Tutorial 3

Exercise 1

The rate $1/2$ convolutional code has generators

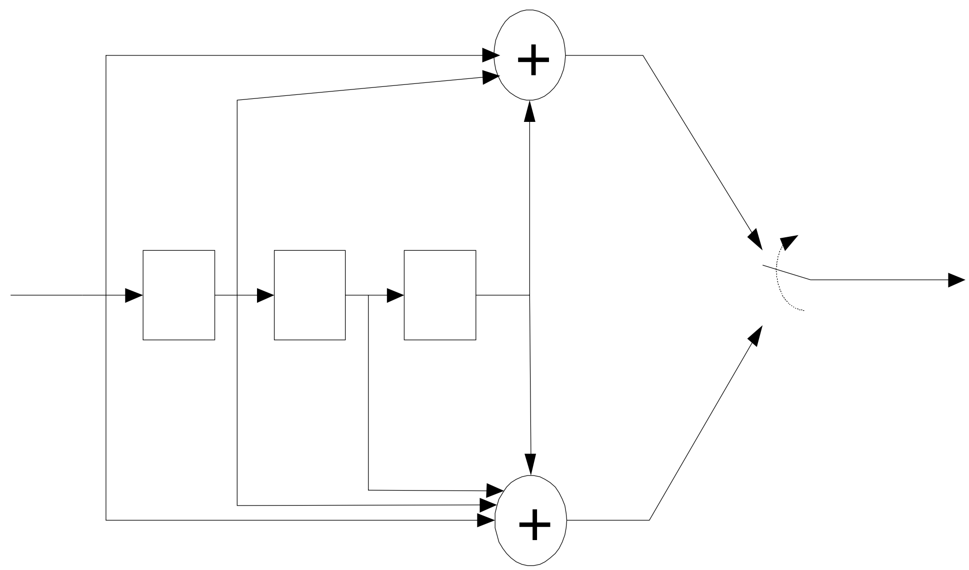
$$g^{(1)} = \begin{pmatrix} 1 & 1 & 0 & 1 \end{pmatrix}$$
$$g^{(2)} = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}.$$

Represent this code as:

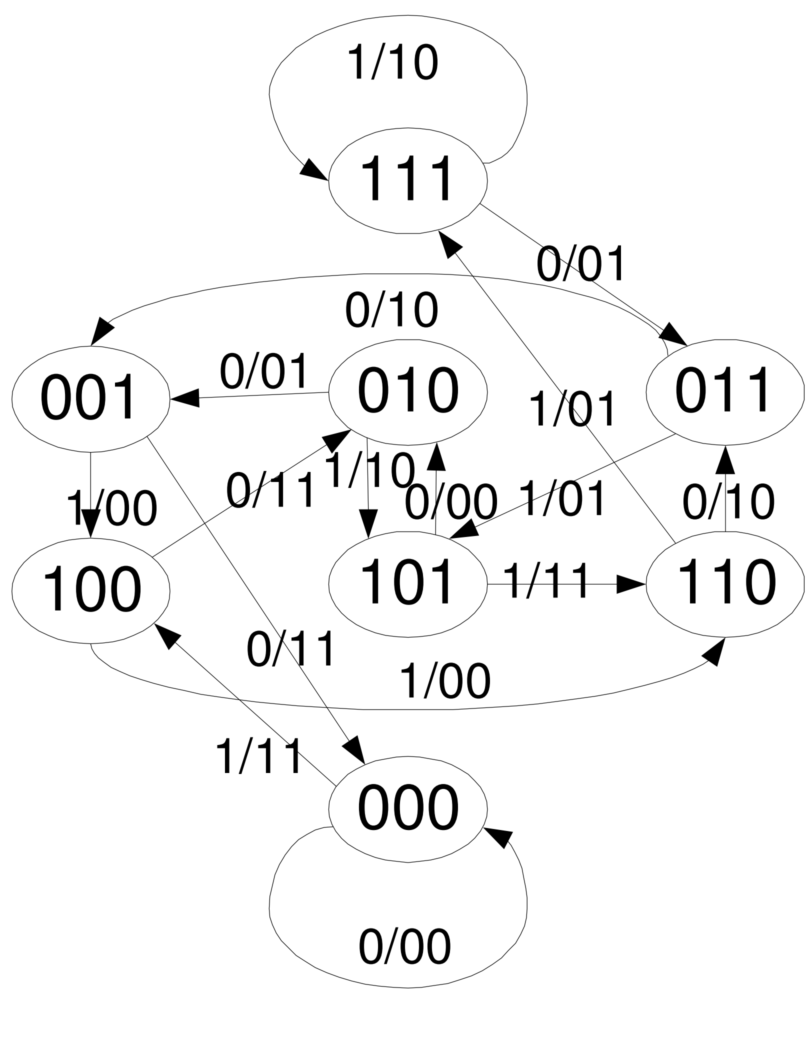
a) Shift register.
b) State diagram.
c) Tree.
d) Trellis.
e) Matrix.

Solution 1

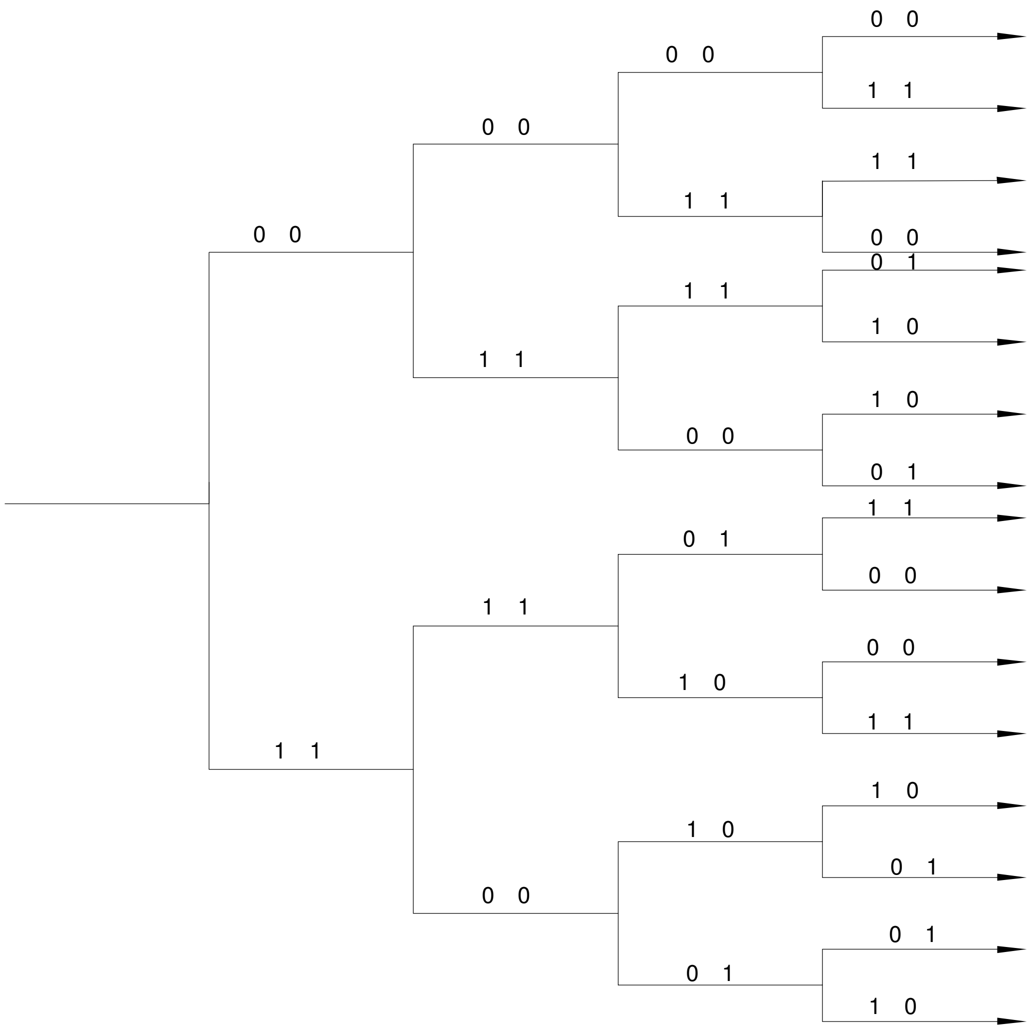
a)



b)

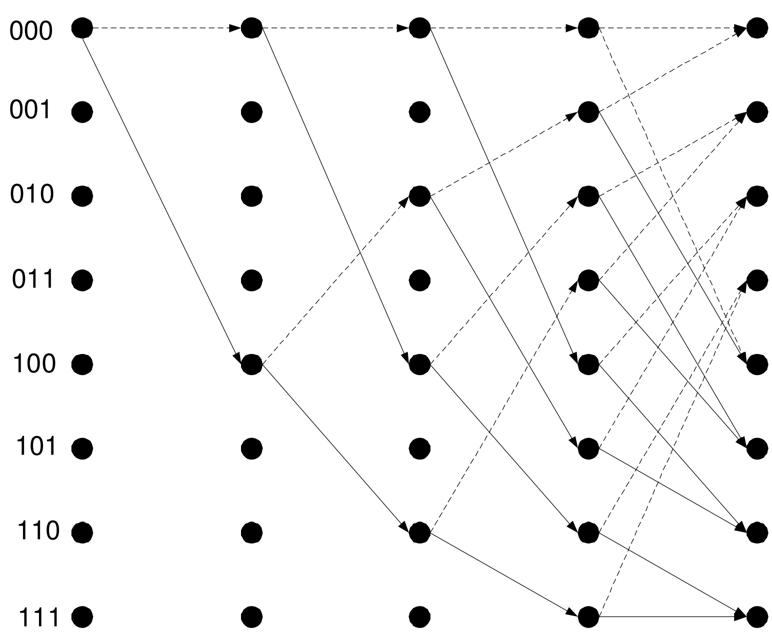


c)



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

d)



000
001
010
011
100
101
110
111

e)

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix}$$

Exercise 2

Represent the code in the previous exercise as recursive systematic convolutional code.
Calculate the output (coded) sequence for both: systematic and corresponding nonsystematic encoder. The information sequence at the input is $u = \begin{bmatrix} 1\ 0\ 1\ 0\ 0 \end{bmatrix}$.

Solution 2

a) recursive systematic convolutional code

$$u(z) = u_0 + u_1 z^1 + u_2 z^2 + \dots$$

$$g^{(1)}(z) = 1 + z^2 + z^3$$

$$g^{(2)}(z) = 1 + z + z^2 + z^3$$

$$s(z) = \begin{bmatrix} u_0 + u_1 z + u_2 z^2 + u_3 z^3 + u_4 z^4 \end{bmatrix} \begin{bmatrix} 1 + z^2 + z^3 \\ 1 + z + z^2 + z^3 \end{bmatrix}$$

$$s(z) = \begin{bmatrix} 1 + z^2 \end{bmatrix} \begin{bmatrix} 1 + z^2 + z^3 \\ 1 + z + z^2 + z^3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 + z^2 + z^3 + z^2 + z^4 + z^5 \\ 1 + z + z^2 + z^3 + z^2 + z^3 + z^4 + z^5 \end{bmatrix} = \begin{bmatrix} 1 + z^3 + z^4 + z^5 \\ 1 + z + z^4 + z^5 \end{bmatrix}$$

$$\left. \begin{aligned} s^{(1)} &= \begin{bmatrix} 1\,0\,0\,1\,1\,1 \end{bmatrix} \\ s^{(2)} &= \begin{bmatrix} 1\,1\,0\,0\,1\,1 \end{bmatrix} \end{aligned} \right\} s = \begin{bmatrix} 1\,1\,0\,1\,0\,0\,1\,0\,1\,1\,1\,1 \end{bmatrix}$$

b) Systematic convolutional code

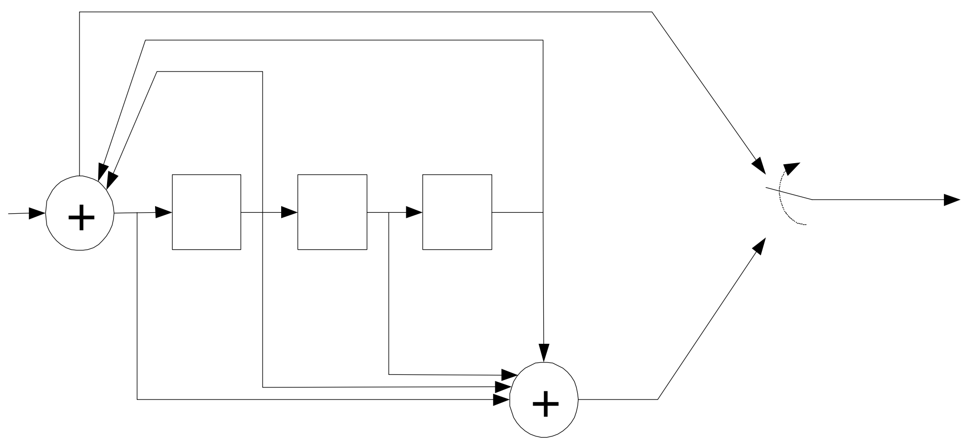
$$s^{(1)} = u(z) \cdot g^{(1)}(z)$$

$$s^{(2)} = u(z) \cdot g^{(2)}(z)$$

Divide both sequences with $g^{(1)}(z)$ and you get new sequence where one contains the systematic bit and the other is generated by a recursive digital filter.

$$s^{(1)} = u(z)$$

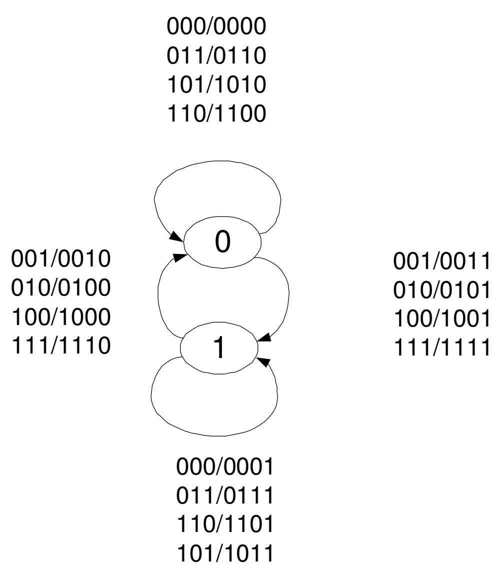
$$s^{(2)} = u(z) \cdot \frac{g^{(2)}(z)}{g^{(1)}(z)}$$



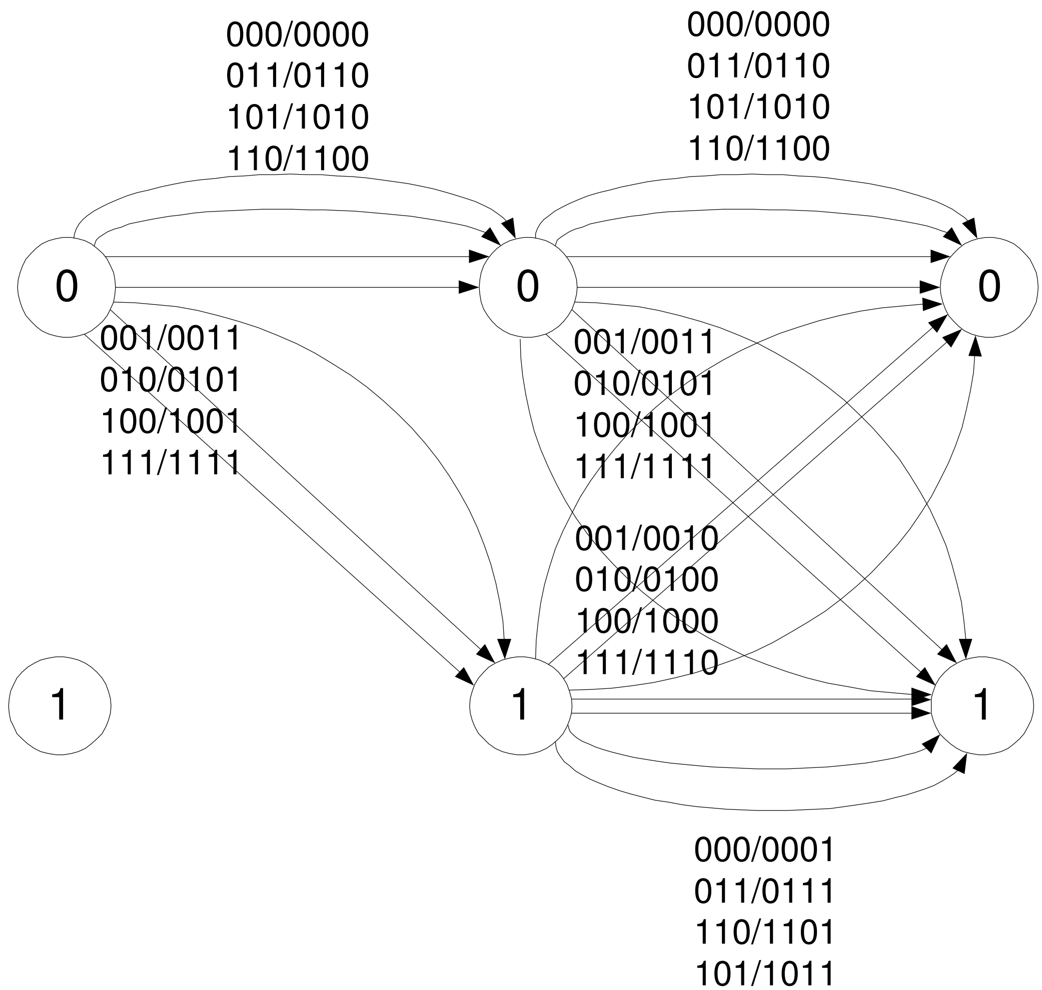
Exercise 3

The encoder starts from zero state, collects three bits and adds to them a parity bit calculated based on the previous state and the bit values. The information bits and parity bits are transmitted. In next interval the process is repeated but the initial state is equal to parity bit value in the next interval. Draw the trellis for given code. (This type of code is called zigzag code).

Solution 3



The state diagram



Trellis of the code

Exercise 4

The encoder block diagram is given in the figure below.

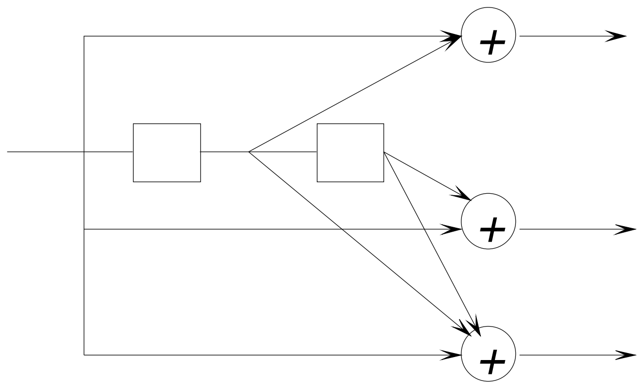
We find for it the impulse response, polynomial description, state space description, trellis description, and how Viterbi algorithm decodes

- An impulse response $g_i^{(j)}$ is obtained for the i-th output of an encoder by applying a single 1 at the j-th input followed by a string of zeros. In our example there is only one input stream $i=1$ and three output steams $j=3$. The impulse responses for the decoder in figure are

$\overline{g}^{(0)} = (110)\ \overline{g}^{(1)} = (101)\ \overline{g}^{(0)} = (111)$.

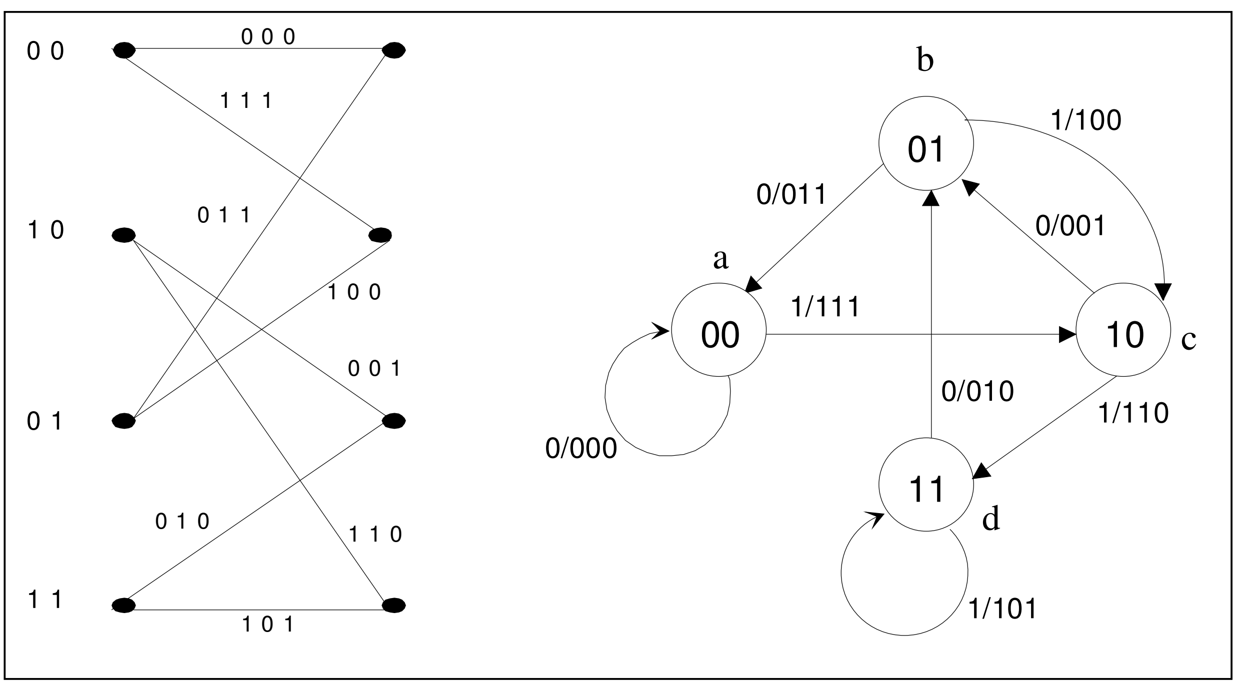
- The transfer function matrix is found by applying the delay transform to the impulse responses. The indeterminate D indicates a delay and its exponent denotes the number of time units the coefficient is delayed.
- The D transform of the impulse response:

$$G(D) = \begin{bmatrix} 1+D & 1+D^2 & 1+D+D^2 \end{bmatrix}$$



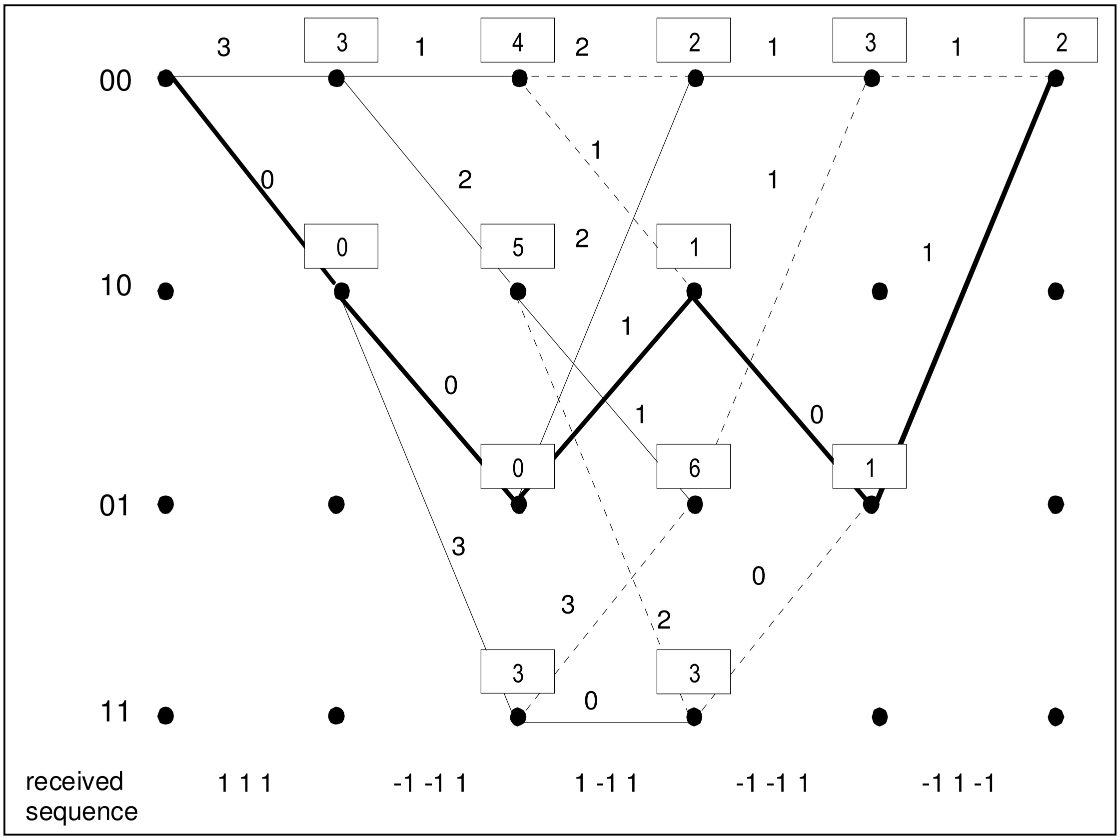
Solution 4

One step of the trellis is shown in the figure below. Compare to the state diagram shown on the right.



Decoding example: Decode the message using hard decision Viterbi algorithm.

- The transmitted bits were 111 001 011 000 000. This corresponds to message bits 10000.
- An encoded message at the output of the hard decision detector is 101 001 011 110 111
- The shift register values of the encoder are initially 00, and the message is terminated with a stream of zero bits.
- At the end encoder is filled with a stream of zeros which drives the decoder to known end state.

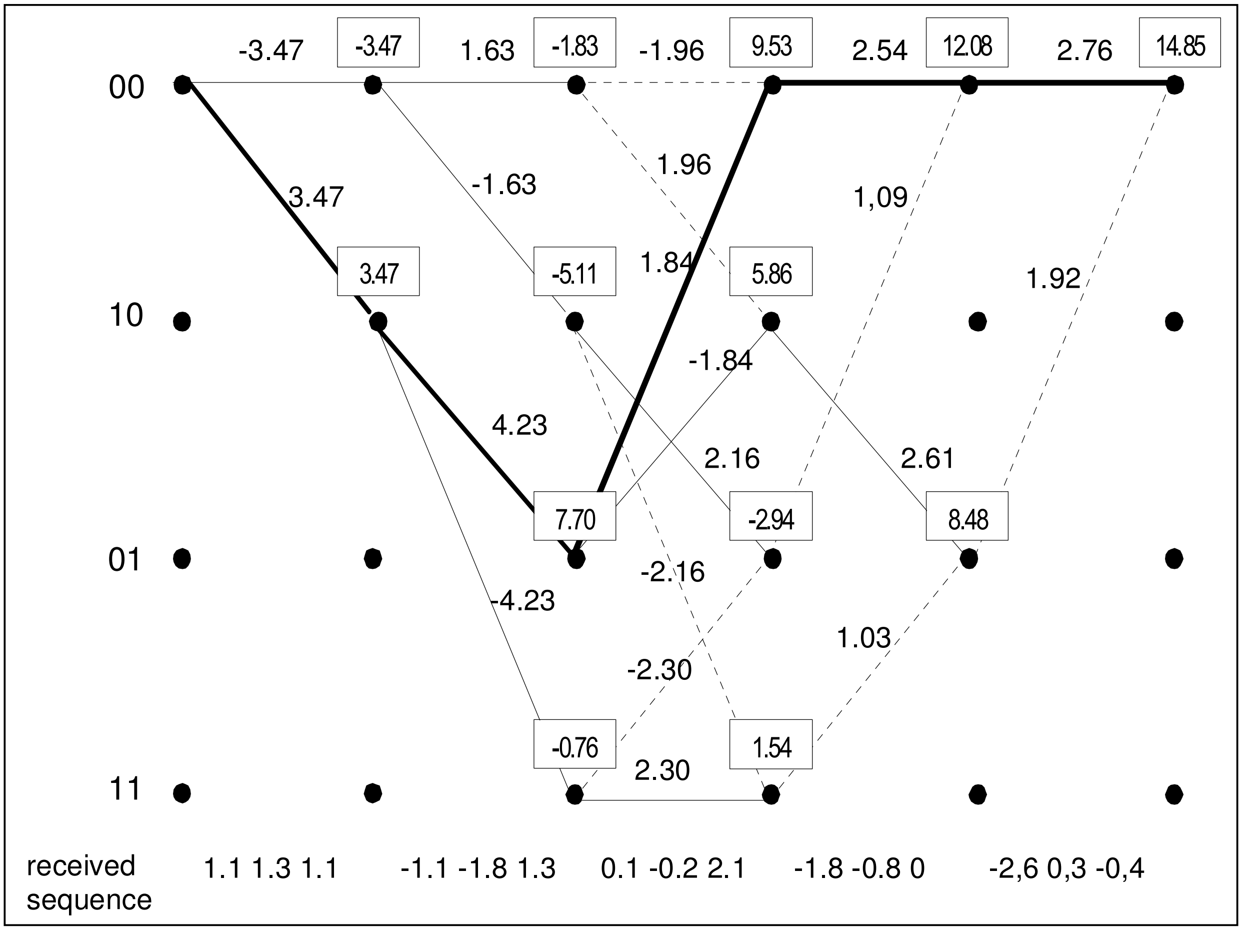


- Hard Decision Viterbi decoder: The transmitted sequence is compared to the candidate sequences and only hard decision of the decision are provided (1 0)
- Solid lines depict branch survivors, broken lines terminating branches. In case the metrics are equal we choose the lower path. The final survivor path shown in bold.

Soft Decision Viterbi decoder

- The received message at the output of a soft decision detector

y = [1.1139 1.2993 1.0568 -1.0944 -1.8377 1.2943 0.0662 -0.1643 2.0667 -1.8236 -0.7580 0.0334 -2.5940 0.2510 -0.4236]



Exercise 5

Describe the probability of a binary number d being equal to +1: $p(d=+1)$ trough the loglikelihood probability $L(p)=\ln\dfrac{p(d=+1)}{p(d=-1)}$

Solution 5

Because d is the binary variable $p(d=-1)=1-p(d=+1)$.

We can express

$$L(p)=\ln\frac{p(d=+1)}{1-p(d=+1)}$$

$$e^{L(p)}=\frac{p(d=+1)}{1-p(d=+1)}$$

$$e^{L(p)}\big(1-p(d=+1)\big)=p(d=+1)$$

$$p(d=+1)=\frac{e^{L(p)}}{1+e^{L(p)}}$$

Exercise 6

Consider the encoder on Figure below.

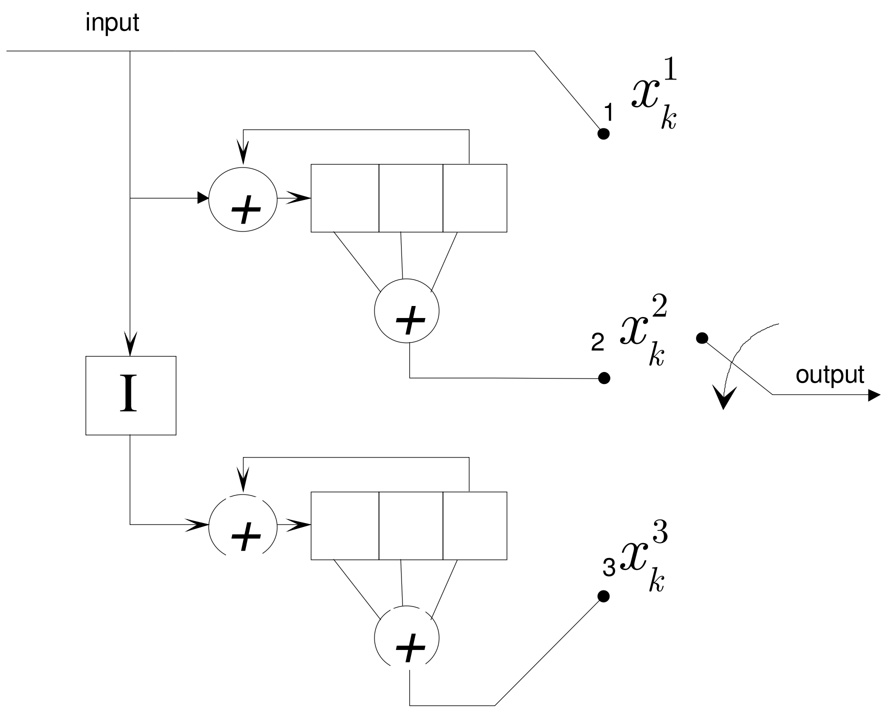
Describe the decoders by the state diagram.

Assume that the I is a 3×3 block interleaver. Derive the output sequence c_k of the encoder if the input data sequence is

$$d_k = \left[1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\right].$$

Give also the input sequence to the second encoder. (The data sequence after interleaver.)

Describe what kind of impact the two last bits of the input sequence have on the termination of the trellis of each encoder.

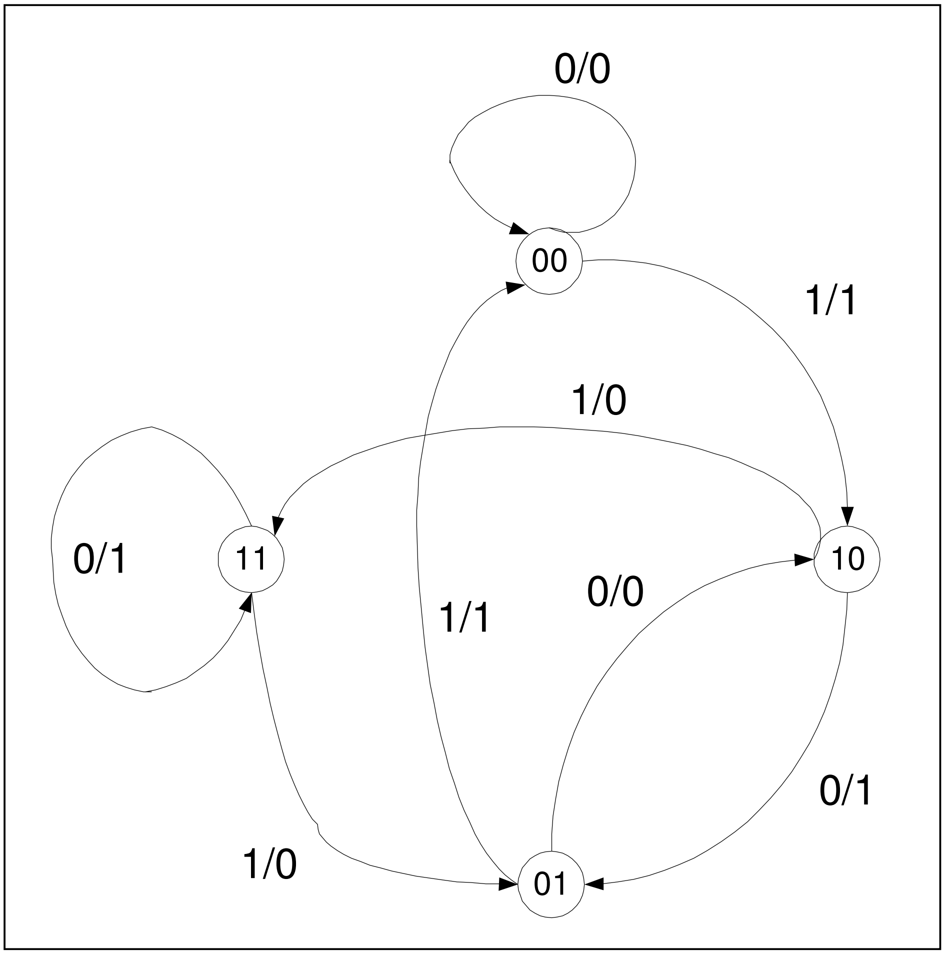


Solution 6

The bit sequence at the interleaver output is

$$d_k = \left[1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\right] \Rightarrow d_{i,k} = \left[1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1\right]$$

The encoder state diagram is



The encoder outputs are

$x_k^1 = \left[1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\right]$

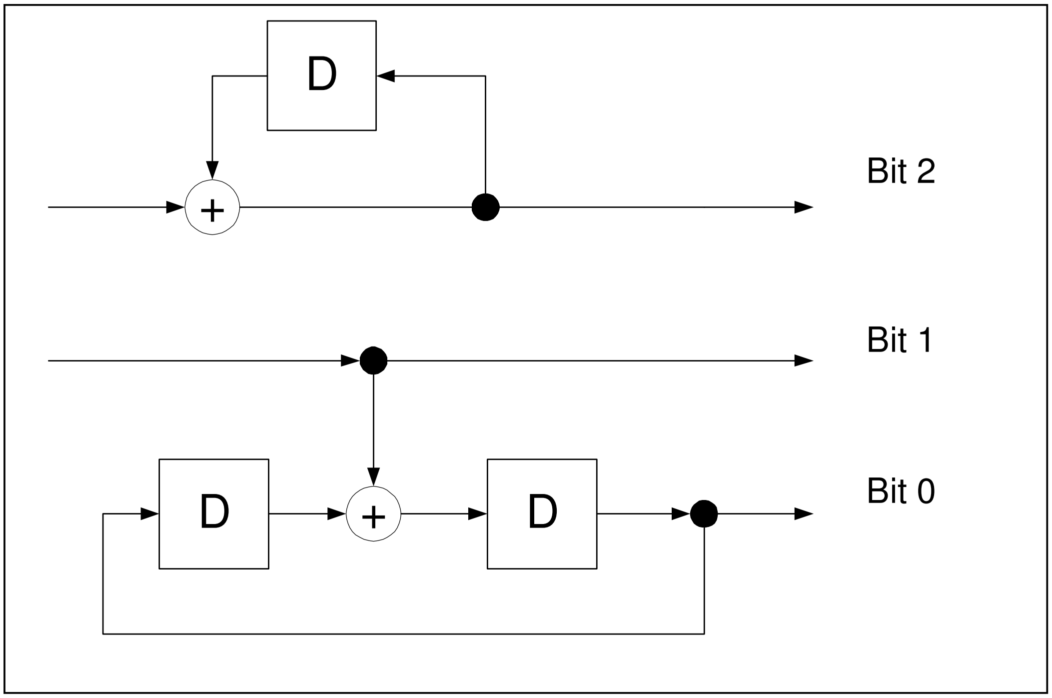
$x_k^2 = \left[1\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\right]$

$x_k^3 = \left[1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\right]$

Two last bits set the encoder to the 00 state. For this example this state is reached by both of the constituent encoders.

Exercise 7

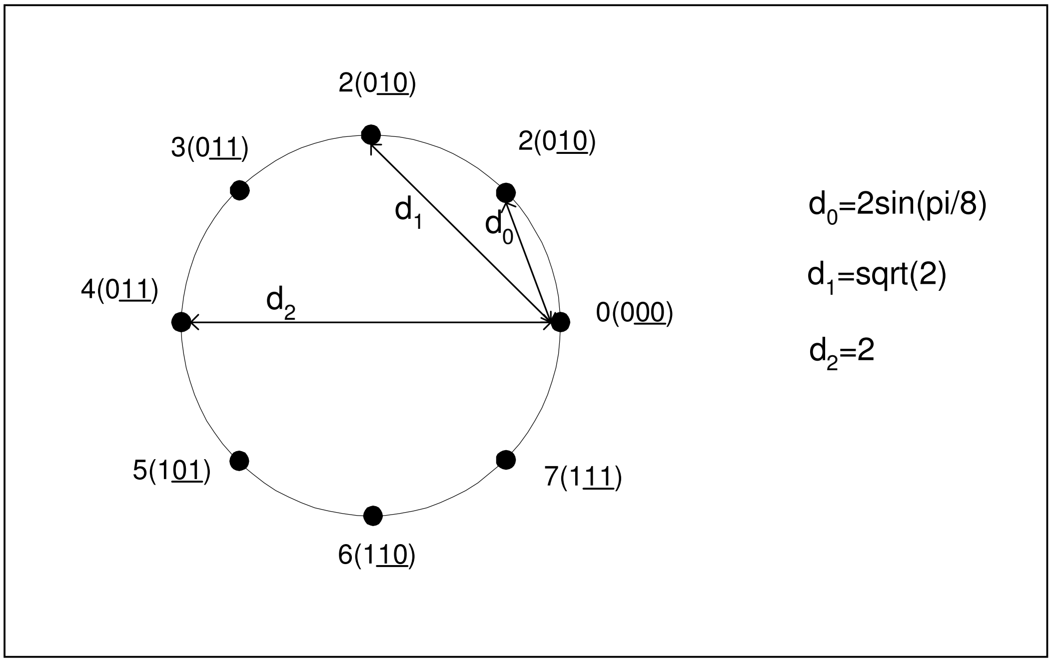
TCM mapping



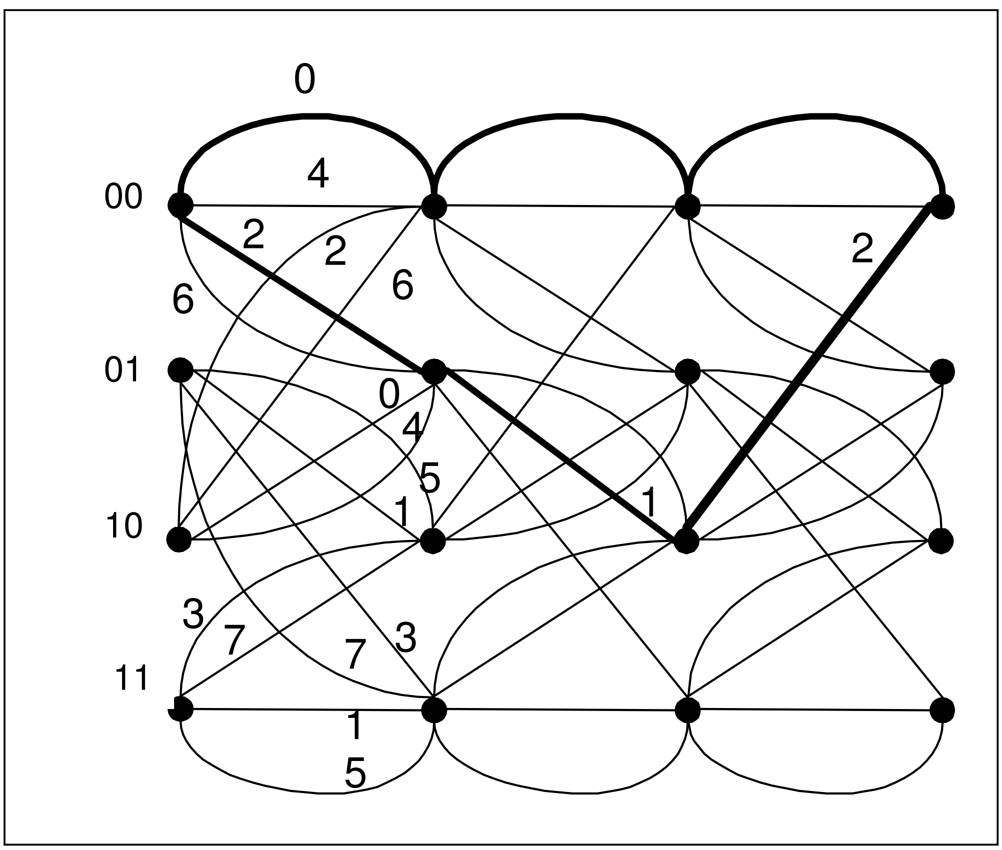
Given the trellis encoder on the figure above.

Label the signals and determine the squared inter signal distance and average signal energy.

Solution 7



The transition trellis of the code is



The minimum free distance is minimum from the parallel paths and diverging paths.

$$d_{free} = \min\left\{d_2; \sqrt{d_1^2 + d_0^2 + d_1^2}\right\}$$

$$= \min\left\{2; \sqrt{2 + \left(2\sin\frac{\pi}{8}\right) + 2}\right\} = 2$$

Exercise 8

By using encoder in previous exercise

Encode the bit sequence [1 0 0 1 1 1]

Draw the path trough the trellis for both decoders.

The transmitted complex bits are $\begin{bmatrix} i & -\frac{1}{\sqrt{2}}(1+i) & i \end{bmatrix}$

the received bit values are

[0.1546 + 1.1110i -0.6984 - 1.1119i 0.5504 + 1.0893i]

Calculate the estimate of the ML sequence by using Viterbi algorirhm

Solution 8

We arrange the input bits into pairs where in the first row is every second bit and in the second row every first bit.

$$\begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{matrix}$$

We feed these pairs of bits into the encoder and generate output

$$\begin{matrix} & \textit{time} & \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{matrix} \Rightarrow 2\,5\,2$$

Calculate the maximum likelihood path probability trough the trellis

Since we know that we start from state 00 our calculations are simplified. We have to evaluate the probabilities only for transitions from state 00.

We are using Viterbi algorithm and accordingly to that sum together the path weights in the trellis. At each merging trellis node we remember only the path that has higher weight.

The weights to the edges are assigned accordingly to the probabilities of corresponding symbols. Since I and Q branches are independent we can calculate the weight as multiplication of two independent Gaussian probabilities.

$$p(x_1, x_2) = \frac{1}{2\pi\sigma} \exp\left(-\frac{(x_1 - m_1)^2}{2\sigma^2} - \frac{(x_2 - m_2)^2}{2\sigma^2} \right)$$

Were we are assuming that the noise variance is same for both branches. Taking logarithm and removing the terms that are equal for every symbol we can express the weight as:

$$\frac{1}{\sigma^2}\left(x_1 m_1 + x_2 m_2\right).$$

We have to calculate this value for each possible transition. Since term $\frac{1}{\sigma^2}$ is scaling constant common for all the symbols we drop it from subsequent calculations.

In first stage we have four possible transitions from 00 state to 00 state with symbols [0 2] and corresponding weights [0.1546 –0.1546] and to state 01 with symbols [4 6] and weights [1.111 –1.111].

In interval 2 we have transitions from state 00 and 01.

00 -> 00

Initial state 00 to	Sumbols	Weight	Weight along the path
00	[0 2]	-0.6984 0.6984	-0.5438 0.8530
01	[4 6]	-1.1119 1.1119	-0.9573 1.2665

Initial state 01 to	Symbols	Weight [symbol1 symbol2]	Weight along the path
10	[1 5]	[-1.2801 1.2801]	[-0.1691 2.3911]
11	[3 7]	[-0.2924 0.2924]	[-0.8186 1.4034]

In the stage 3 we have to calculate the weight for all the transitions. In this stage we have to select also the maximum value from different merging paths. The weights for each symbol are.

0.5504 1.1594 1.0893 0.3811 -0.5504 -1.1594 -1.0893 -0.3811

The weights after merging we get in each state the following weights

Sate	00	01	10	11
Weight	3.48	2.94	2.42	2.56

We see that the weight at the state 00 is highest to this weight corresponds path trough the states [00 01 10 00] and symbol sequence [2 5 2]. This symbol sequence is generated by the bit pairs

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

We see that the decoder has found the correct sequence.