S-72.341 CODING METHODS

Tutorial 8, Solutions

1. (Wicker, problem 11.1) Prove that the convolutional encoder in Figure 1 generates a linear code.

   A code is linear if sum of every two code word is a code word. A code word can be expressed as:

$$y_i^{(j)} = \sum_{l=0}^{2} x_{i-l} g_l^{(i)} .$$

   let $\overline{y}$ and $\overline{y}'$ be two district code words. The sum of these code words is

$$y_i^{(j)} + y_i^{(j)\prime} = \left(\sum_{l=0}^{2} x_{i-l} g_l^{(j)} \right) + \left(\sum_{l=0}^{2} x'_{i-l} g_l^{(j)} \right) = \sum_{l=0}^{2} \left(x_{i-l} + x'_{i-l} \right) g_l^{(i)} .$$

   That is a new valid code word. The code is linear.

2. (Wicker, problem 11.2)
   a) 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 1 are

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

   b) 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:

$$\mathrm{G}(D) = \left[1 + D \quad 1 + D^2 \quad 1 + D + D^2 \right]$$

   c) The output code word is calculated by applying the polynomial of a transfer function G(D) to the polynomial of an input sequence. The polynomial corresponding to the input sequence x=(11101) is

$$\mathrm{X}(D) = \left[1 + D + D^2 + D^4 \right]$$

   Multiplying this with the G(D) we get

$$\mathrm{Y}(D) = \left[1 + D^3 + D^4 + D^5 \quad 1 + D + D^3 + D^6 \quad 1 + D^2 + D^5 + D^6 \right]$$

   The inverse D transform and grouping of output bits gives
   $\overline{y} = (111, 010, 001, 110, 100, 101, 011)$.

d) Given an input sequence of kL bits, the trellis diagram has $L+m$ stages, and code words have length $n(L+m)$.

Code rate is $\dfrac{kL}{n(L+m)}$

The code in Figure 1 has $k=1, m=2, n=3$. Number of input bits is $L=5$

With these parameters the code rate is $\dfrac{5}{3\cdot(5+2)}=0.2381$.

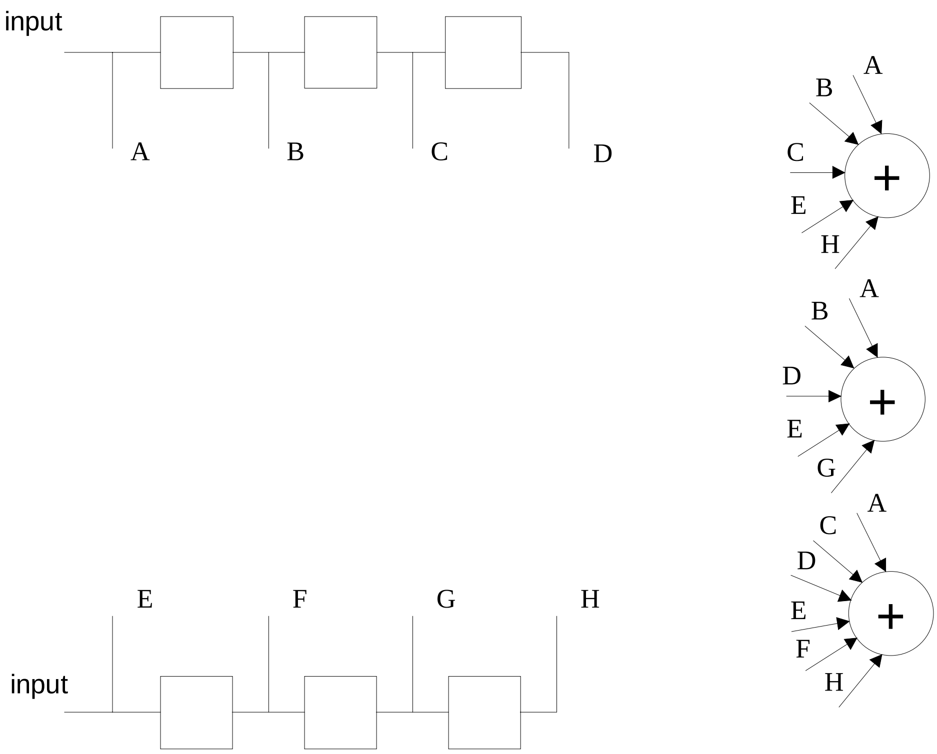
3. (Wicker, problem 11.4)

The transfer functions between each input and output are

$\bar{g}_0^{(0)}=(1110)\ \bar{g}_0^{(1)}=(1101)\ \bar{g}_0^{(2)}=(1011)$

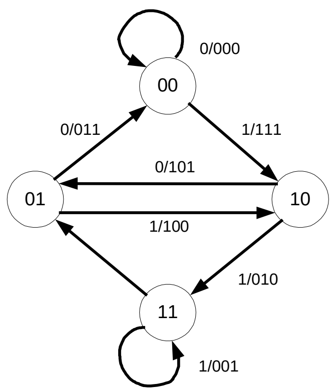
$\bar{g}_1^{(0)}=(1001)\ \bar{g}_1^{(1)}=(1010)\ \bar{g}_1^{(2)}=(1101)$

Each 1 stands for connection between the delay element and output. The decoder is depicted in Figure below.



4. (Wicker, problem 11.7)

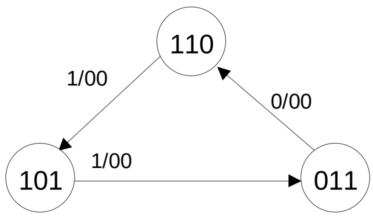
The code is 1/3 code with memory two. It has three states and the state diagram is in Figure below.



5. (Wicker, problem 11.11)

a) A convolutional code is said to be catastrophic if its corresponding state diagram contains a circuit in which a nonzero input sequence corresponds to an all zero output sequence. This conditions can be found also from the transfer functions. A code is not catastrophic if and only if $\text{GCD}\left(G^{(0)}(D),G^{(1)}(D),\ldots,G^{(n-1)}(D)\right)=D^{l}$ for some nonnegative integer l.
The D transforms of the transfer functions are
$\text{G}^{(0)}(D)=D\left(1+D+D^{2}\right)$
$\text{G}^{(1)}(D)=1+D^{3}=(1+D)\left(1+D+D^{2}\right)$
$\text{GCD}\neq D^{l}$ i.e. the described code is a catastrophic code.

b) The loop in a state diagram that for nonzero input generates zero output code word is given in Figure below. The finite weight code word that forces the encoder to proceed along this loop is $x=(11011011011\ldots)$



6. Consider the (3,1,2) code with $G(D)=\left(1+D^{2}\quad 1+D+D^{2}+D^{3}\right)$. Find the GCD of its generator polynomial.

a) The generator polynomials can be expressed as

$G^{(0)}(D) = 1 + D^2$

$G^{(1)}(D) = 1 + D + D^2 + D^3 = (1 + D^2)(1 + D)$

$GCD = 1 + D^2$

b) The code has memory three with 8 corresponding states. Depending on a content of sift registers we number the states as following.

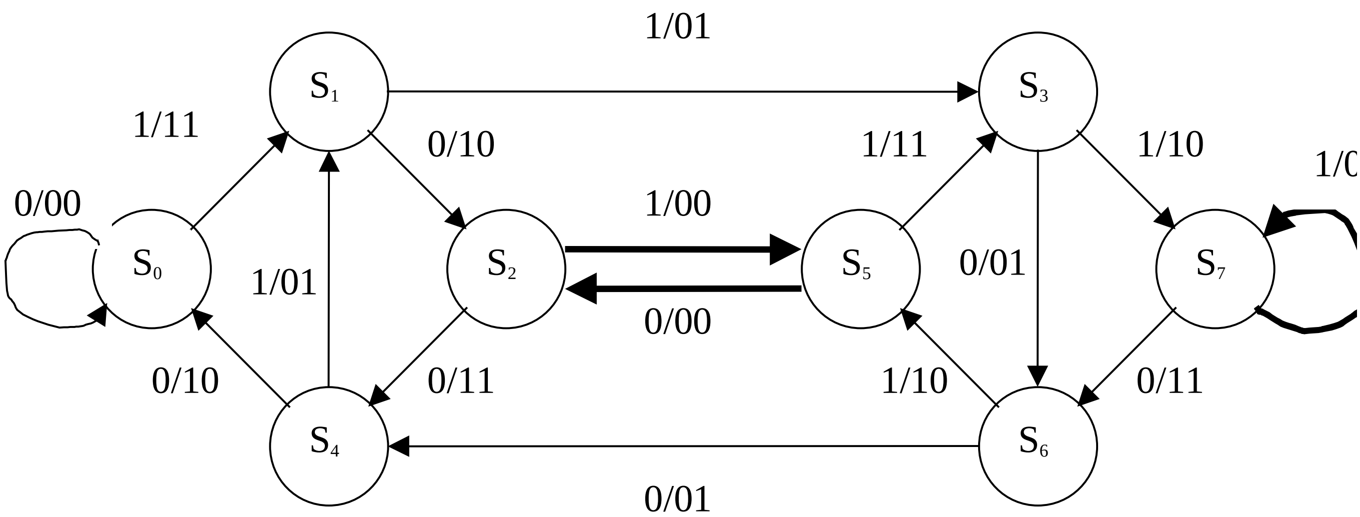
$S_0 \leftrightarrow (000) \quad S_4 \leftrightarrow (001)$

$S_1 \leftrightarrow (100) \quad S_5 \leftrightarrow (101)$

$S_2 \leftrightarrow (010) \quad S_6 \leftrightarrow (011)$

$S_3 \leftrightarrow (110) \quad S_7 \leftrightarrow (111)$

The state diagram of the code is given on Figure below.



c) The code has two loops that generate for nonzero information word the all zero code word.

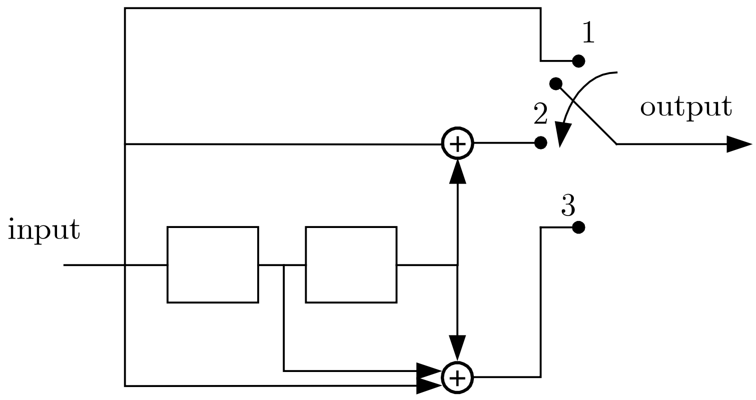
$x_{10} = (1010101010\ldots)$

$x_1 = (1111111111\ldots)$

d) $GCD \neq D^l$ i.e. the described code is a catastrophic code.

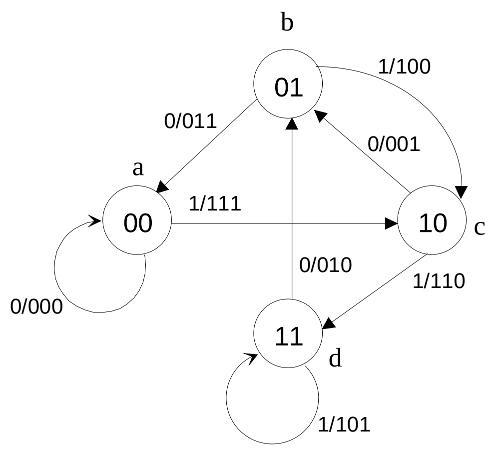
7. Transfer function and minimum free distance.

a) Code polynomials in binary form are $\mathbf{g}_1 = [1 \quad 0 \quad 0]$, $\mathbf{g}_2 = [1 \quad 0 \quad 1]$ ja $\mathbf{g}_3 = [1 \quad 1 \quad 1]$. The encoder is shown below.

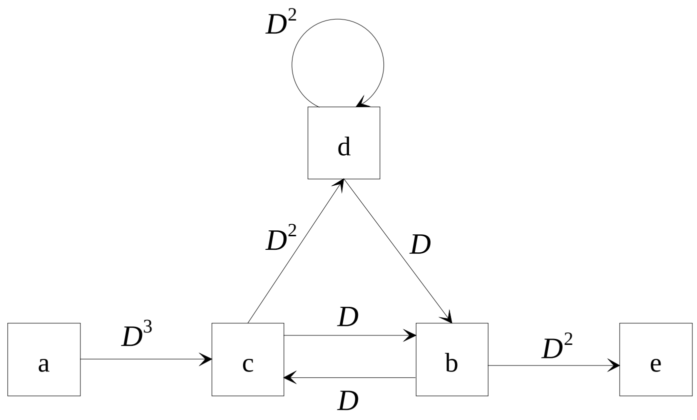


b) The code is systematic, since the output 1 is the same as the input bit.

c) State diagram is shown in the figure below.



The transfer function $T(D)$ can be determined from a modified state diagram. Place the Hamming distance from the all-zero transition to each branch of the state diagram in the form D^d, where d is the Hamming distance. The all-zero transition is divided into two nodes. The transfer function (also called weight distribution generating function) from node a $\rightarrow$ e can be solved by using signal flow graph techniques, or Mason's formula.



In the following we solve $T(D)$ by using the state equations:

$$\begin{aligned} X_c &= D^3 X_a + D X_b \\ X_b &= D X_c + D X_d \\ X_d &= D^2 X_c + D^2 X_d \\ X_e &= D^2 X_b \end{aligned}$$

The transfer function to be solved is

$$T(D) = \frac{X_e}{X_a}.$$

One way is to solve

$$X_b = \frac{D^4 X_a + D X_d}{1 - D^2}$$

and

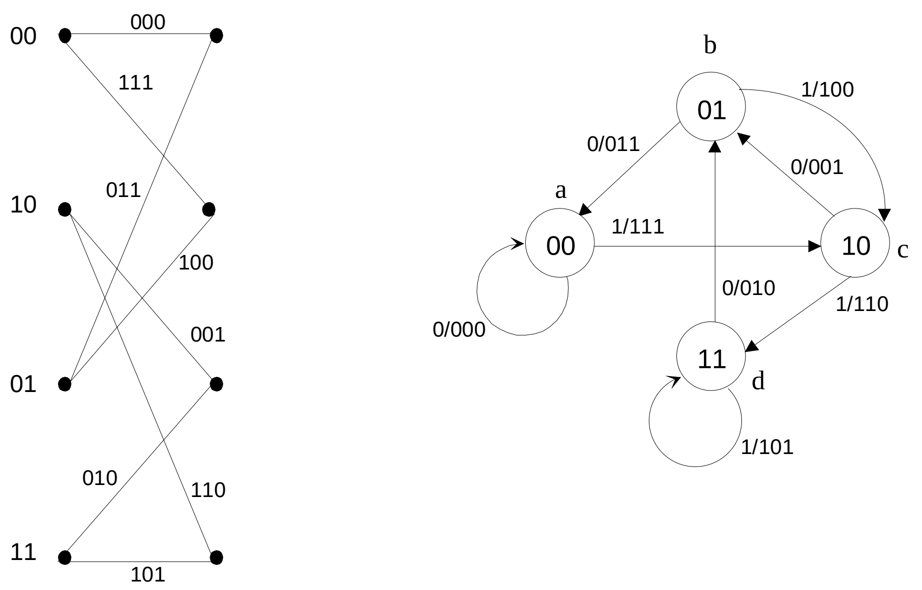
$$X_d = \frac{D^5}{1 - 2D^2} X_a .$$

Inserting these into the bottom equation to yields

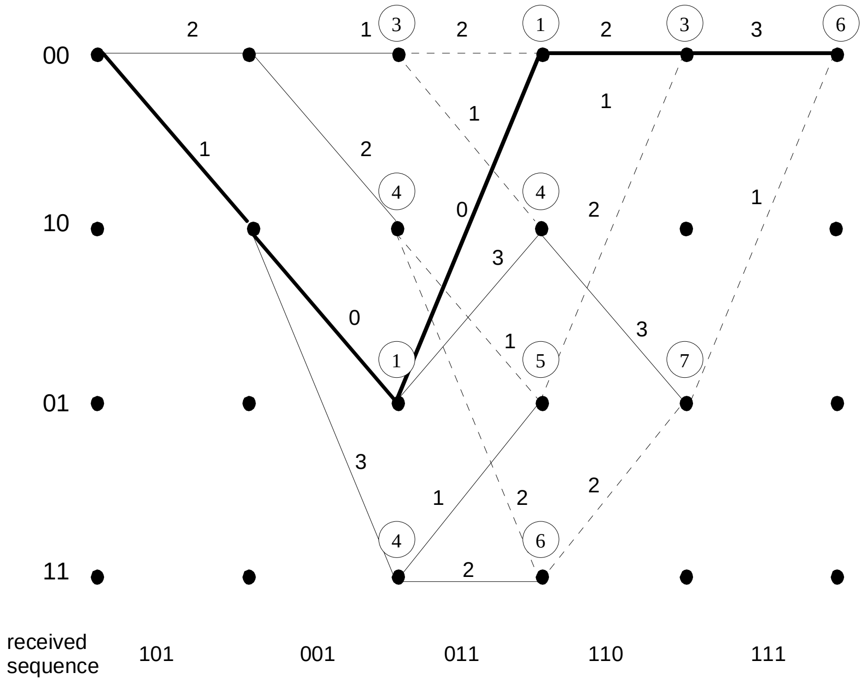
$$
\begin{aligned}
X_e &= \frac{D^2\left(D^4 X_a + D X_d\right)}{1-D^2} \\
&= \frac{D^6}{1-D^2} X_a + \frac{D^3 D^5}{1-2D^2}\frac{1}{1-D^2} X_a \\
&= \frac{X_a}{1-D^2}\frac{D^6 - D^8}{1-2D^2} \\
&= \frac{D^6}{1-2D^2} X_a \\
\Rightarrow T(D) &= \frac{D^6}{1-2D^2} = D^6 + 2D^8 + 4D^{10} + 8D^{12} + \ldots
\end{aligned}
$$

The series reveals the weight distribution of paths diverging from the all-zero path. There is one path of weight 6, two of weight 8 and so on. The smallest weight is called the minimum free distance of the code, d_{free}. The code in this problem has $d_{free} = 6$.

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



Trellis is shown below. Decoding starts at encoder state 00. At the end encoder is filled with a stream of zeros which drives the decoder to known end state. 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.



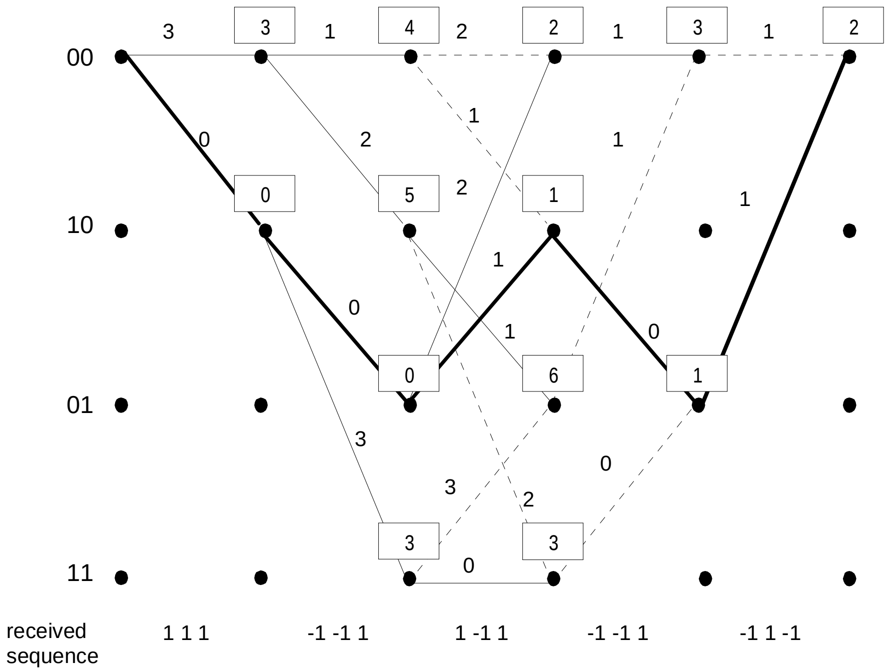
The transmitted bits were 111 001 011 000 000. This corresponds to message bits 10000.

9. A hard decision decoder transfers the input soft values to the hard values. 111 001 101 001 010. Before the decoding we insert into trellis given a priory information: the initial state is 00, and at the end encoder is filled with a stream of zeros which drives the decoder to known end state, 00.

   The principles of the hard decision decoder are described in the previous exercise.

   The decoder minimises the distance between allowed and received sequences. At each state decoder calculate the weight for input branches and selects the one which weight is less. The weight of the branch is sum of the weight at previous state and hamming distance between the expected and received code word.

   Figure below describes values at each decoder state. 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.



The decoder suggest for transmitted sequence 1 0 1 0 0. The total Hamming weight of the path is 2.

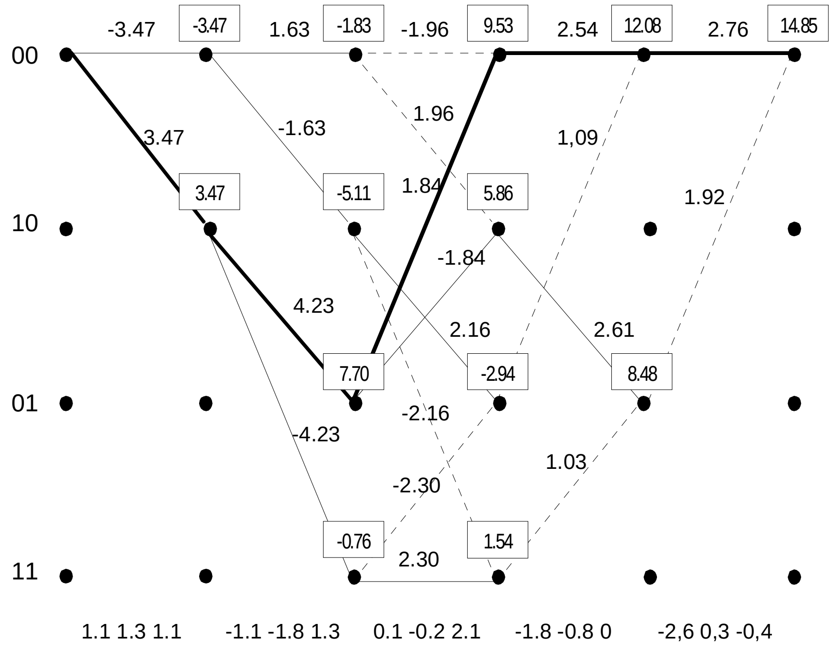
In a soft-decision decoder the path metric is inner product of the received word and the code word.

$$M\left(r_i^{(j)} \mid y_i^{(j)}\right) = r_i^{(j)} y_i^{(j)}$$

Derivation of this equation is given in the book in the chapter 12.2.2.

The decoder is implemented similarly as in hard decision case. Now however the decoder selects the branch that has maximal weight.

The decoding process can be followed in the figure below. The decoder suggest for the transmitted sequence 1 0 0 0 0. Notice that this differs from the result of the hard decision decoding.



The signal to noise ratio is relationship between the signal energy and noise energy. The noise energy is equal to the variance of the noise.

We calculate first the amplitude of the received symbols. By subtracting the amplitude from the symbols we get values of noise with mean zero. From there can be calculated the noise variance per symbol. The variance per bit is sum of the variances per symbols.

The signal energy per bit is 3. For each bit are transmitted sequence of $n = 3$ symbols. An energy per received symbol is $\frac{E_b}{n} = \frac{3}{3} = 1$ and so is the amplitude.

The noise in channel is calculated by subtracting from the received sequence at the detector output the transmitted sequence. The amplitude of the symbols is 1. The calculated variance per symbol is $N = 0.77$ and per bit it is three times of it $N_b = 3 \cdot 0.77$. The signal energy was three and SNR is

$$SNR = \frac{E_b}{N} = \frac{3}{3 \cdot 0.77} = 1.299 \Rightarrow 1.13\, dB .$$