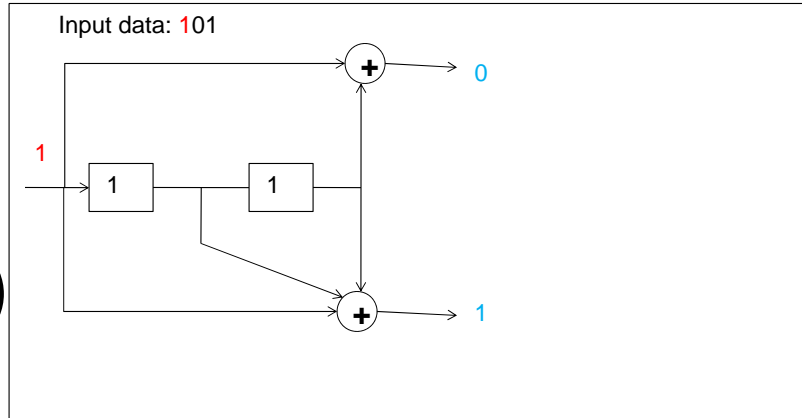
1

2

3

4

5



Instruction for the animator

- After first sentence in DT, red 1 must appear .
- Then the circles should blink and blue 01 must appear.

Text to be displayed in the working area (DT)

- Input 1 is given to the encoder and the corresponding outputs of modulo 2 adders are 01.

Step 14:

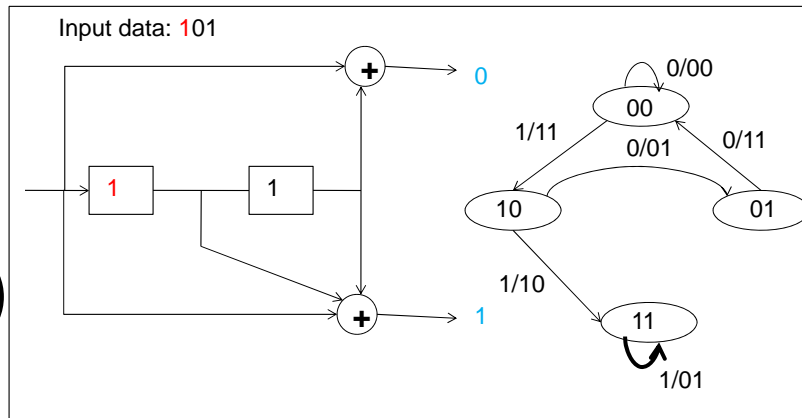
1

2

3

4

5



Instruction for the animator

- Then right shift the bits so that red 1 appears in first box.
- Then the second figure should appear.
- The text in DT should be displayed with the second figure.

Text to be displayed in the working area (DT)

- This is the state diagram.

Step 15:

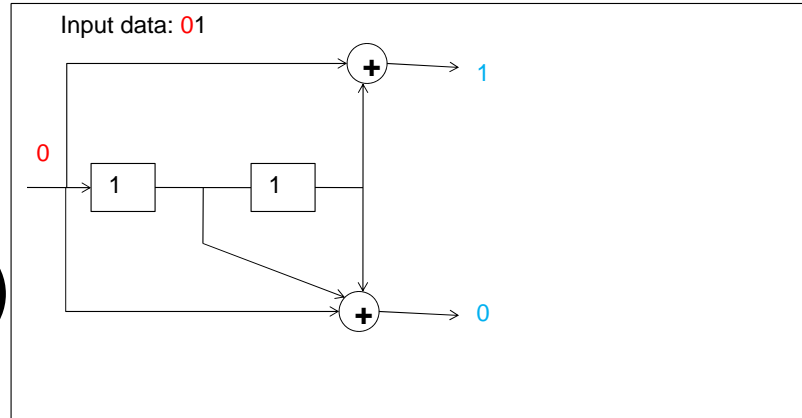
1

2

3

4

5



Instruction for the animator

- After first sentence in DT, red 0 must appear .
- Then the circles should blink and blue 10 must appear.

Text to be displayed in the working area (DT)

- Input 0is given to the encoder and the corresponding outputs of modulo 2 adders are 10.

Step 16:

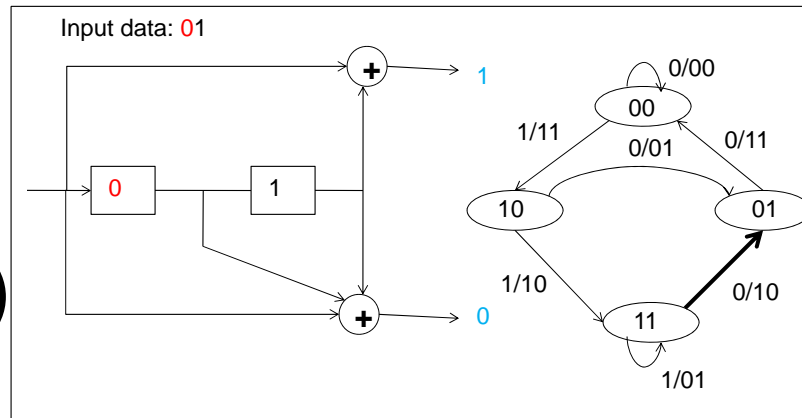
1

2

3

4

5



Instruction for the animator

- Then right shift the bits so that red 0 appears in first box.
- Then the second figure should appear.
- The text in DT should be displayed with the second figure.

Text to be displayed in the working area (DT)

- This is the state diagram.

Step 17:

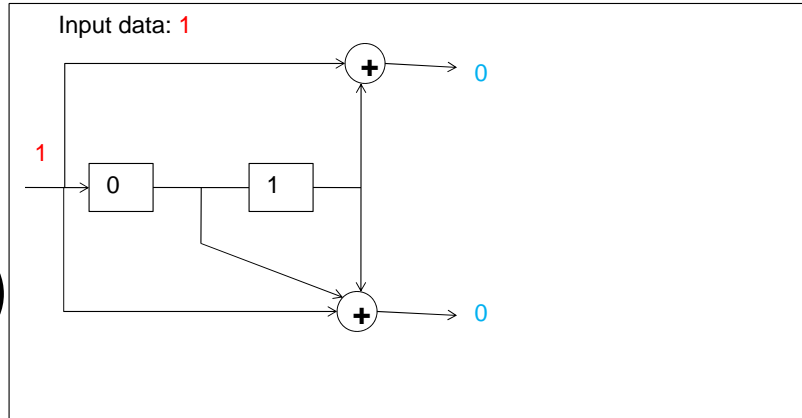
1

2

3

4

5



Instruction for the animator

- After first sentence in DT, red 1 must appear .
- Then the circles should blink and blue 0s must appear.

Text to be displayed in the working area (DT)

- Input 1s given to the encoder and the corresponding outputs of modulo 2 adders are 00.

Step 18:

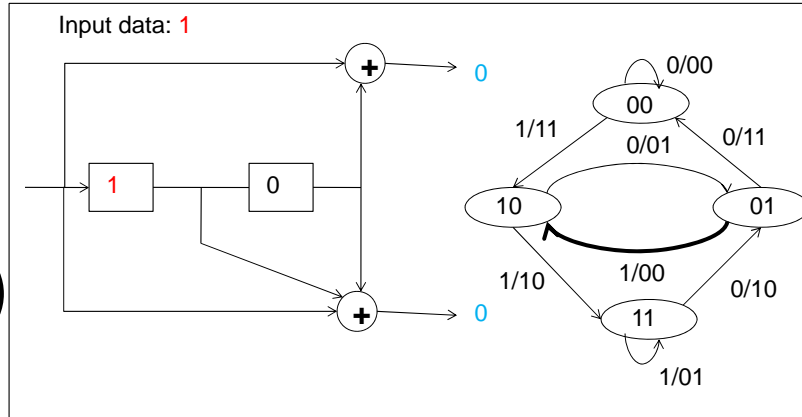
1

2

3

4

5



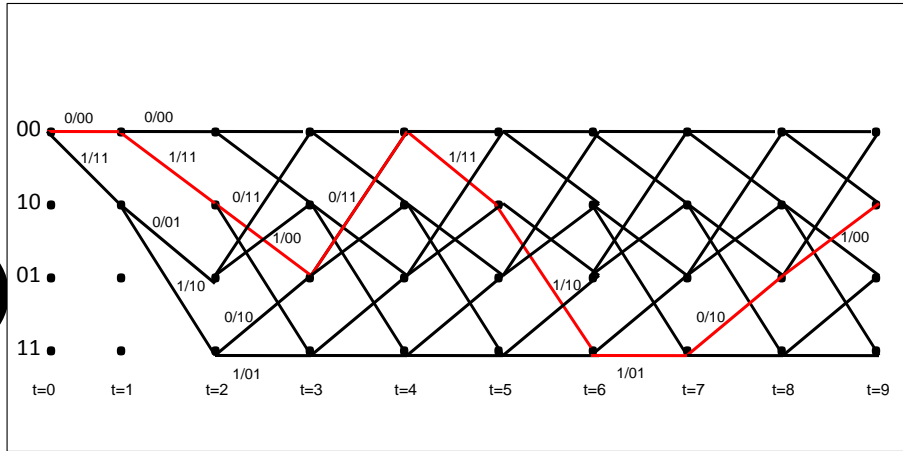
Instruction for the animator

- Then right shift the bits so that red 1 appears in first box.
- Then the second figure should appear.
- The text in DT should be displayed with the second figure.

Text to be displayed in the working area (DT)

- This is the state diagram.

Step 19: Trellis Diagram



Instruction for the animator

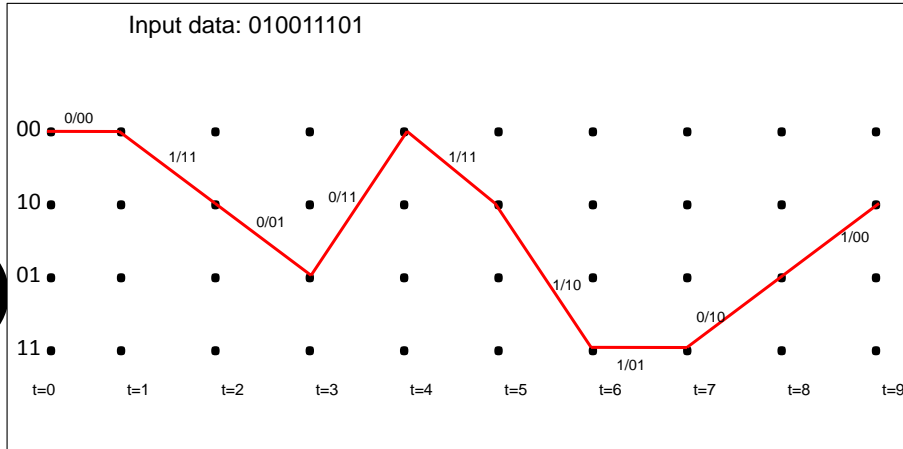
- The figure in step 19 must be shown (red lines should also be shown as black lines).
- Then the red lines must be shown.

Text to be displayed in the working area (DT)

- This is the trellis diagram with all the possible transitions

Step 20:

Input data: 010011101



Instruction for the animator

- The figure in step 20 is shown.
- The text in DT is displayed.

Text to be displayed in the working area (DT)

- The trellis diagram for the given input 010011101
- The output sequence is 00110111100100

Master Layout 2

1

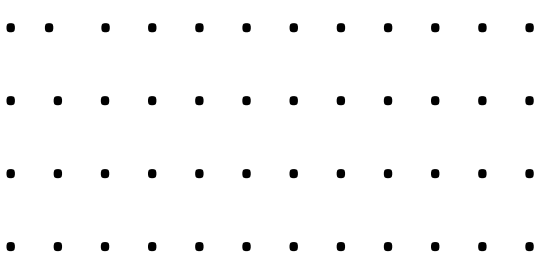
2

3

4

5

Part 1 – Encoder
Part 2 – Decoder



Trellis Diagram

The output sequence is 11, 10, 10, 11, 11, 01, 00, 01

1

2

3

4

5

Step 1:

Instruction for the animator	Text to be displayed in the working area (DT)
	<ul style="list-style-type: none"> Consider an example When the data sequence 1 1 0 0 1 0 1 0 is applied to the encoder, the coded output bit sequence is 11 10 10 11 11 01 00 01. The coded output sequence passes through a channel, producing the received sequence $r = [11\ 10\ \underline{00}\ \underline{10}\ 11\ 01\ 00\ 01]$. The two underlined bits are flipped by noise in the channel.

Step 2:

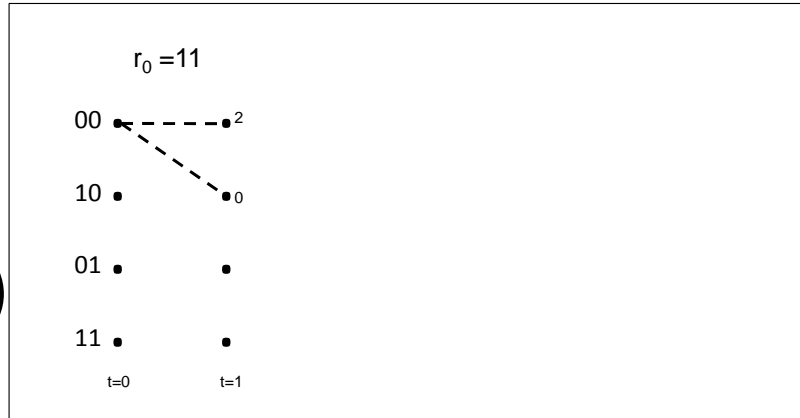
1

2

3

4

5



Instruction for the animator

- Except the dotted lines everything should appear
- After the text in DT is displayed, the dotted lines must appear

Text to be displayed in the working area (DT)

- The received sequence is $r_0 = 11$.
- Computing the metric to each state at time $t=1$ by finding the hamming distance between r_0 and the possible transmitted sequence along the branches of the first stage of the trellis. Since state 0 is the initial state, there are only two paths, with path metrics 2 and 0.

Step 3:

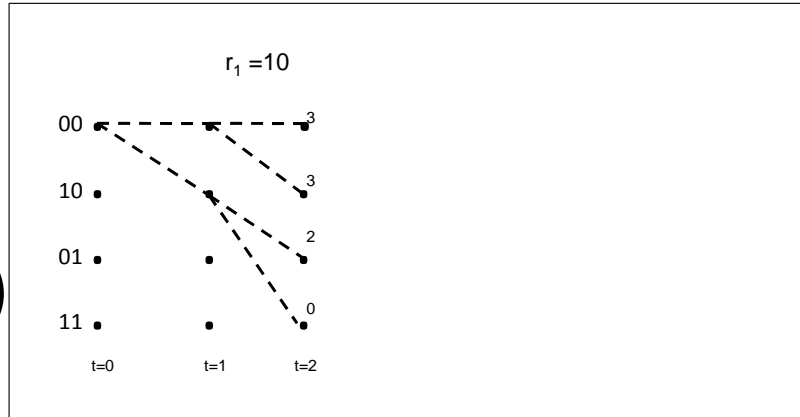
1

2

3

4

5



Instruction for the animator

- Except the new dotted lines everything should appear
- After the text in DT is displayed, the new dotted lines must appear

Text to be displayed in the working area (DT)

- The received sequence is $r_1 = 10$.
- Each path at time $t=1$ is extended, adding the path metric to each branch metric.

Step 4:

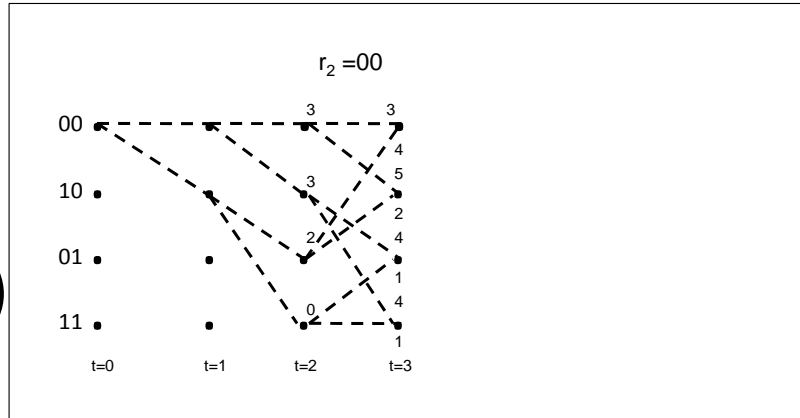
1

2

3

4

5



Instruction for the animator

- Except the new dotted lines everything should appear
- After the text in DT is displayed, the new dotted lines must appear

Text to be displayed in the working area (DT)

- The received sequence is $r_2 = 00$.
- Each path at time $t=2$ is extended, adding the path metric to each branch metric.
- There are multiple paths to each node at time $t=3$

Step 5:

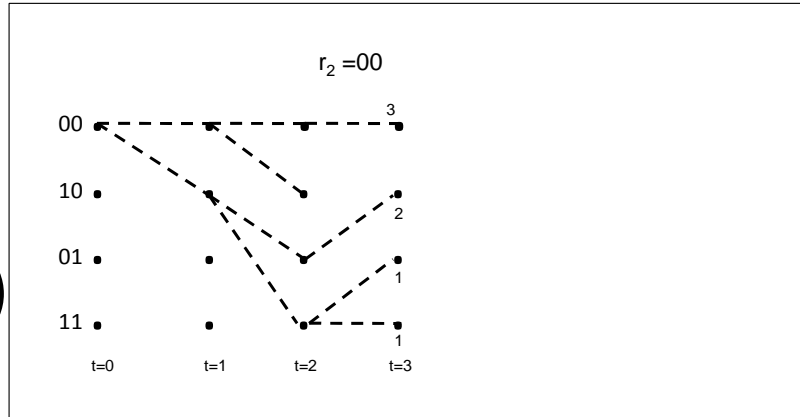
1

2

3

4

5



Instruction for the animator

- After the text in DT is displayed, some dotted lines must disappear and the figure in step 5 is shown.

Text to be displayed in the working area (DT)

- Select the path to each node with best metric and eliminate the other paths.

Step 6:

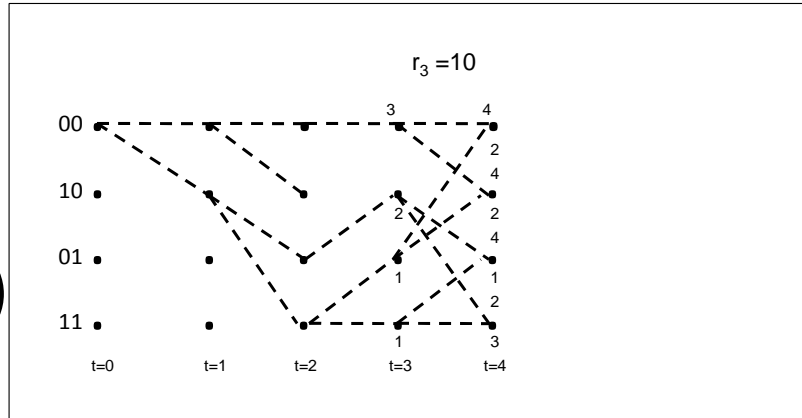
1

2

3

4

5



Instruction for the animator

- Except the new dotted lines everything should appear
- After the text in DT is displayed, the new dotted lines must appear

Text to be displayed in the working area (DT)

- The received sequence is $r_3 = 10$.
- Each path at time $t=3$ is extended, adding the path metric to each branch metric.
- Here, the best path to each state is selected. In selecting the best paths, some of the paths to some states at earlier times have no successors, these paths can be deleted now.

Step 7:

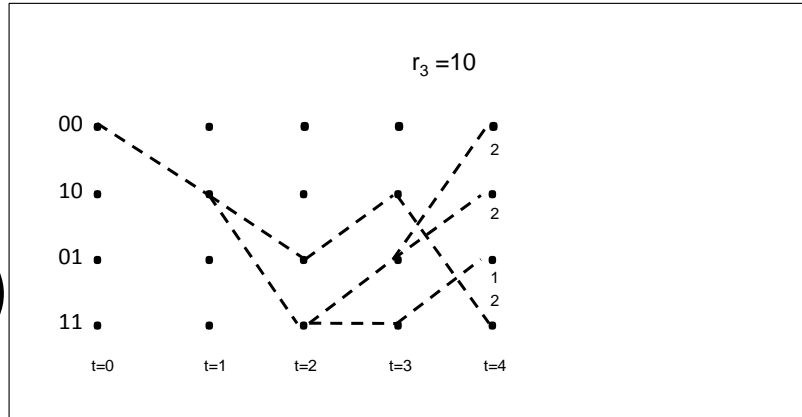
1

2

3

4

5



Instruction for the animator

- After the text in DT is displayed, some dotted lines must disappear and the figure in step 7 is shown.

Text to be displayed in the working area (DT)

- This is the resulting figure after selecting the best paths.

Step 8:

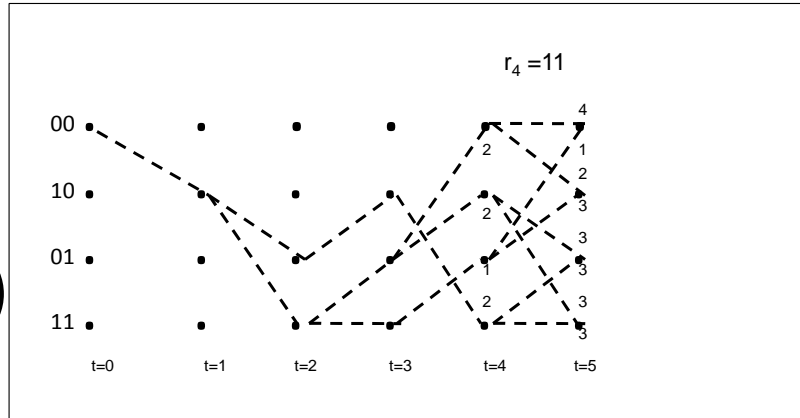
1

2

3

4

5



Instruction for the animator

- Except the new dotted lines everything should appear
- After the text in DT is displayed, the new dotted lines must appear

Text to be displayed in the working area (DT)

- The received sequence is $r_4 = 11$.
- Each path at time $t=4$ is extended, adding the path metric to each branch metric.
- In this case, there are multiple paths into states 2 and 3 with same path metrics but only one of the paths must be selected. So the choice can be made arbitrarily.

Step 9:

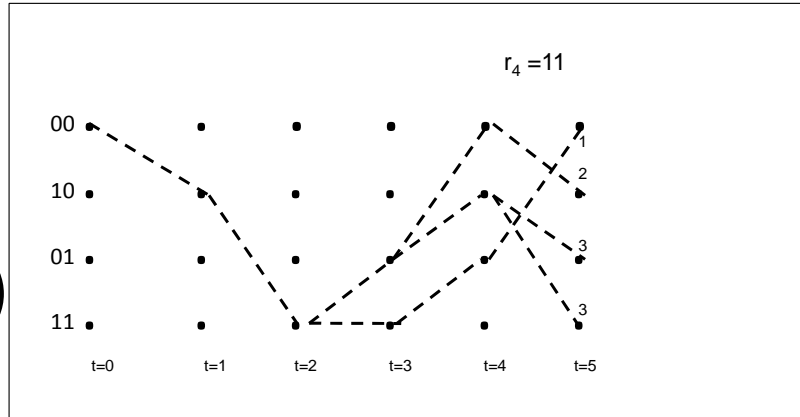
1

2

3

4

5



Instruction for the animator

- After the text in DT is displayed, some dotted lines must disappear and the figure in step 9 is shown.

Text to be displayed in the working area (DT)

- This is the resulting figure after selecting the best paths.

Step 10:

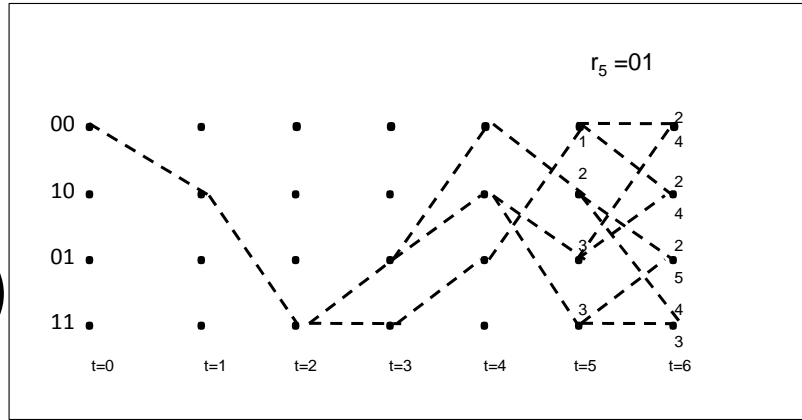
1

2

3

4

5



Instruction for the animator

- Except the new dotted lines everything should appear
- After the text in DT is displayed, the new dotted lines must appear

Text to be displayed in the working area (DT)

- The received sequence is $r_5 = 01$.
- Each path at time $t=5$ is extended, adding the path metric to each branch metric.

Step 11:

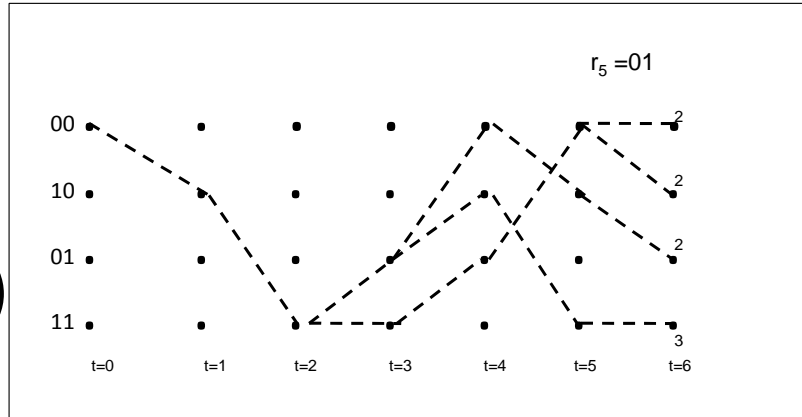
1

2

3

4

5



Instruction for the animator

- After the text in DT is displayed, some dotted lines must disappear and the figure in step 11 is shown.

Text to be displayed in the working area (DT)

- This is the resulting figure after selecting the best paths.

Step 12:

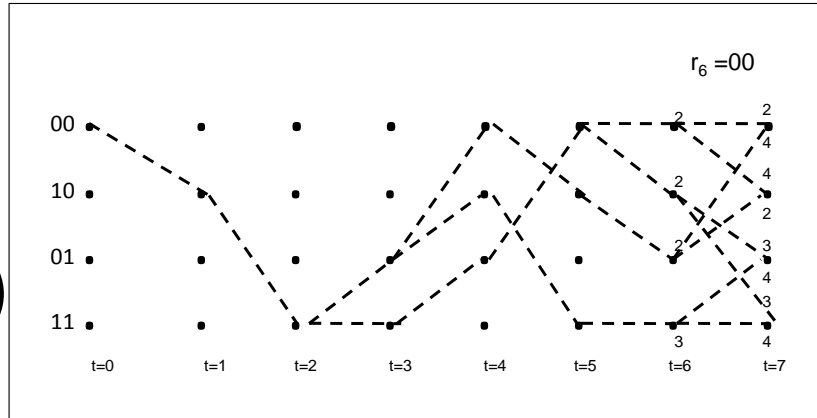
1

2

3

4

5



Instruction for the animator

- Except the new dotted lines everything should appear
- After the text in DT is displayed, the new dotted lines must appear

Text to be displayed in the working area (DT)

- The received sequence is $r_6 = 00$.
- Each path at time $t=6$ is extended, adding the path metric to each branch metric.

Step 13:

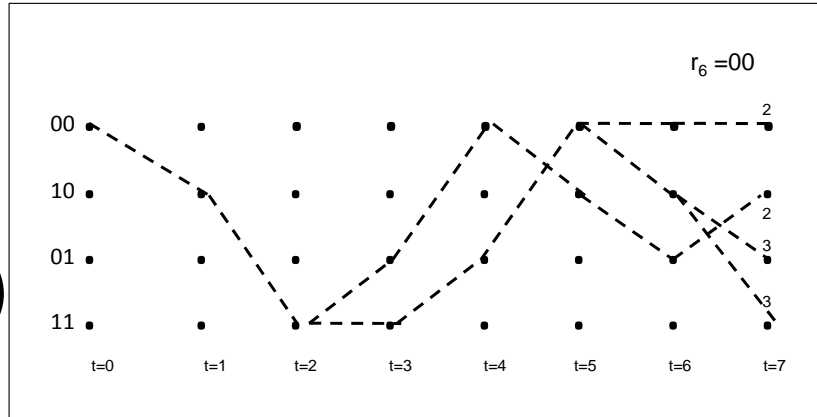
1

2

3

4

5



Instruction for the animator

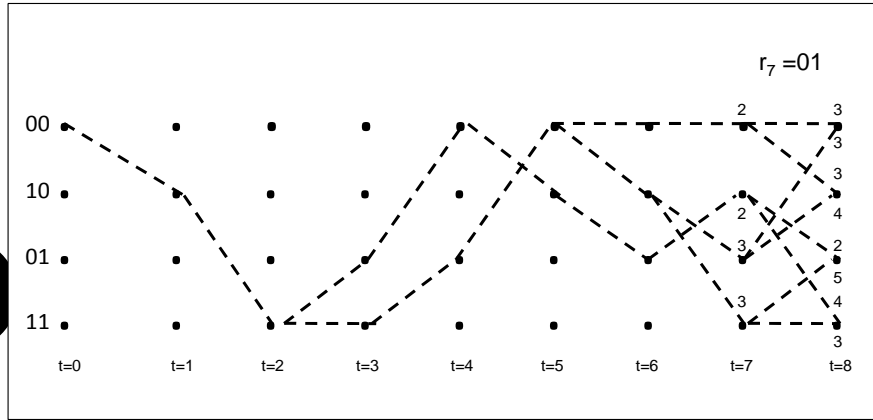
- After the text in DT is displayed, some dotted lines must disappear and the figure in step 13 is shown.

Text to be displayed in the working area (DT)

- This is the resulting figure after selecting the best paths.

Step 14:

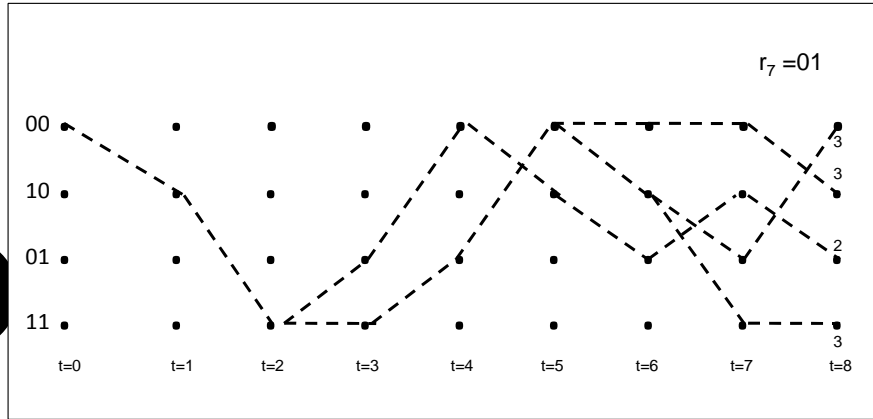
- 1
- 2
- 3**
- 4
- 5



Instruction for the animator	Text to be displayed in the working area (DT)
<ul style="list-style-type: none"> Except the new dotted lines everything should appear After the text in DT is displayed, the new dotted lines must appear 	<ul style="list-style-type: none"> The received sequence is $r_7 = 01$. Each path at time $t=7$ is extended, adding the path metric to each branch metric.

Step 15:

- 1
- 2
- 3**
- 4
- 5



Instruction for the animator	Text to be displayed in the working area (DT)
<ul style="list-style-type: none"> After the text in DT is displayed, some dotted lines must disappear and the figure in step 15 is shown. 	<ul style="list-style-type: none"> This is the resulting figure after selecting the best paths.

Step 16:

1
2
3
4
5

$r_7 = 01$

Instruction for the animator	Text to be displayed in the working area (DT)
<ul style="list-style-type: none"> After the text in DT is displayed, the solid line must be shown. 	<ul style="list-style-type: none"> Selection of the final path with the best metric is done. The input/output pairs are indicated on each branch. The recovered input bit sequence is same as the original bit sequence. Thus, out of the sequence of 16bits, two bit errors have been corrected.

Electrical Engineering
Project OSCAR
Open Source Courseware Animations Repository

Slide 1

Slide 3

Slide 44,45

Slide 47

Slide 46

Introduction
Definitions
Analogy
Test your understanding (questionnaire)
Lets Sum up (summary)
Want to know more... (Further Reading)

Encoder:

Encoder

State Diagram

Interactivity:

Try it yourself

- Place an input box to enter the input data
- Place a dropdown box with options 0 and 1

42

Credits

Electrical Engineering Project OSCAR
Open Source Courseware Animations Repository

Slide 1 Slide 3 Slide 44,45 Slide 47 Slide 46

Introduction Definitions Analogy Test your understanding (questionnaire) Lets Sum up (summary) Want to know more... (Further Reading)

Decoder:

• • • • • • • • • • • • •
 • • • • • • • • • • • • •
 • • • • • • • • • • • • •
 • • • • • • • • • • • • •

Trellis Diagram

Interactivity:

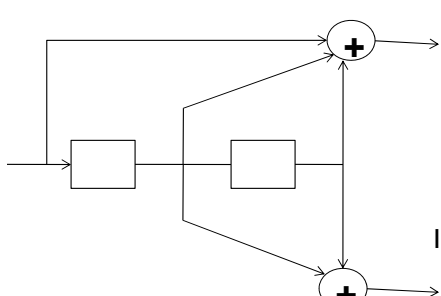
Try it yourself

- Place an input box to enter the 16-bit output data sequence
- Place a dropdown box with options 00, 10, 01, 11

43

Credits

Questionnaire

1. An encoder with n registers will have _____ states.
 Answers: a) 2^n b) $2^n + 1$
2. 

For the given encoder, what will be the output sequence?
 Input data: 01011100
4. Answers: a) 00 11 01 00 10 01 10 11
 b) 00 10 11 10 01 10 00 11
 c) 00 10 11 01 01 10 00 11
 d) 00 11 01 00 01 01 10 11

Questionnaire

1

3. The received sequence is 00 10 11 11 10 01. Among the four choices given below, which is the sequence nearest to the received sequence in terms of Hamming distance?

2

Answers: a) 00 10 01 10 10 01 b) 00 11 01 10 00 01
c) 00 10 11 00 01 10 d) 00 10 11 10 00 11

3

The answers are given in red

4

5

Links for further reading

Reference websites:

http://en.wikipedia.org/wiki/Convolutional_code

Books: Error correction coding – Todd K. Moon,
John Wiley & Sons, INC

Research papers:

Summary

- A convolutional encoder is a finite state machine. An encoder with n binary cells will have 2^n states.
- The most commonly known graphical representation of a code is the trellis representation. A code trellis diagram is simply an edge labeled directed graph in which every path represents a code sequence.
- This representation has resulted in a wide range of applications of convolutional codes for error control in digital communications.
- Viterbi algorithm for decoding a bitstream that has been encoded using forward error correction based on a convolutional code.