CT216

Introduction to Communication System

Convolution Coding

Lab 3 And Group : 3

Channel coding scheme

Prof. Yash Vasavada
Mentor : Aparna Kumari

Honour code

- **We declare that :**

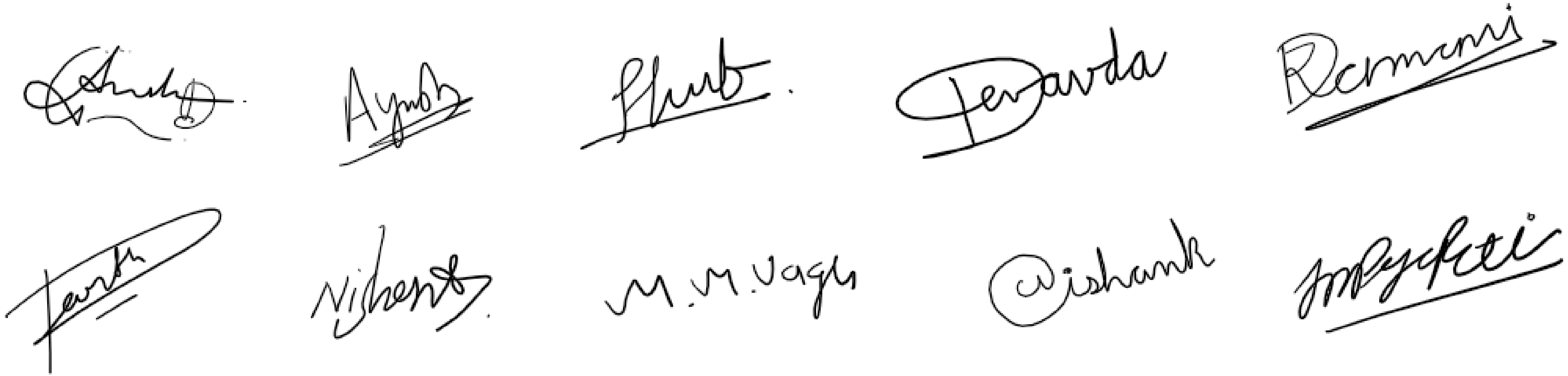
  → The work that we are presenting is our own work.

  → We have not copied the work (the code, the results, etc.) that someone else has done.

  → Concepts, understanding and insights we will be describing are our own.

  → We make this pledge truthfully. We know that violation of this solemn pledge can carry grave consequences.

- **Signed by: all the members of the project group.**



Outline:

- What is Communication
- Introduction to Convolution Code
- Encoding in convolution coding scheme
- BPSK Modulation & AWGN Channel
- Hard-decision Viterbi decoding (HDD)
- Soft-decision Viterbi decoding (SDD)
- Transfer function
- Graph Analysis
- Analytics
- Bibliography

What is Communication?

- Communication is the process of exchange of information (here in form of bits).
- It requires the transmission and receiving of bits with the help of transmitter and receiver respectively.
- It requires the channel (as a medium) through which bits are transmitted
- After the transmission of bits some noise may be introduced due to which the bits may be altered and it may generate errors at the receiver
- To remove the errors and ensure that the correct bits are received, encoding and decoding of message is introduced.
- For the encoding and decoding of the message different types of Coding schemes are introduced.
- Convolution code is one of the famous coding schemes.

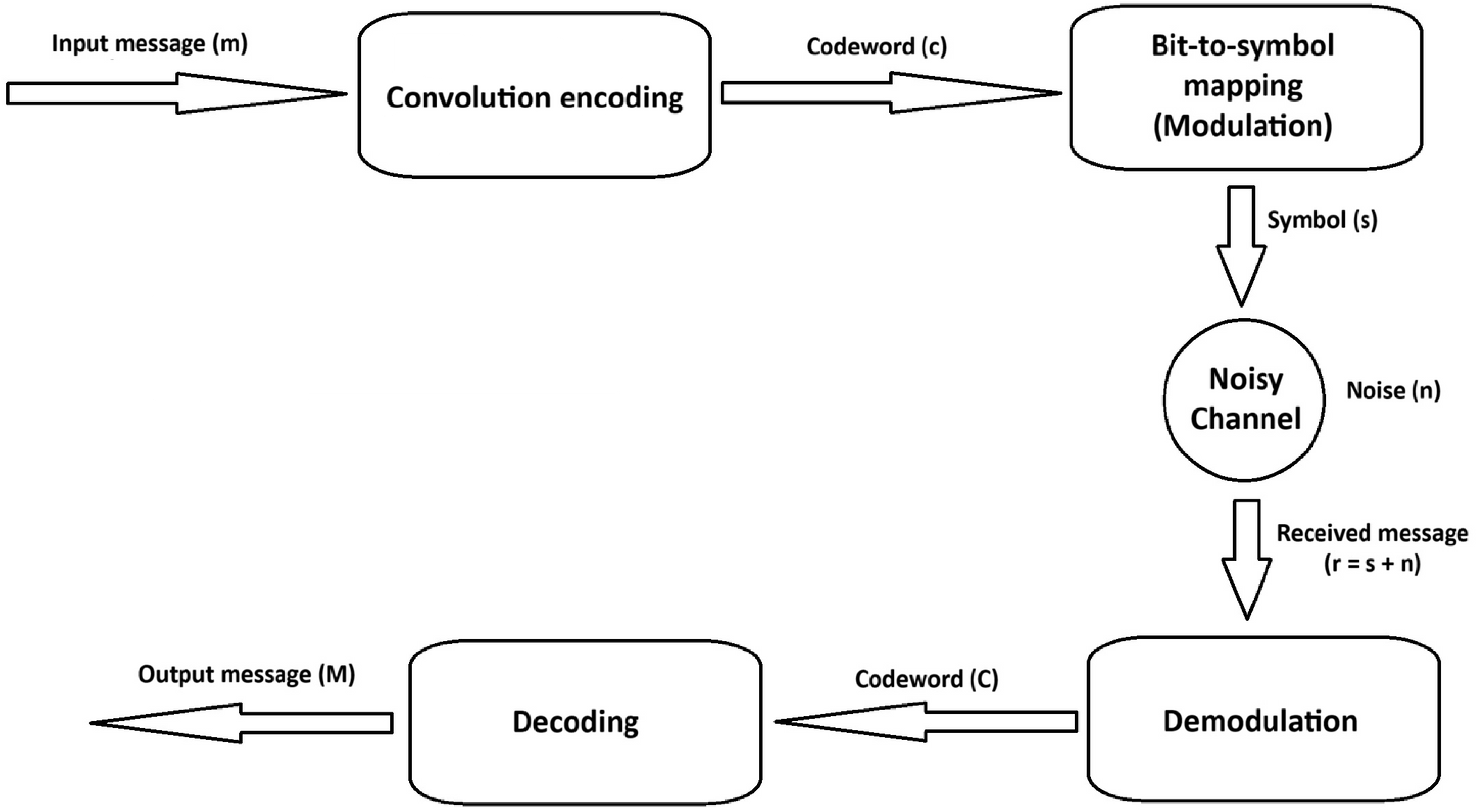
Introduction to Convolution Code

- Convolutional codes were introduced in 1955 by Peter Elias.
- In convolution code, we use the memory elements called registers and these memory indicates how the previous input bits affects the generation of the output bits.
- In convolutional coding, k successive information bits are encoded continuously without breaking their sequence. They generate n bits encoding output using generator matrix(G).
- Thus, Convolution codes are described by coding rate(R=k/n) and constraint length(Kc). Where Kc is largest number of consecutive input bits on which any particular output depends.

Why Convolution Code?

- More efficient than earlier error correction codes
- Efficient for encoding long data streams.
- Suitable for continuous data transmission

Block Diagram

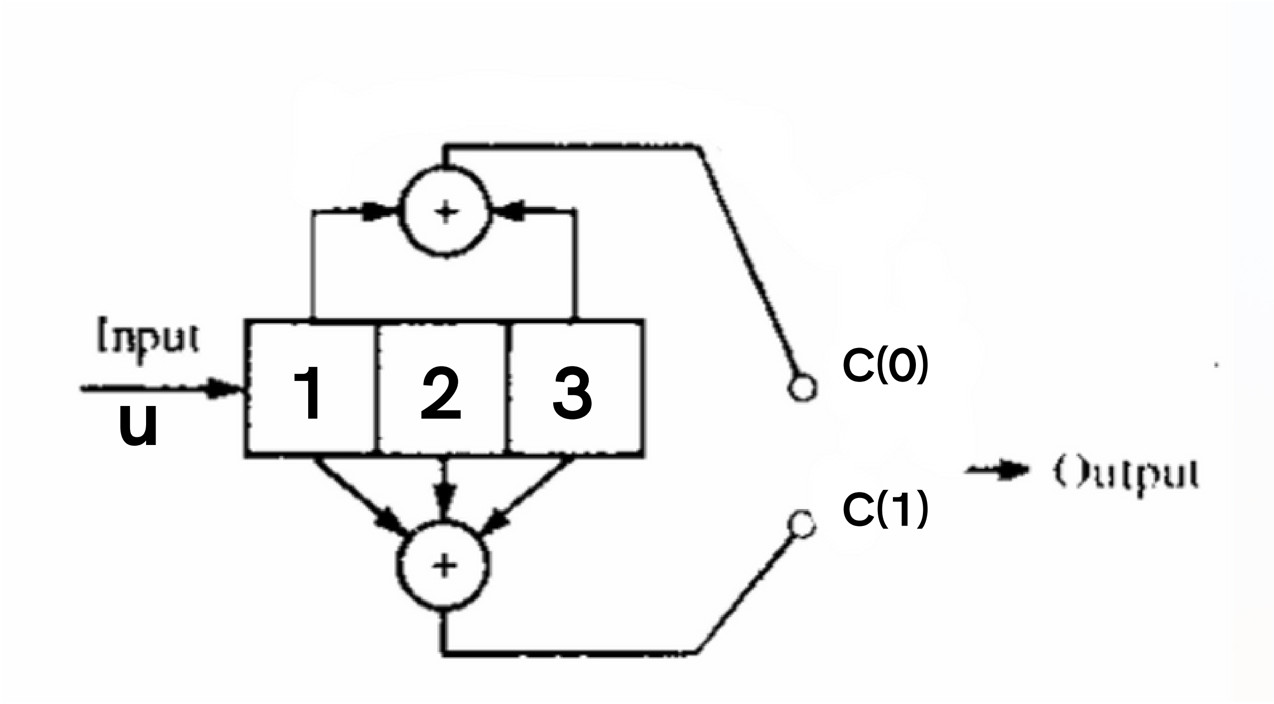


Encoding

Encoding in convolution coding scheme

For R=1/2, k = 1, n = 2, Kc = 3

- **Encoding visual representation :**



R = k/n: denotes encoding rate

k: denotes numbers of input bits

n: denotes numbers of output bits

Let g(0)=[101] and g[1]=[111]

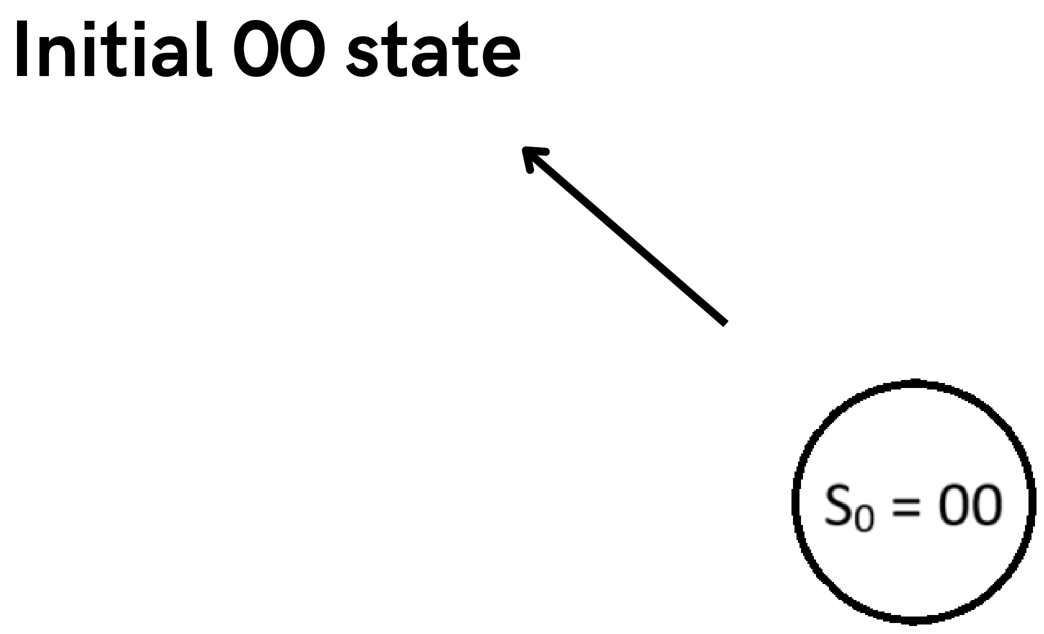
→ Here 1st block is for new coming input bit

→ Here 2nd and 3rd block is the part of shift register

→ As input is coming from left side this whole block of size Kc will shift in the right direction

Generating state Diagram

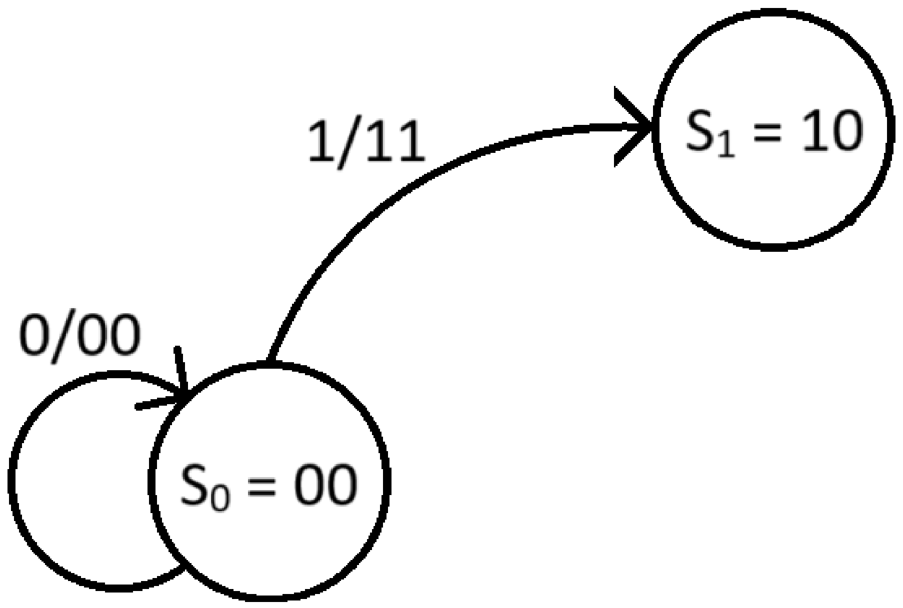
For R=1/2, k = 1, n = 2, Kc = 3



Initial state is 00. It will receive input 0 or 1, and will generate output and going to the next state.

Generating state Diagram

For R=1/2, k = 1, n = 2, Kc = 3

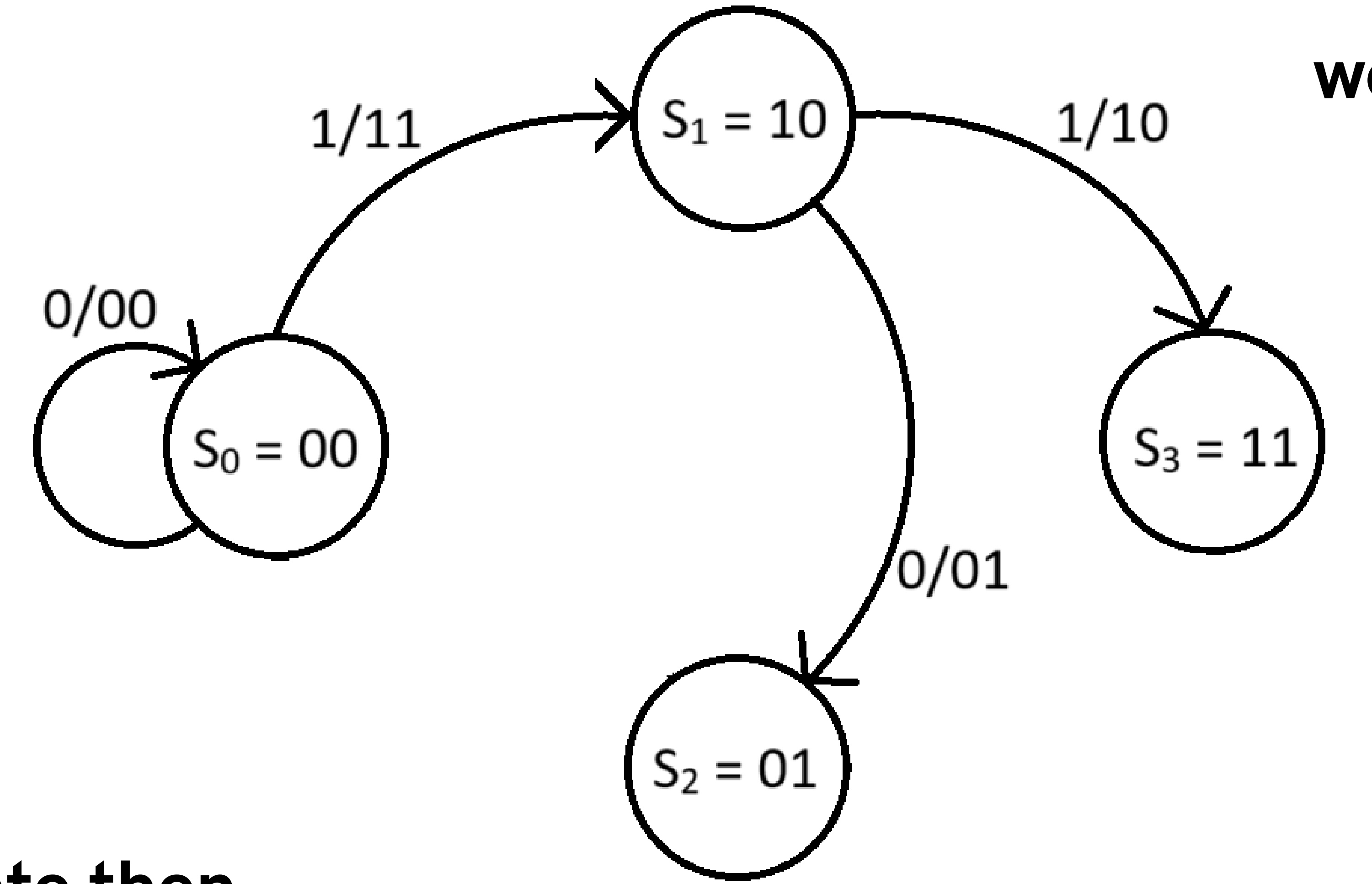


If we give input 1 then we will get output 11 and we will move to 10 state

If we give input 0 then we will get output 00 and we will remain on same state

Generating state Diagram

For R=1/2, k = 1, n = 2, Kc = 3



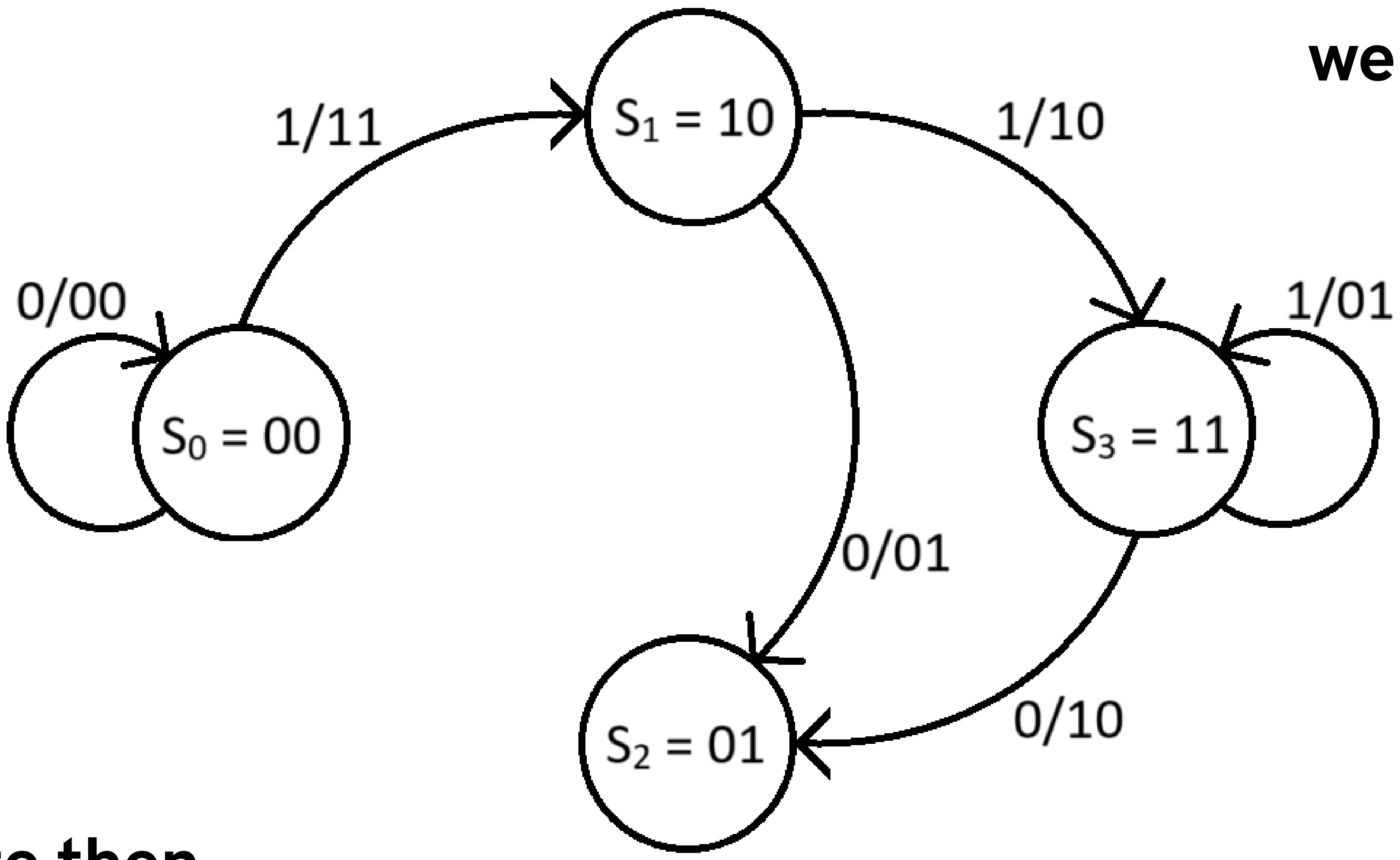
If we give input 0 to 10 state then we will get output 01 and we will move to state 01

If we give input 1 to 10 state then we will get output 10 and we will move to state 11

Prof. Yash Vasavada

Generating state Diagram

For R=1/2, k = 1, n = 2, Kc = 3



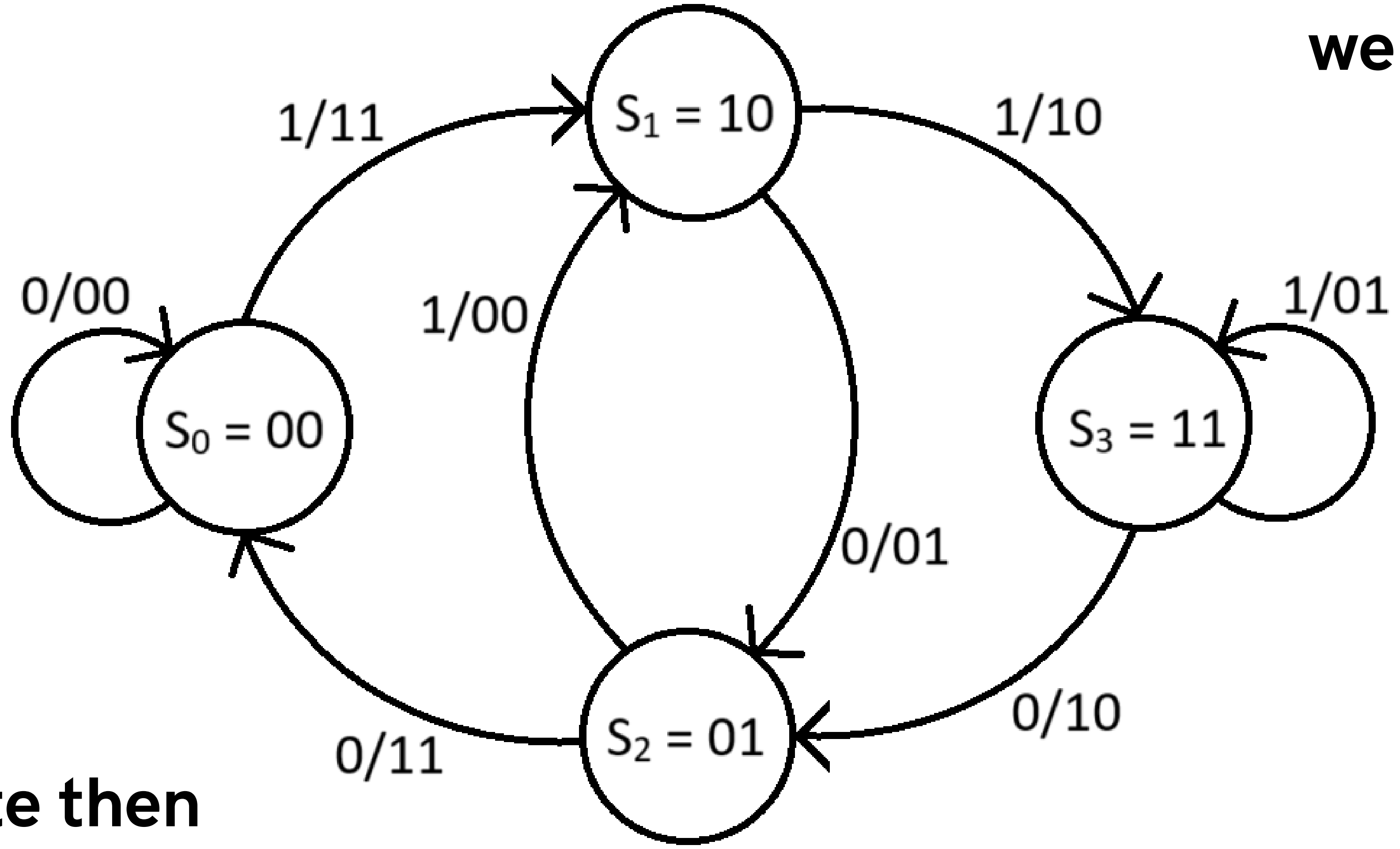
If we give input 0 to 11 state then we will get output 10 and we will move to state 01

If we give input 1 to 11 state then we will get output 01 and we will remain on same state

Generating state Diagram

For R=1/2, k = 1, n = 2, Kc = 3

If we give input 1 to 01 state then we will get output 00 and we will move to state 10



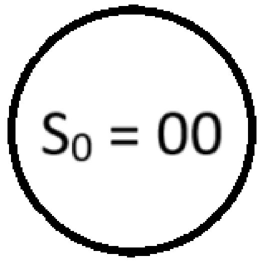
If we give input 0 to 01 state then we will get output 11 and we will move back to our initial state 00.

Encoding Example

For R=1/2, k = 1, n = 2, Kc = 3

- Let input message m = 1001
- Adding Kc-1 zeroes at end of the message to reach all zero state
- After feeding zeros m = 100100

Here we are going to encode message 1001 using state diagram



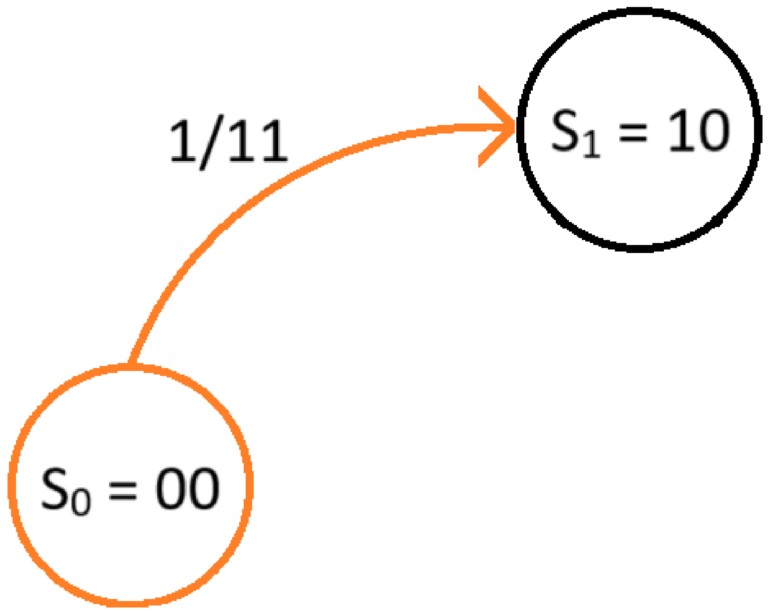
- To get output we use generator metrix and for next state we need to see in register sequence which we already done in previous state diagram.

Encoding in convolution coding scheme

For R=1/2, k = 1, n = 2, Kc = 3

m = 1 0 0 1 0 0

As we can see here. we give first bit 1 of message as input to state 00



Encoded message : 11

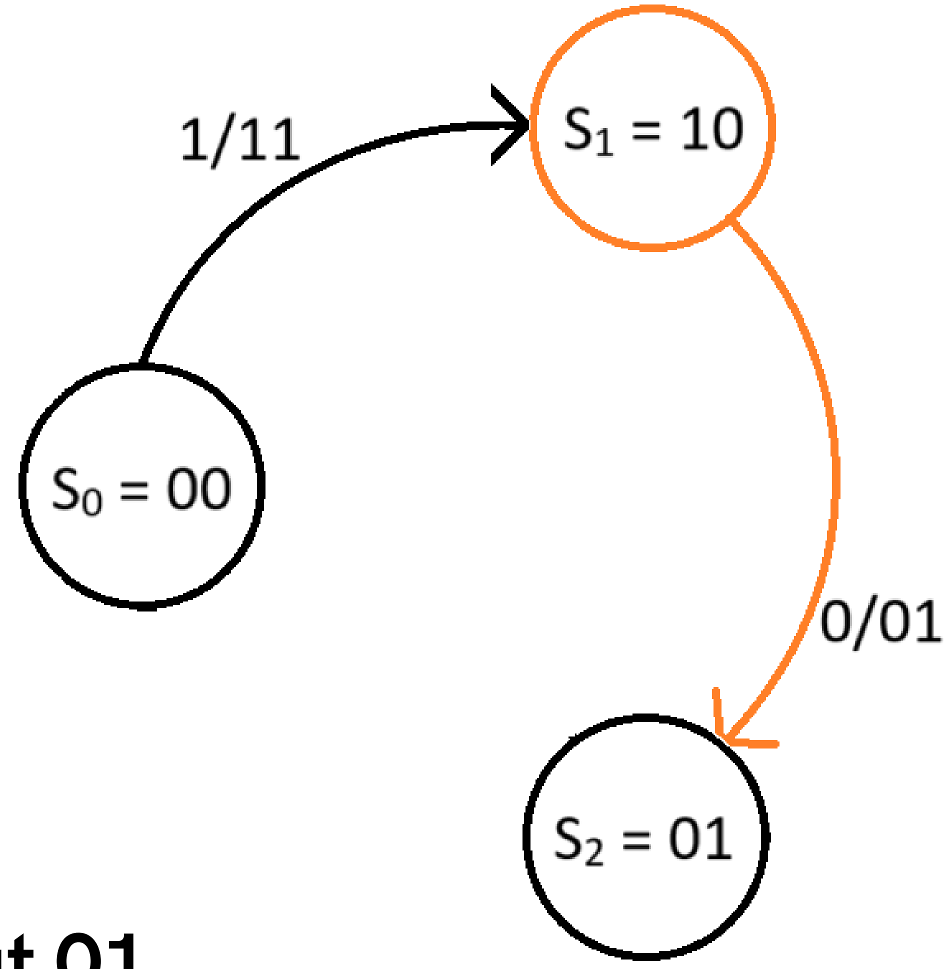
While doing so, we got output 11 and move to next state 10

Encoding in convolution coding scheme

For R=1/2, k = 1, n = 2, Kc = 3

m = 1 0 0 1 0 0

Now, we will give second bit 0 of message as input to state 10



Encoded message : 11 01

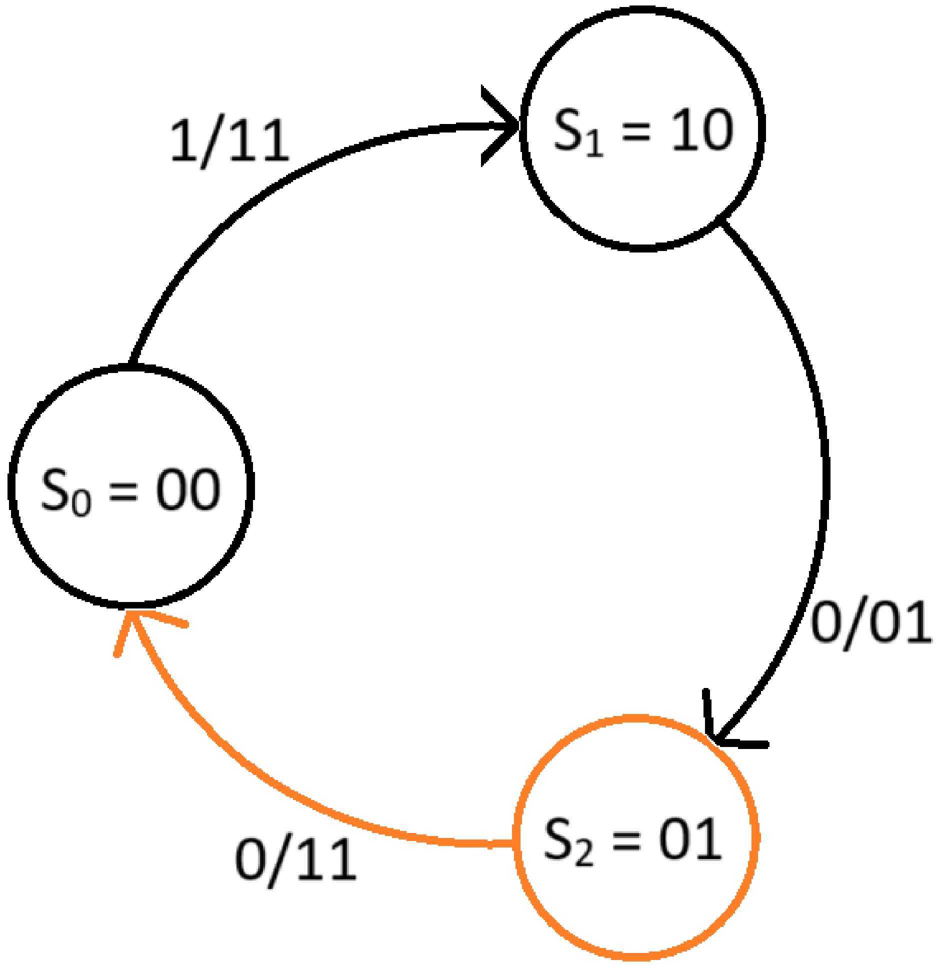
While doing so, we got output 01 and move to next state 01

Encoding in convolution coding scheme

For R=1/2, k = 1, n = 2, Kc = 3

m = 1 0 0 1 0 0

Now, we will give third bit 0 of message as input to state 01



Encoded message : 11 01 11

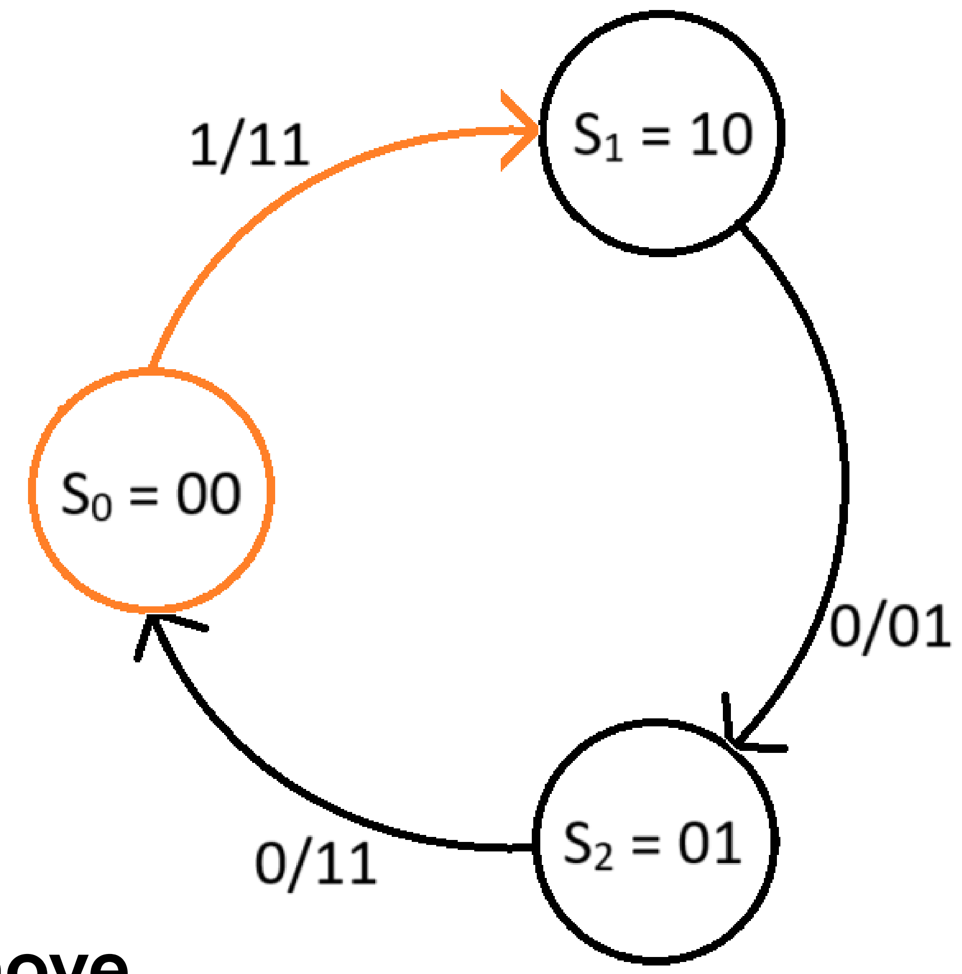
So, we got output 11 and move to state 00

Encoding in convolution coding scheme

For R=1/2, k = 1, n = 2, Kc = 3

m = 1 0 0 1 0 0

Now, we will give forth bit 1 of message as input to state 00



Encoded message : 11 01 11 11

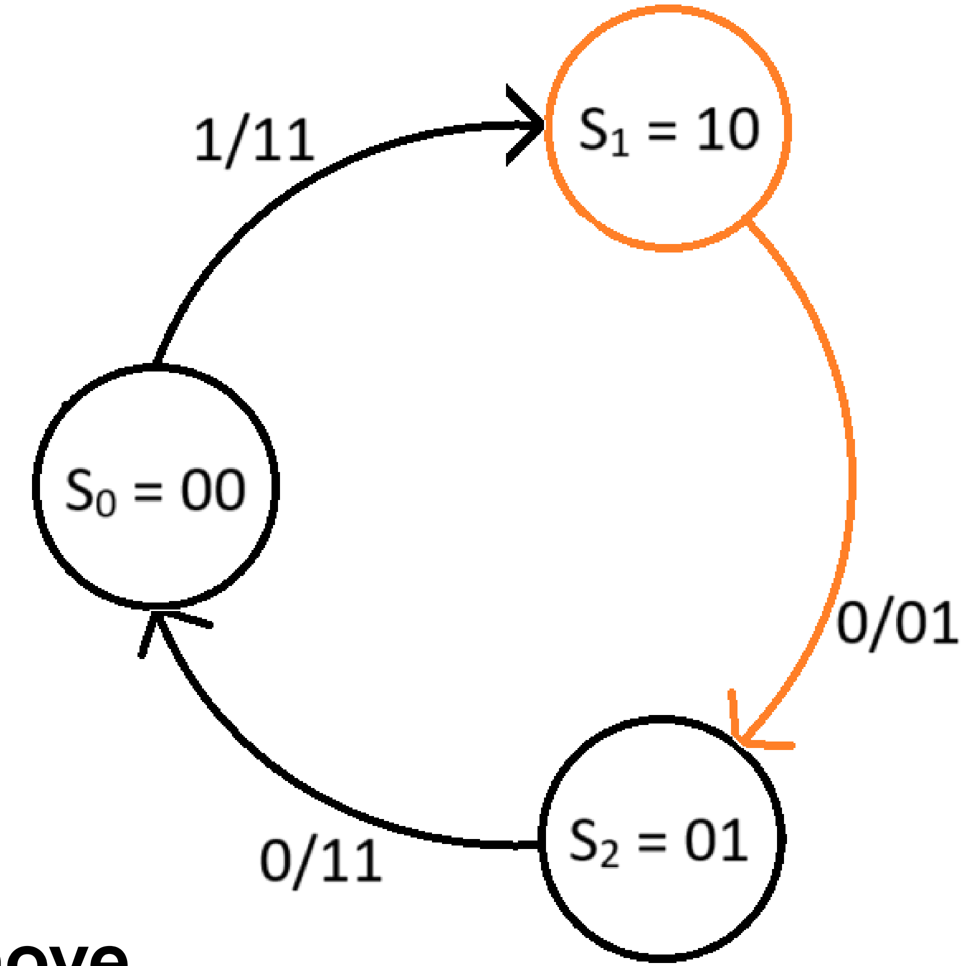
So, we got output 11 and move to next state 10.

Encoding in convolution coding scheme

For R=1/2, k = 1, n = 2, Kc = 3

m = 1 0 0 1 0 0

Now, we will give fifth bit 0 of message as input to state 10



Encoded message : 11 01 11 11 01

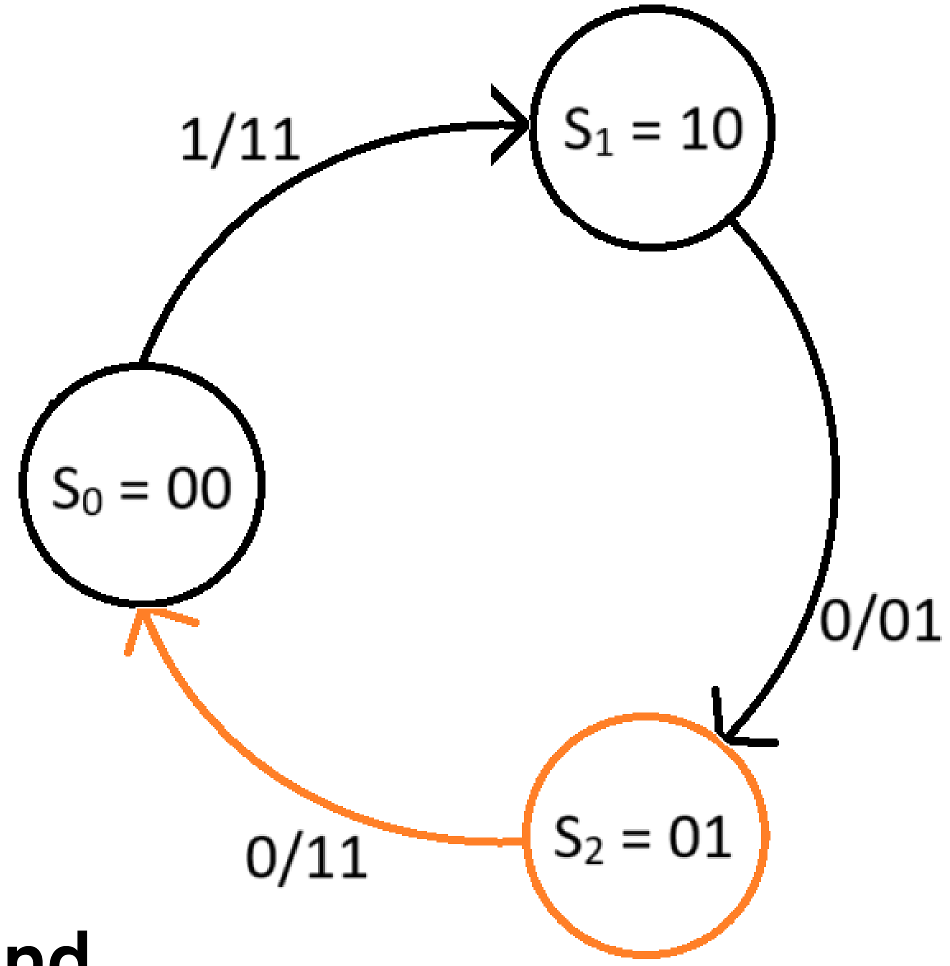
So, we got output 01 and move to next state 01.

Encoding in convolution coding scheme

For R=1/2, k = 1, n = 2, Kc = 3

m = 1 0 0 1 0 0

Now, we will give forth bit 0 of message as input to state 01



Encoded message : 11 01 11 11 01 11

So, we got final output 11 and move to initial state 00.

Modulation

BPSK & AWGN

What is Modulation ?

- A modulator is a bit to symbol mapper which maps set of bits to complex valued symbols and the process of modulating the bits to symbols is called Modulation.

BPSK : Binary Phase Shift Keying (BPSK) is a digital modulation technique for transmitting binary data (0s and 1s) by changing the phase of a signal.

- ‘1' is represented by a 180-degree phase shift
- ‘0' is represented by no phase shift

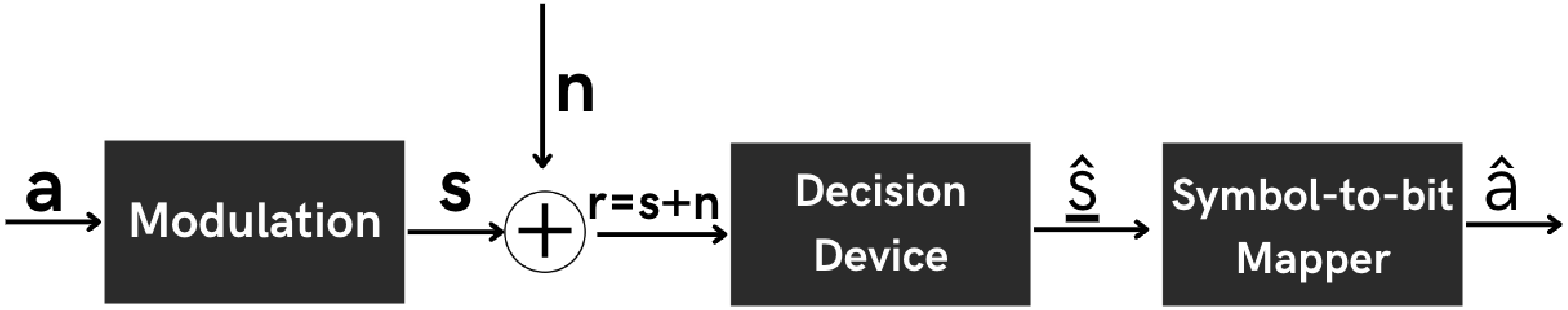
AWGN : properties of Additive White Gaussian Noise

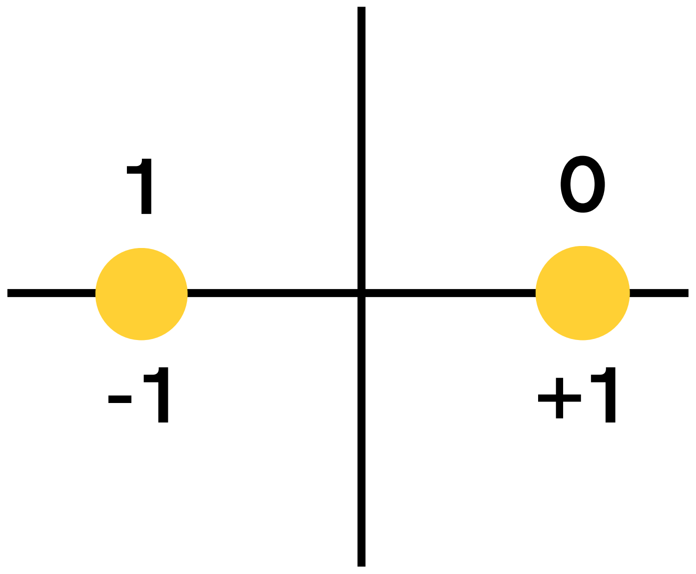
- Additive: noise that added to the signal during transmission.
- White: noise has equal power across all frequencies.
- Gaussian: The probability distribution function (PDF) of the AWGN noise follows a normal distribution, also known as a Gaussian distribution.

How BPSK works?

In this method of modulation:

- **The bit '0' is mapped to symbol '1'**
- **The bit '1' is mapped to symbol '-1'**



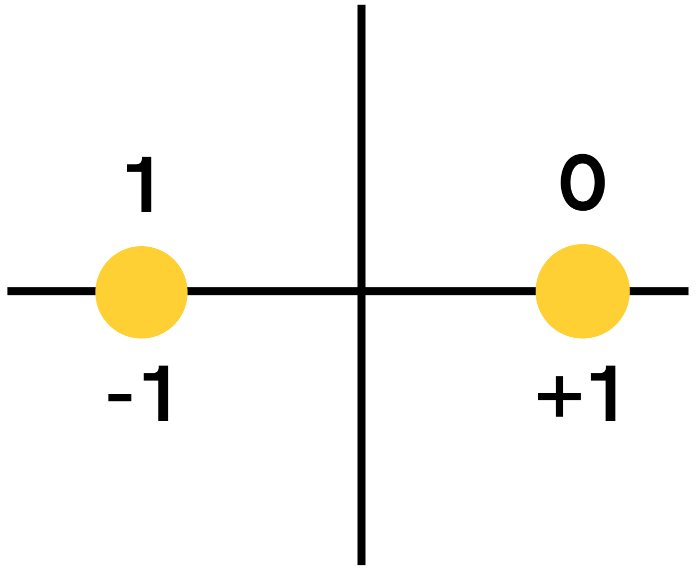


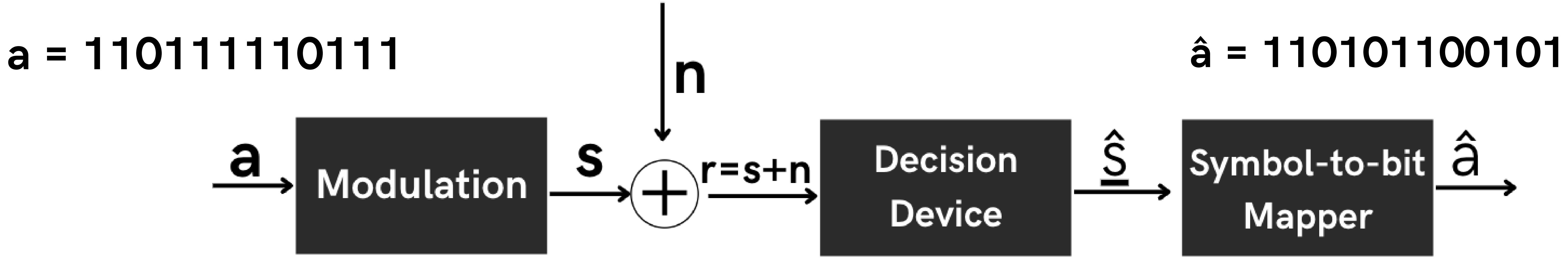
The mapping is done from the equation : s = 1-2a

$$s = \begin{cases} 1 & \text{, bit } a=0 \\ -1 & \text{, bit } a=1 \end{cases}$$

How BPSK works?

- Passing the encoded message(a) to the modulator. we get the received message(â) which has error, at the output of the demodulator.





AWGN Noise

- Noise = $\sigma * \mu$, $\mu \sim N(0,1)$
- Noise power = $\sigma^2 = N_0/2$
- SNR = $2*Es/N_0 = Es/\sigma^2$, where $Es=1$ which is mean of squares of symbols
- Signal energy per information bit = Eb = Es/R
- SNR = $(2*Eb*R)/N_0$
- Eb/N_0 = SNR/(2*R) = $1/(2*\sigma^2*R)$

$$\sigma = \sqrt{\frac{1}{2*R*(Eb/N_0)}}$$

Viterbi

Decoding

Why Viterbi Decoding?

- Inefficient performance of previous other decoding algorithms, reason is that other algorithms used brute force approaches having very high time complexity due to tracking unnecessary nodes.
- Viterbi algorithm was introduced in 1967 by Andrew Viterbi .
- Viterbi decoding uses dynamic programming approach that gives optimal paths and reduce computation complexity too.

Viterbi Decoding of convolution code

Two types of Decoding :

- **Hard Decision Viterbi Decoding (HDD)**

  Hard Decision Decoding works on Hamming Distance

- **Soft Decison Viterbi Decoding (SDD)**

  Soft Decision Decoding works on Euclidian Distance

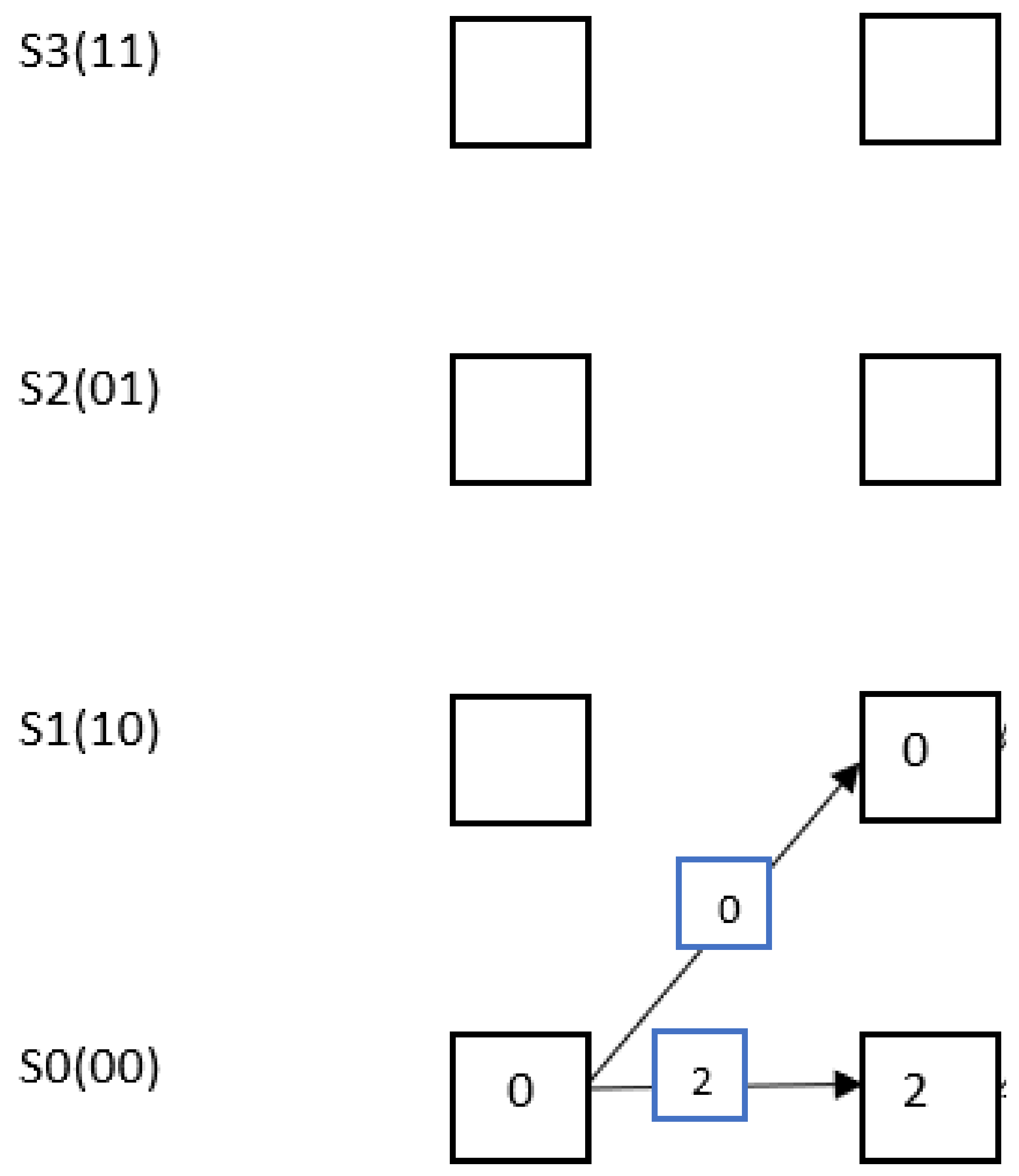
Hard Decision Viterbi Decoding (HDD)

<u>Steps to decode the codeword with Viterbi Algorithm:</u>

- Make Trellis Diagram
- Make a forward traversal in Trellis diagram through all nodes
- To calculate the Hamming distance, you need to XOR the received message's n-size block with the corresponding bits of the output state, then sum the resulting values.
- Each Node contains a partial path metric
- To every node more than one branch may be reaching, so for each branch a branch metric is calculated and added and compared, and then the minimum branch metric is selected.

Trellis Diagram And Hamming Distance

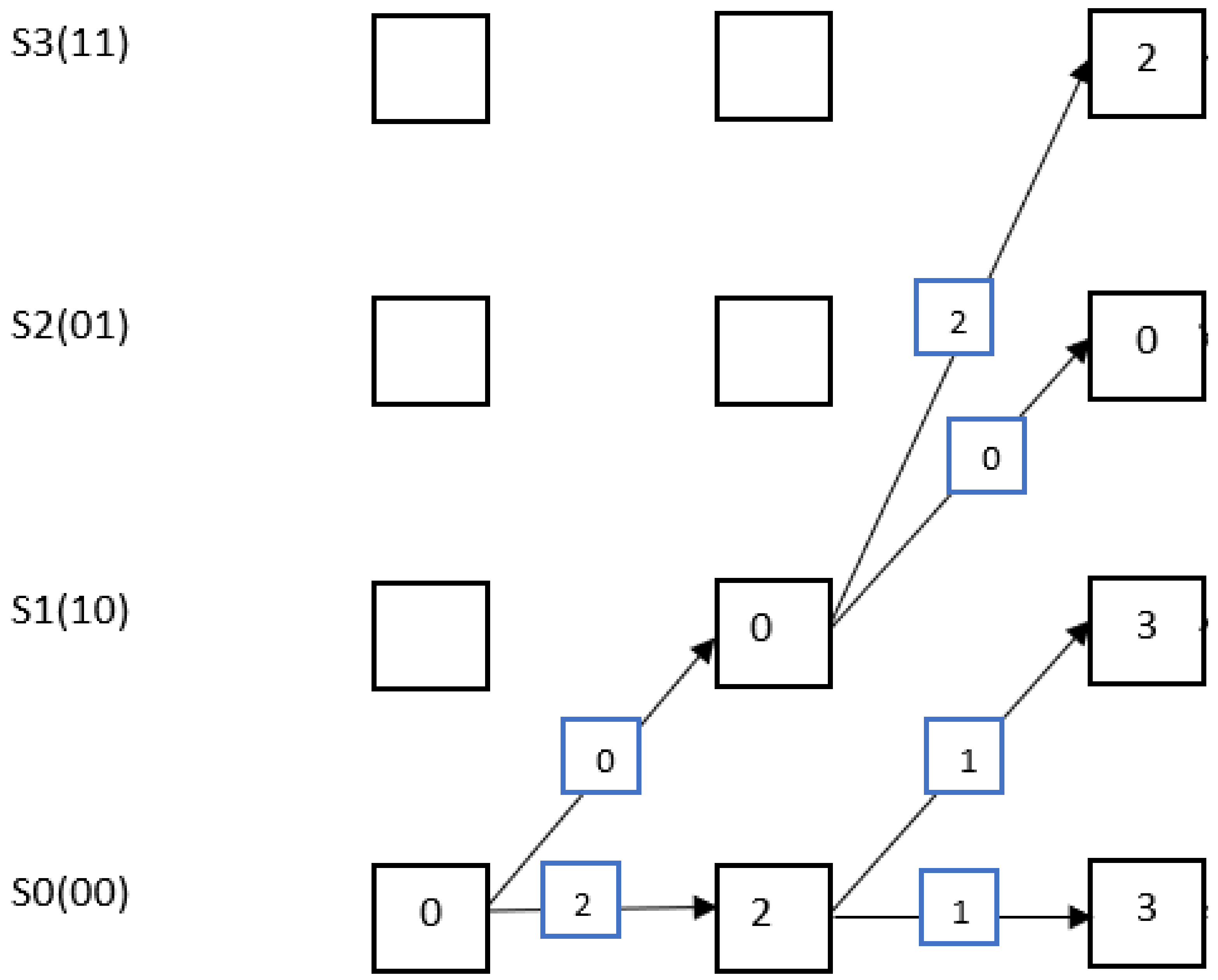
For, received message(â)=110101100101



Received massage 11

Trellis Diagram And Hamming Distance

For, received message(â)=110101100101

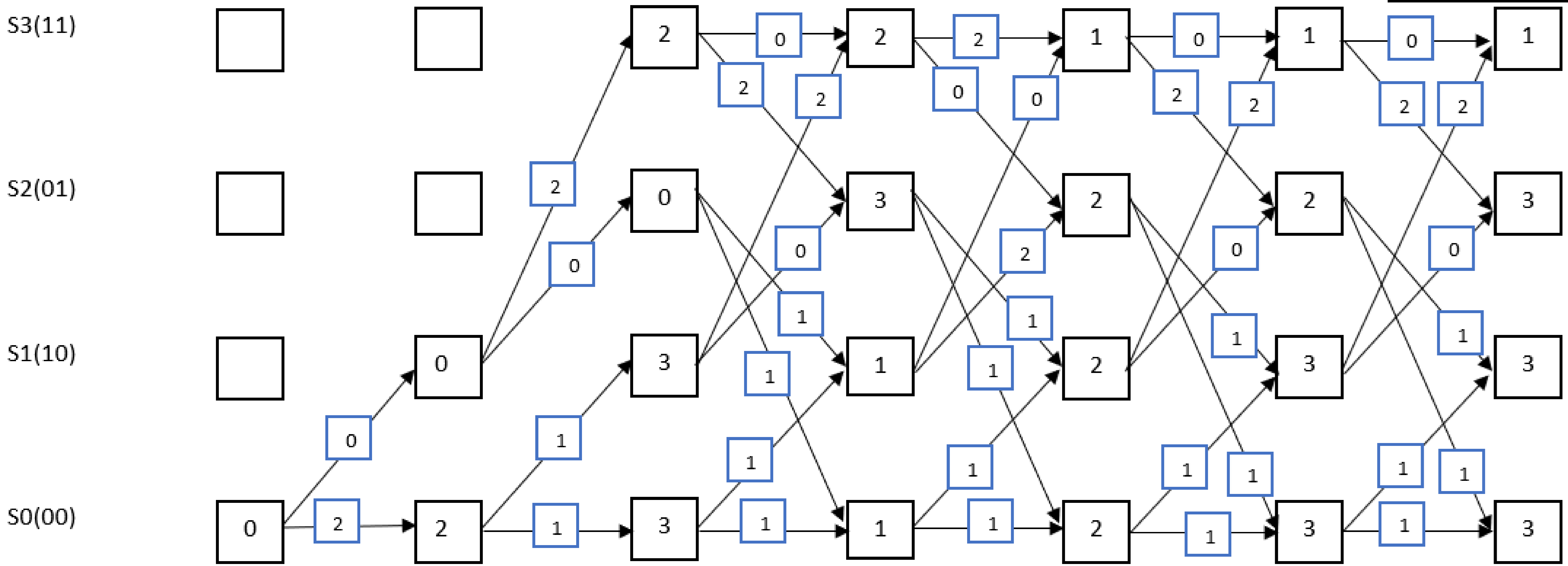


Received massage 11 01

Trellis Diagram And Hamming Distance

For, received message(â)=110101100101

[redacted]



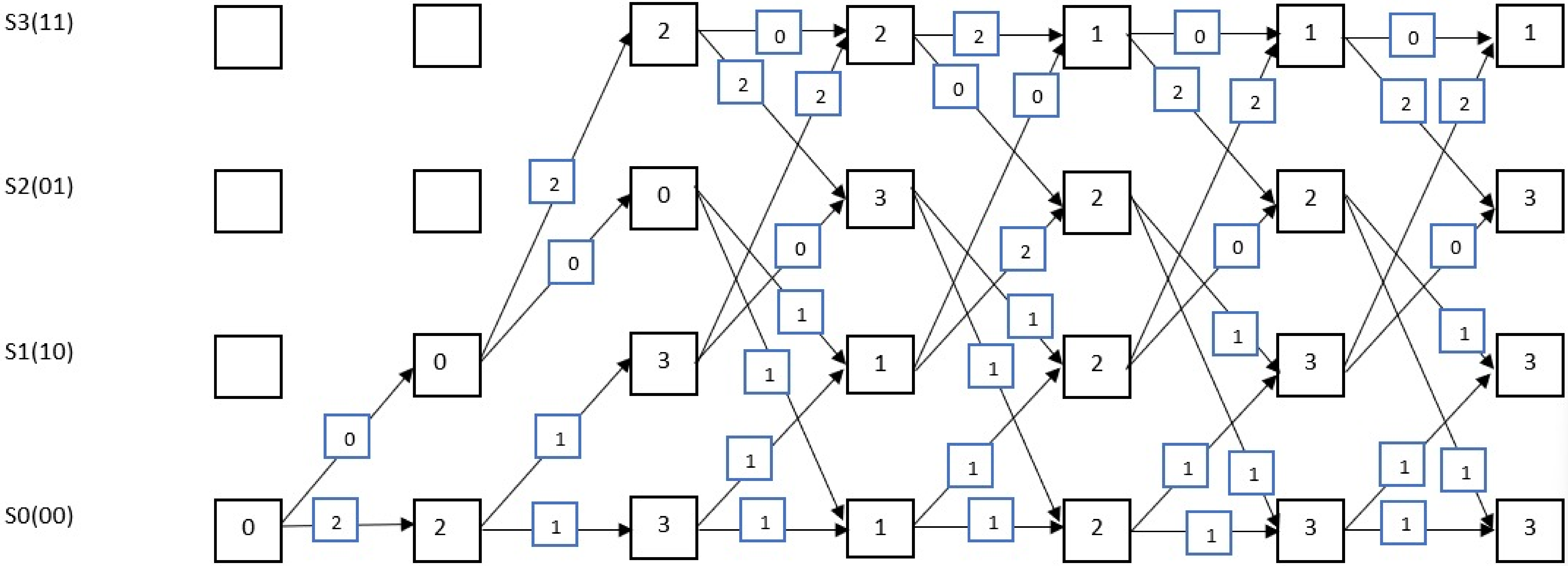
Received massage 11 01 01 10 01 01

Traceback For Viterbi Decoding

- As we reach the end of the Trellis diagram we trace back to the path having the minimum path metric/hamming distance and this traceback determines the value of data bits, which is our decoded output.

Traceback For Viterbi Decoding

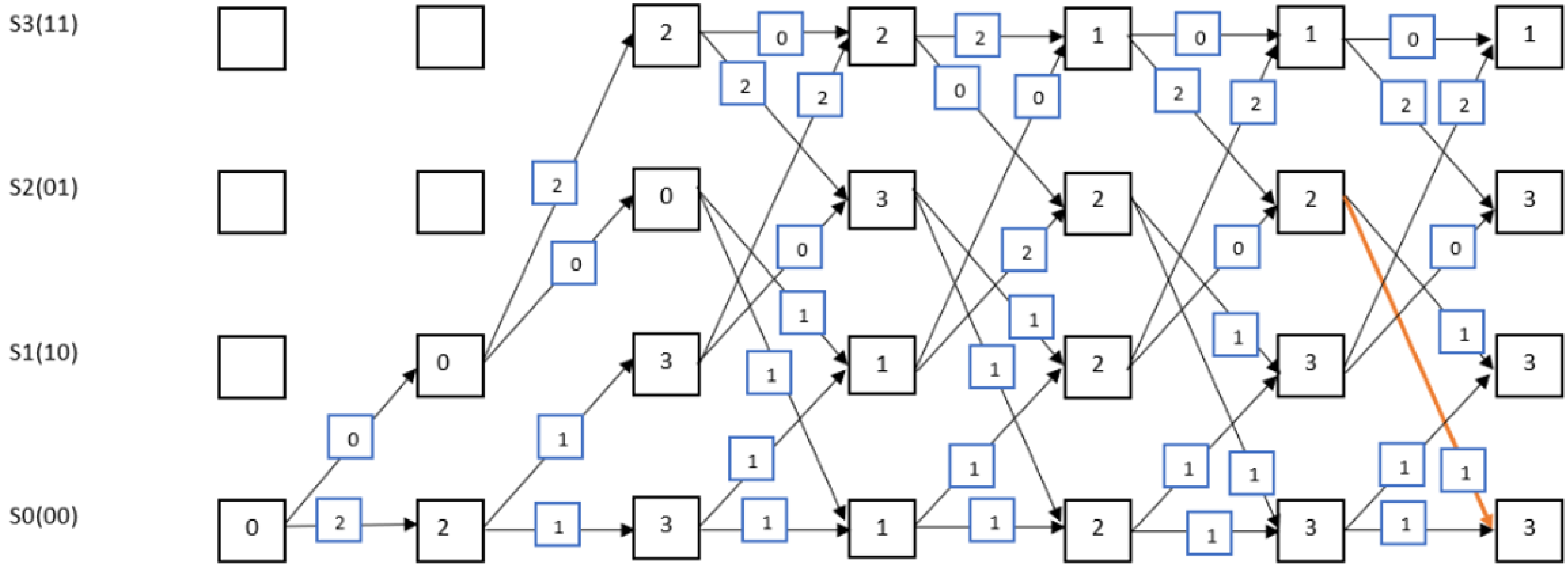
For, received message(â)=110101100101



Received massage 11 01 01 10 01 01

Traceback For Viterbi Decoding

For, received message(â)=110101100101

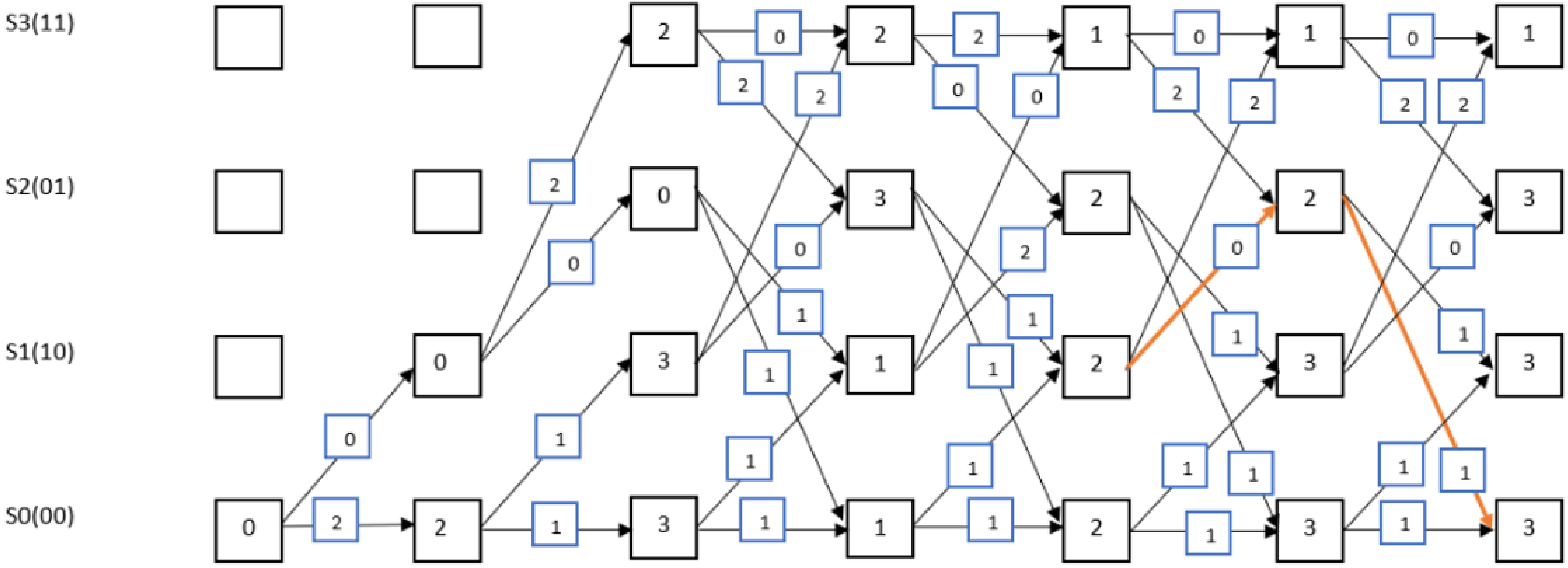


Decoded massage 0

Received massage 11 01 01 10 01 01

Traceback For Viterbi Decoding

For, received message(â)=110101100101



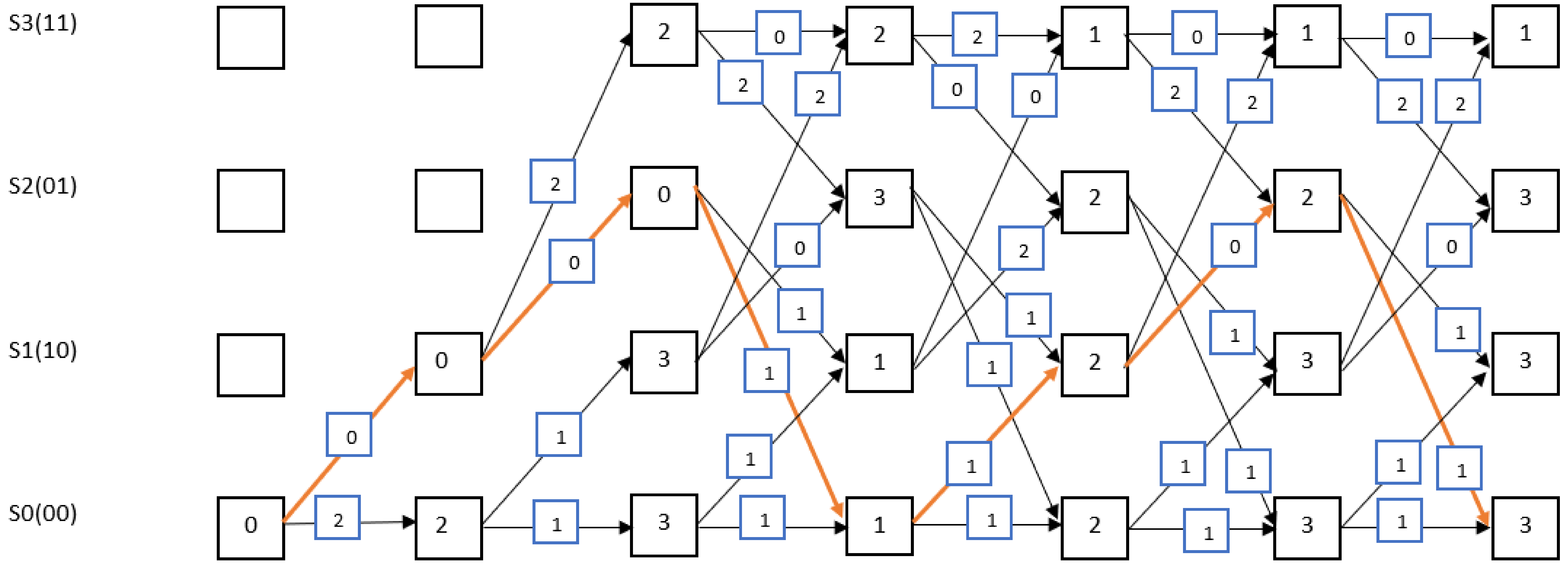
Decoded massage 0 0

Received massage 11 01 01 10 01 01

Traceback For Viterbi Decoding

[redacted]

For, received message(â)=110101100101



Decoded massage	1	0	0	1	0	0
Received massage	11	01	01	10	01	01

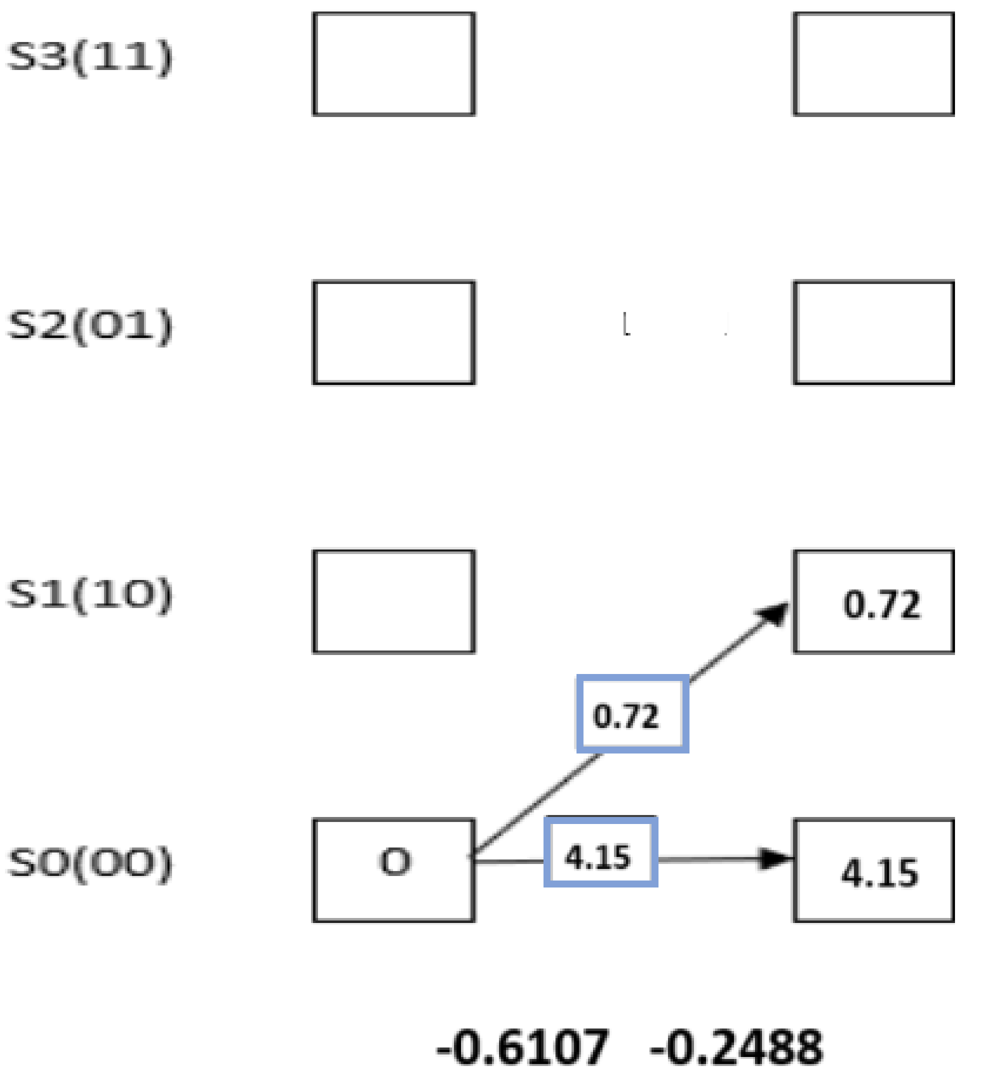
Soft Decision Viterbi Decoding (SDD)

Steps to decode the codeword by viterbi algorithm (SDD) :

- Make trellis diagram
- put block of size n from received message and find Euclidean distance with outputs of the states
- To find Euclidean distance, we take sum of square of the difference of bits of received output bits to the modulated output of corresponding state output.
- To every node more than one branch may be reaching, so for each branch a branch metric is calculated and added and compared, and then the minimum branch metric is selected.

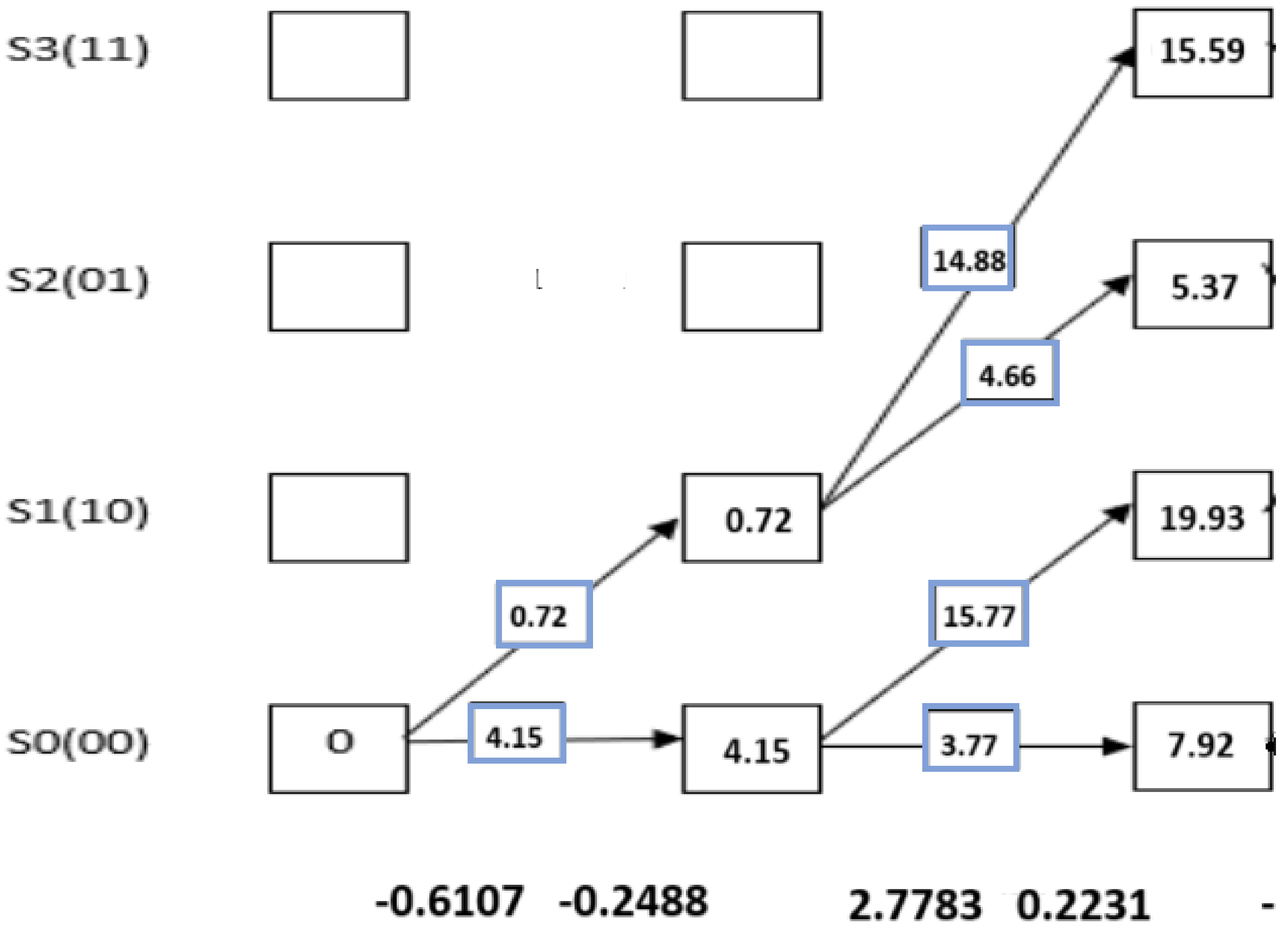
Trellis Diagram And Euclidean Distance

For, received message(â)=110101100101



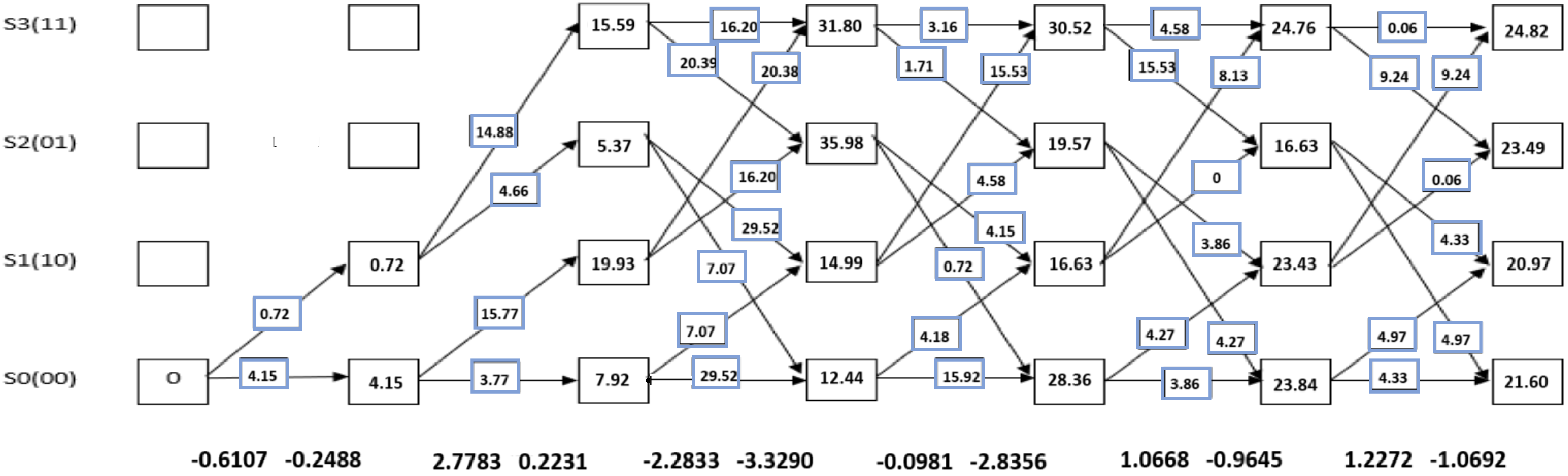
Trellis Diagram And Euclidean Distance

For, received message(â)=110101100101



Trellis Diagram And Euclidean Distance

For, received message(â)=110101100101



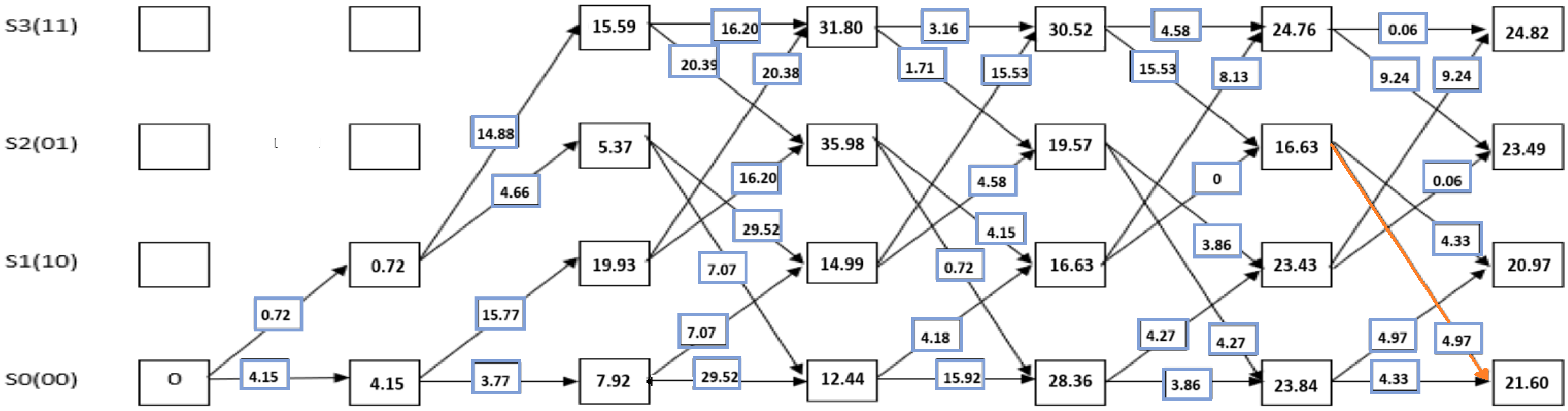
Traceback For Viterbi Decoding

For, received message(â)=110101100101

- **As we reach the end of the Trellis diagram we trace back to the path having the minimum path metric/Euclidean distance and this traceback determines the value of data bits, which is our decoded output.**

Traceback For Viterbi Decoding

For, received message(â)=110101100101

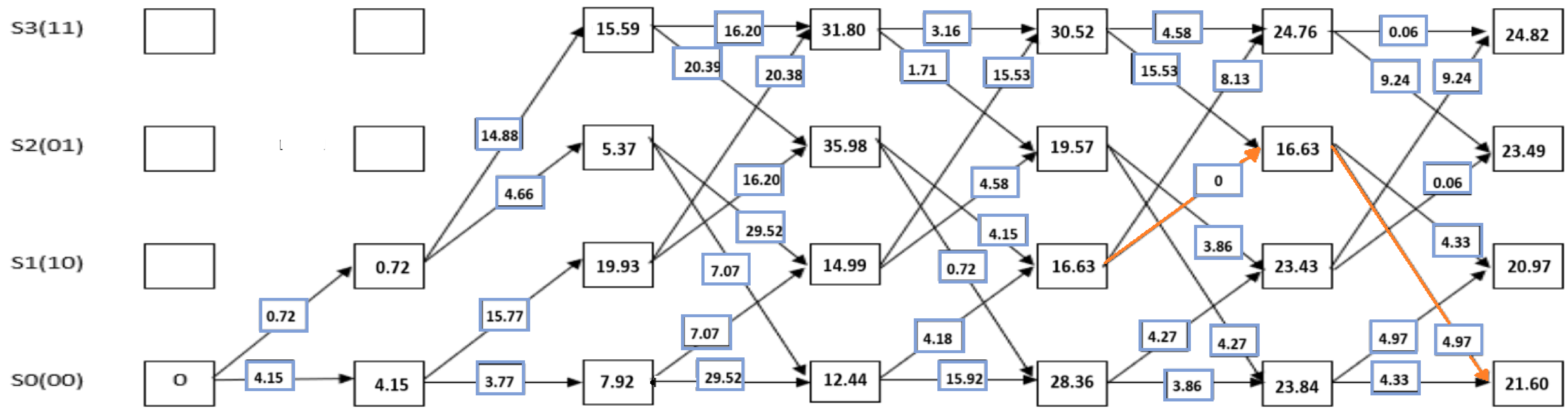


received massage : -0.6107 -0.2488 2.7783 0.2231 -2.2833 -3.3290 -0.0981 -2.8356 1.0668 -0.9645 1.2272 -1.0692

Decoded massage : 0

Traceback For Viterbi Decoding

For, received message(â)=110101100101

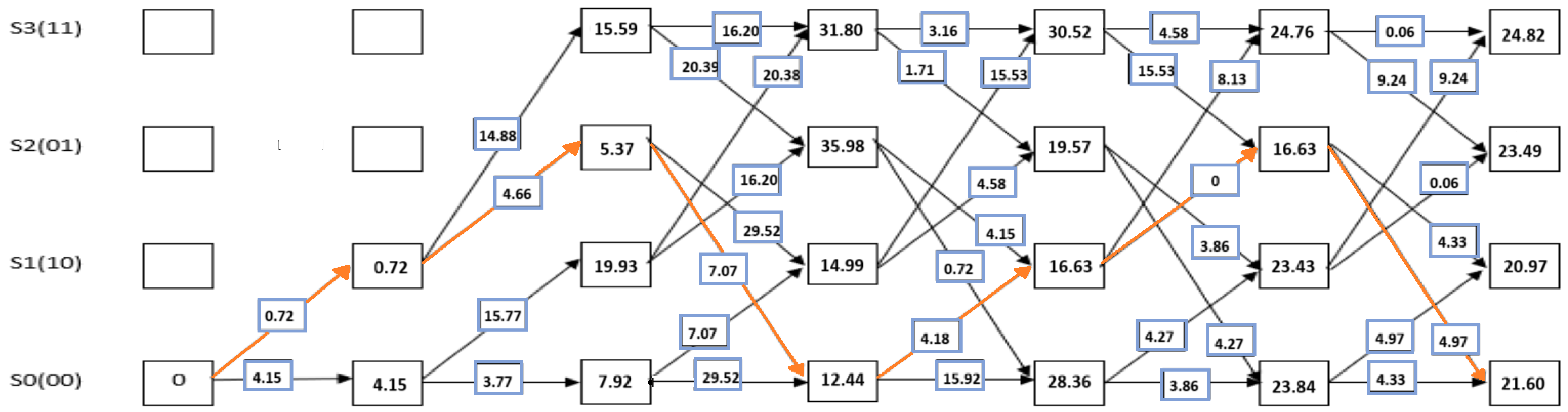


received massage : -0.6107 -0.2488 2.7783 0.2231 -2.2833 -3.3290 -0.0981 -2.8356 1.0668 -0.9645 1.2272 -1.0692

Decoded massage : 0 0

Traceback For Viterbi Decoding

For, received message(â)=110101100101



received massage : -0.6107 -0.2488 2.7783 0.2231 -2.2833 -3.3290 -0.0981 -2.8356 1.0668 -0.9645 1.2272 -1.0692

Decoded massage : 1 0 0 1 0 0

Transfer Function

- Transfer function : A mathematical formula of all the paths that start and end at all zero state.

$$T(D,N) = \sum_{d=d_{free}}^{\infty} a_d D^d N^{f(d)}$$

D Exponent to 'd' : no. of ones in the output code word.

N Exponent to 'f(d)' : no. of ones in the input block (k-bits) at a time.

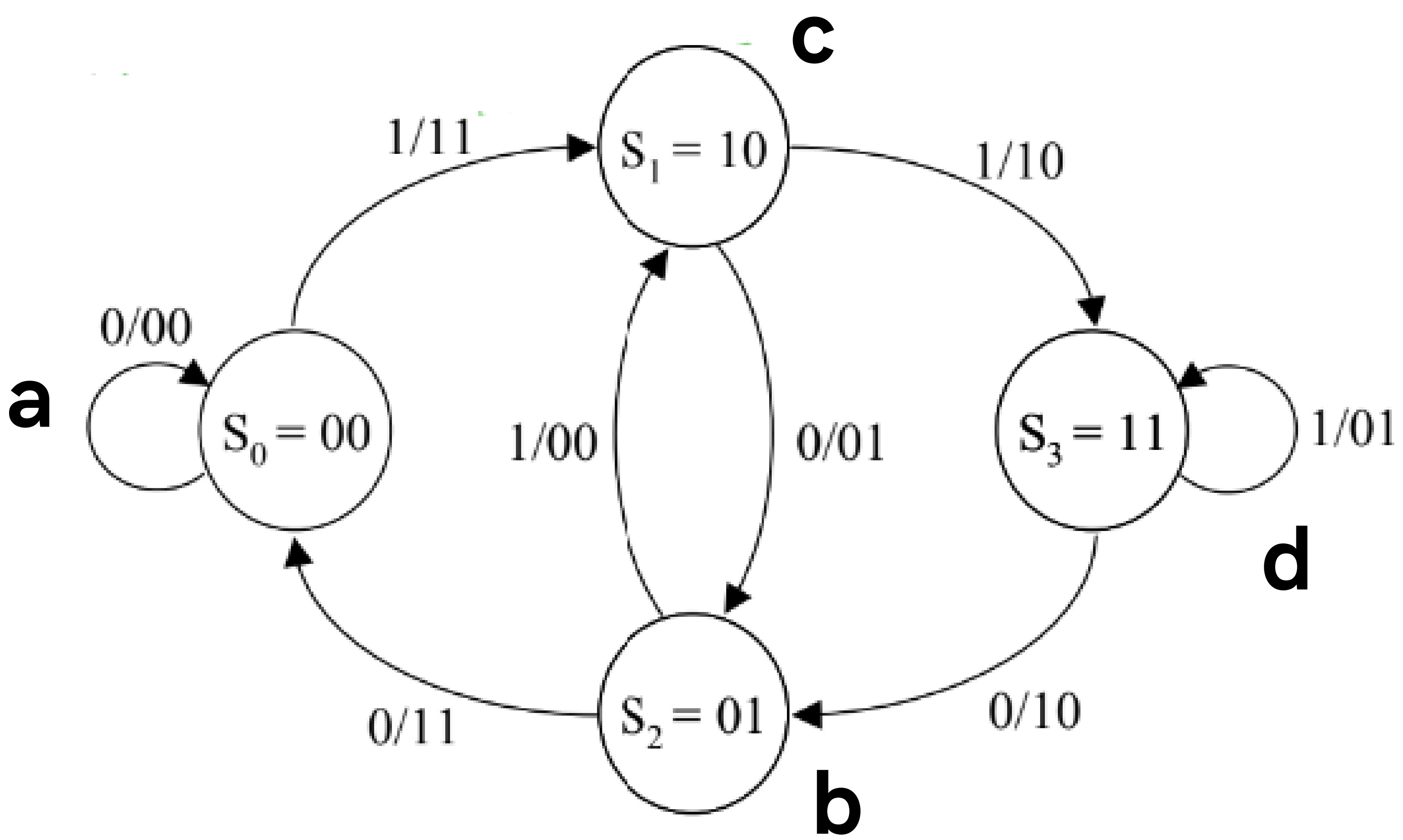
a_d : $coefficient\ of\ D^d$

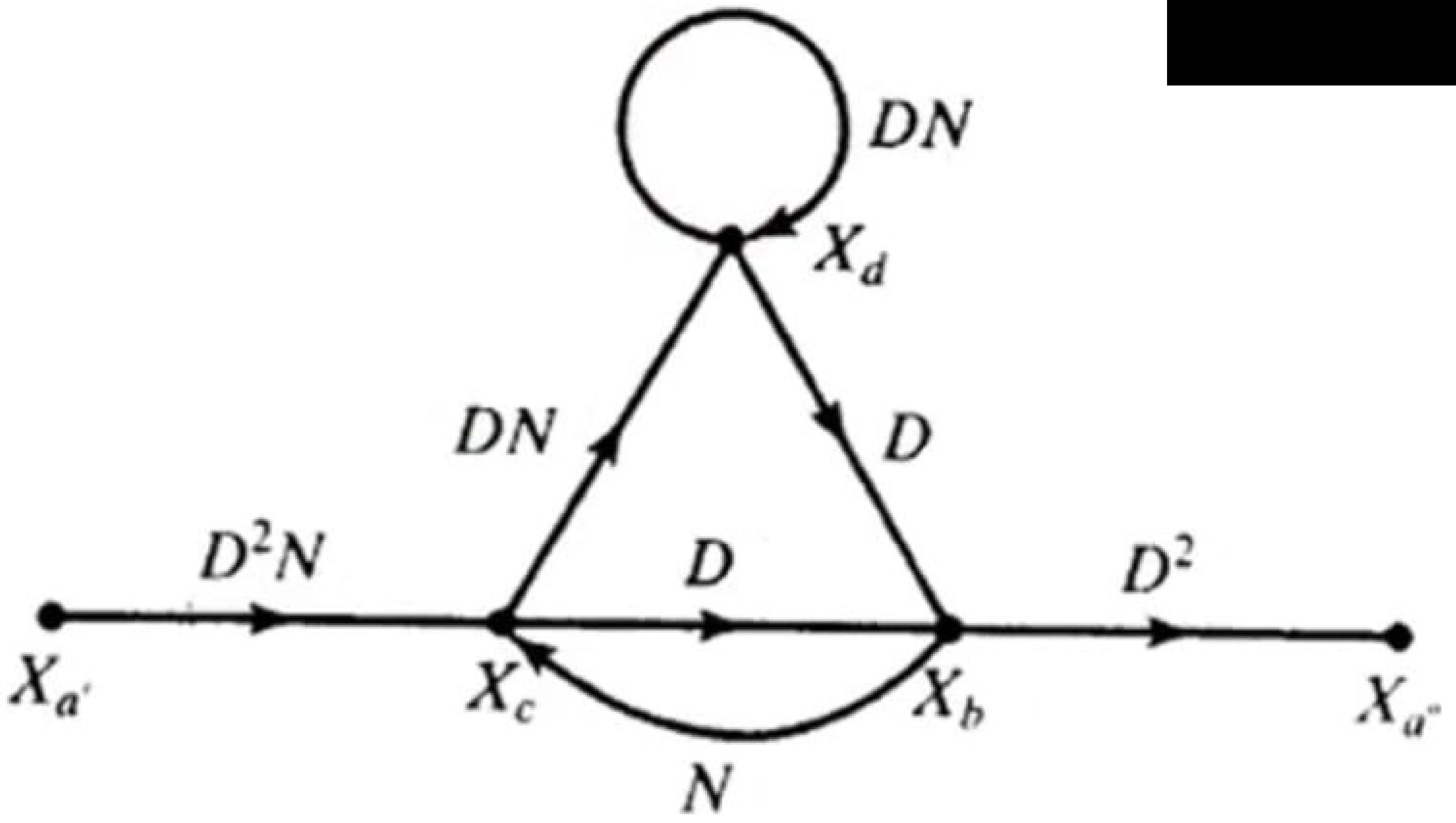
$$a_d = 2^{d-d_{free}}$$

Properties of Transfer function :

- Provide the properties of all paths
- For the first time, Paths start from all zero state, traverse a Trellis path and return to all zero state.

Transfer Function





$$X_c = D^2 N X_{a'} + N X_b$$

$$X_b = D X_d + D X_c$$

$$X_d = D N X_c + D N X_d$$

$$X_{a''} = D^2 X_b$$

$$\frac{X_{a''}}{X_{a'}} = \frac{D^2 X_b}{\frac{1}{D^2 N}(X_c - N X_b)} = \frac{D^4 N X_b}{(X_c - N X_b)}$$ (put the value of X_b and X_c)

$$= \frac{D^5 N}{(1 - 2DN)}$$

Limitation of convolution code & viterbi decoding algorithm

- Specially ,for large state space Viterbi algorithm can be slow and memory consuming process.

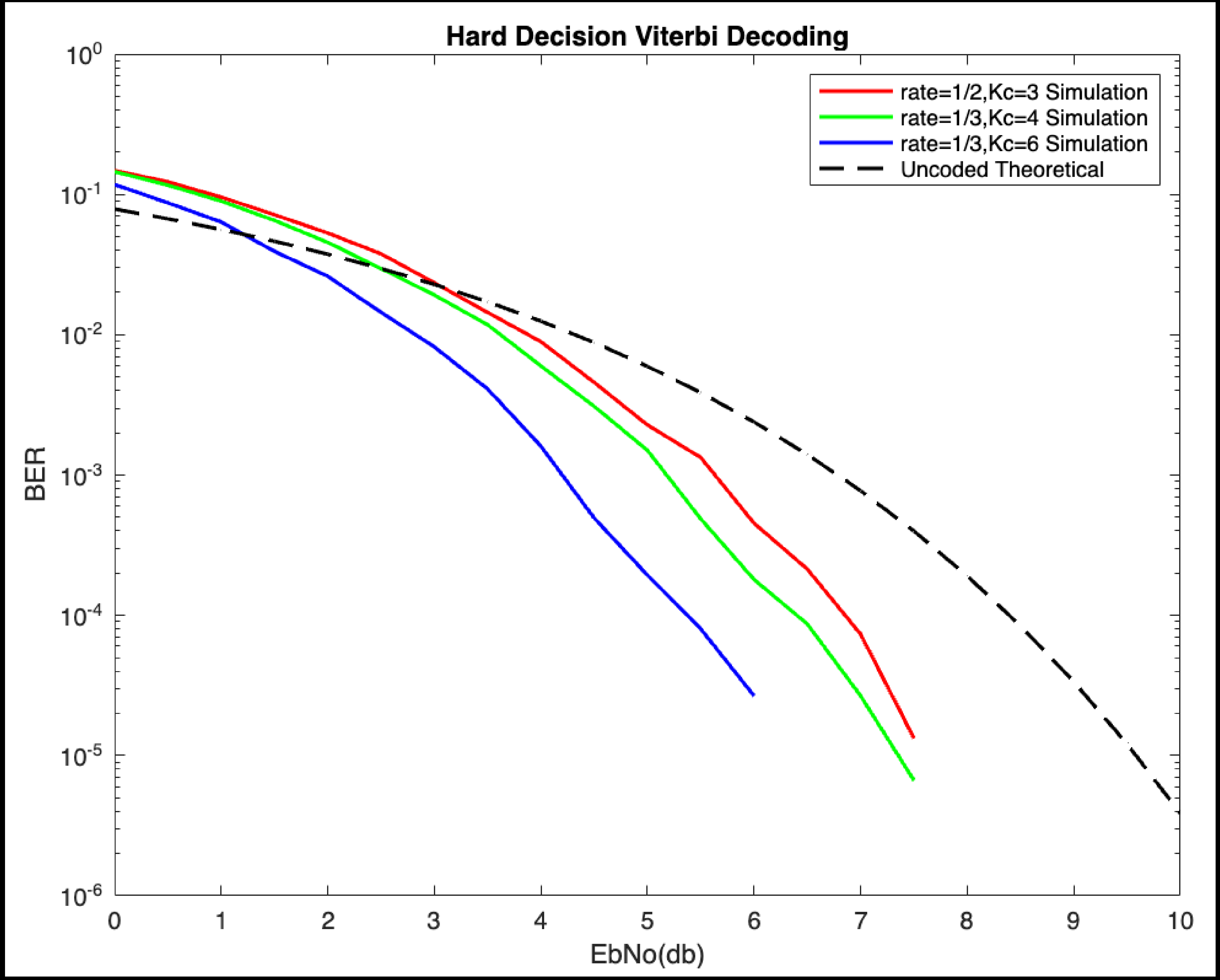
- Convolutional decoding codes can be computationally expensive with respect to block codes.

- Increase in constraint length reduces the error probability but it increases complexity of convolution code exponentially.

Analysis of Hard Decision Decoding graph

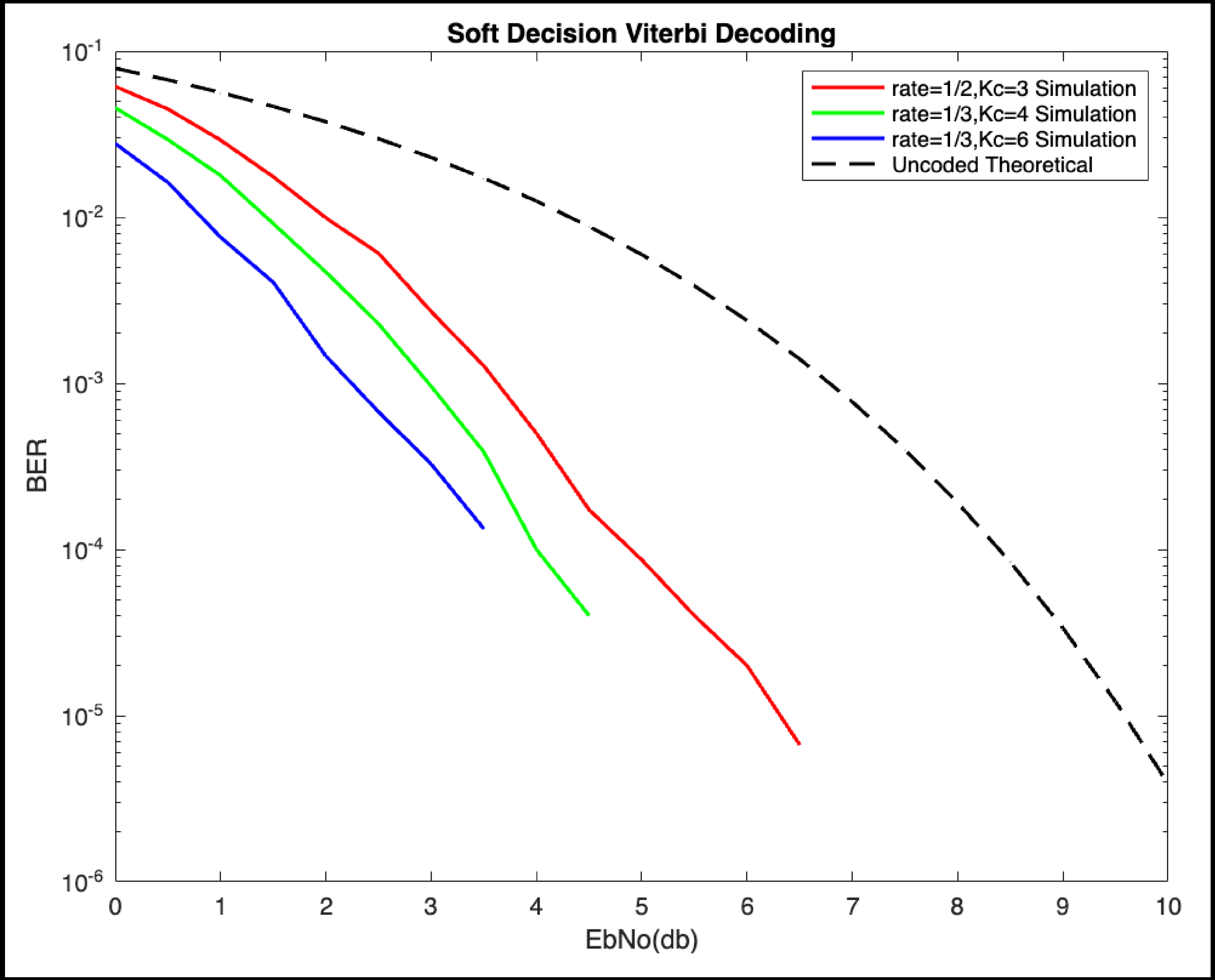
- **The graph is for code rate R=1/2 and Kc=3, R=1/3 and Kc=4, R=1/3 and Kc=6**
- **BER depends on:**
  1. **Rate of the convolution code**
  2. **Constraint length of the code**
- **Behaviour of the Graph:**

  Why graphs are crossed ?



Analysis of Soft Decision Decoding graph

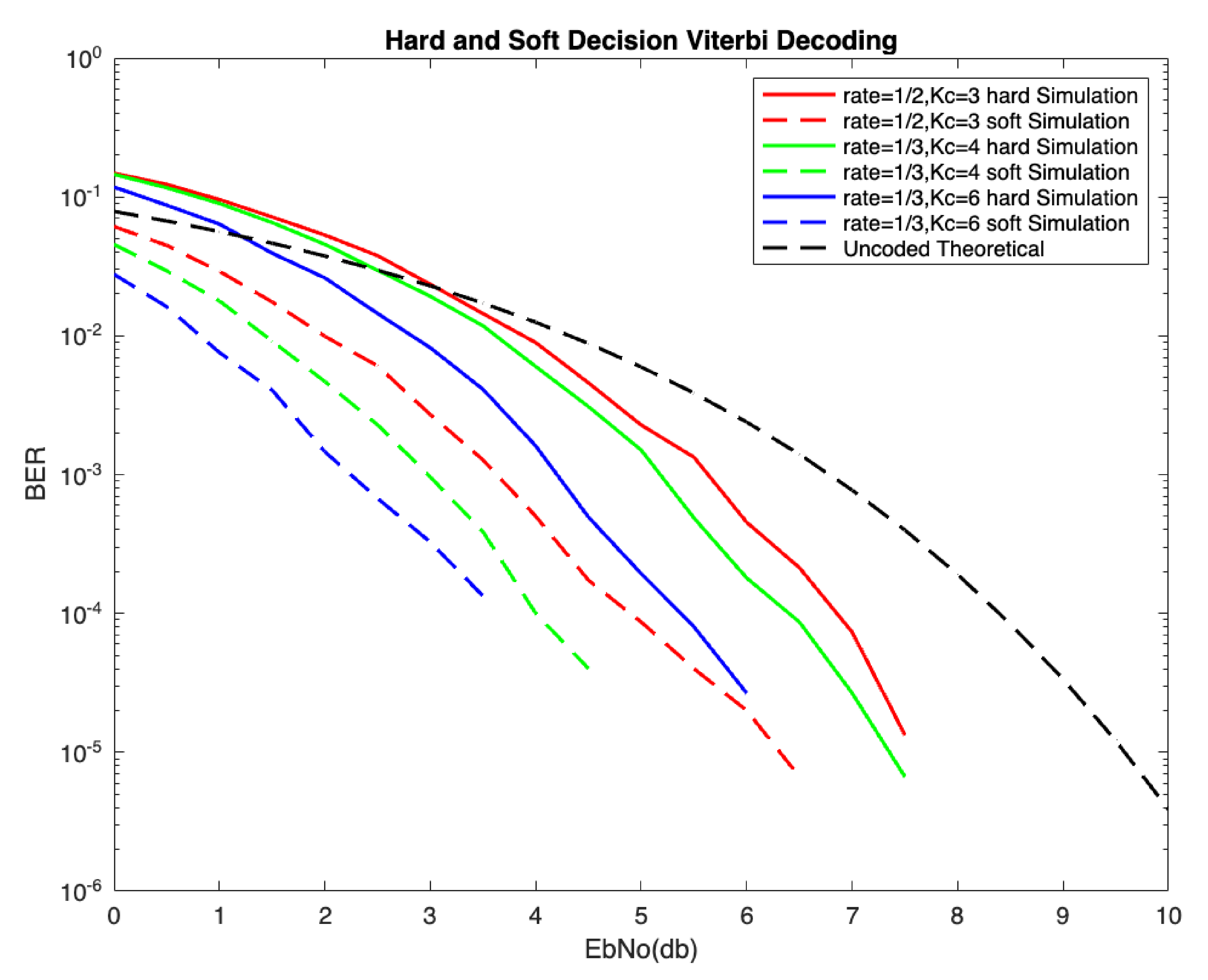
- The graph is for code rate R=1/2 and Kc=3, R=1/3 and Kc=4, R=1/3 and Kc=6
- BER depends on:
  1. Rate of the convolution code
  2. Constraint length of the code
- Behaviour of the Graph:



Analysis of Hard and Soft decoding Graphs

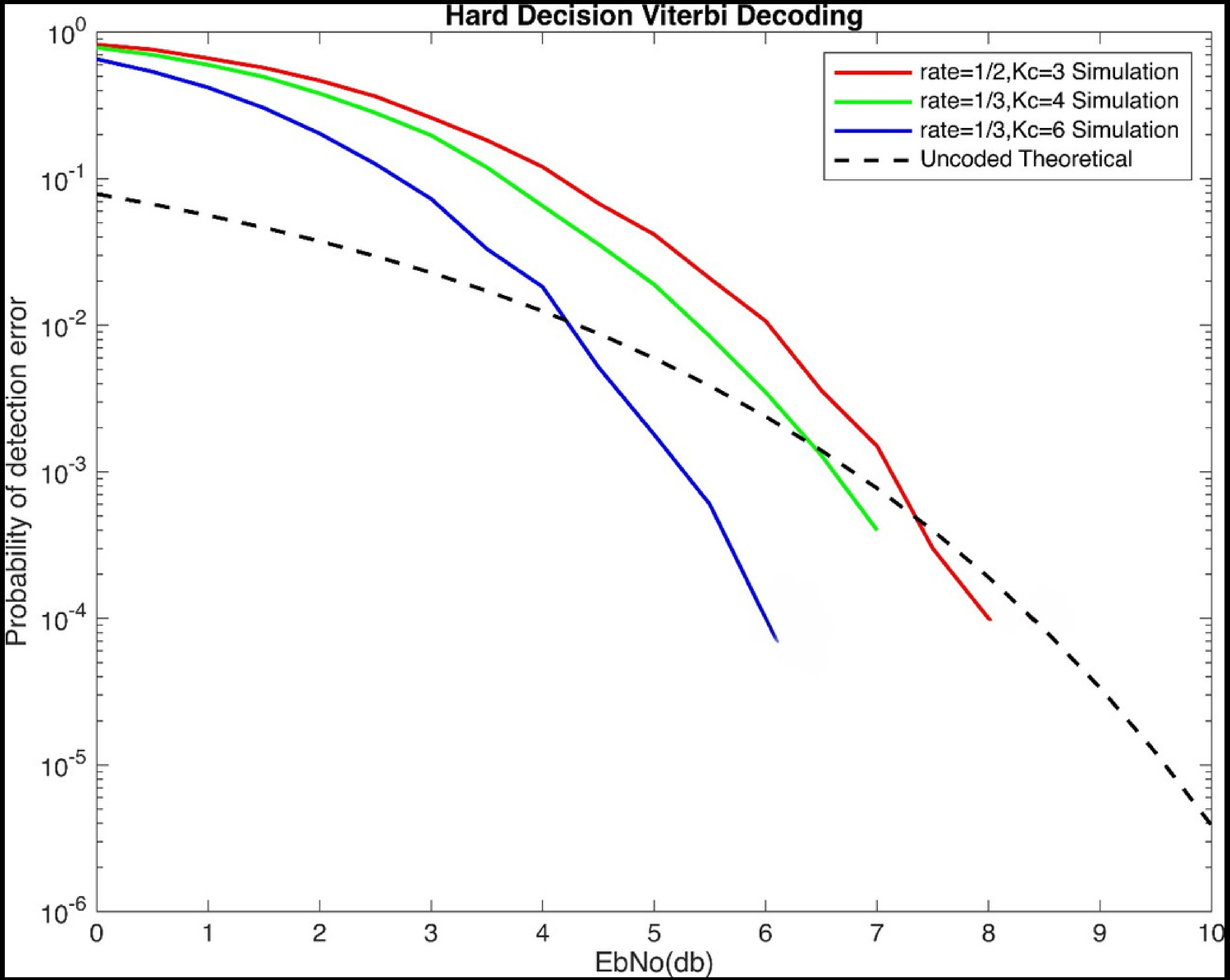
- **Hard Decision Decoding**
  1. Minimum Hamming Distance
  2. In Hard decision we take the received bits as hard 0 and 1.

- **Soft Decision Decoding**
  1. Minimum Euclidian Distance
  2. In soft decision we take the bits as it is.

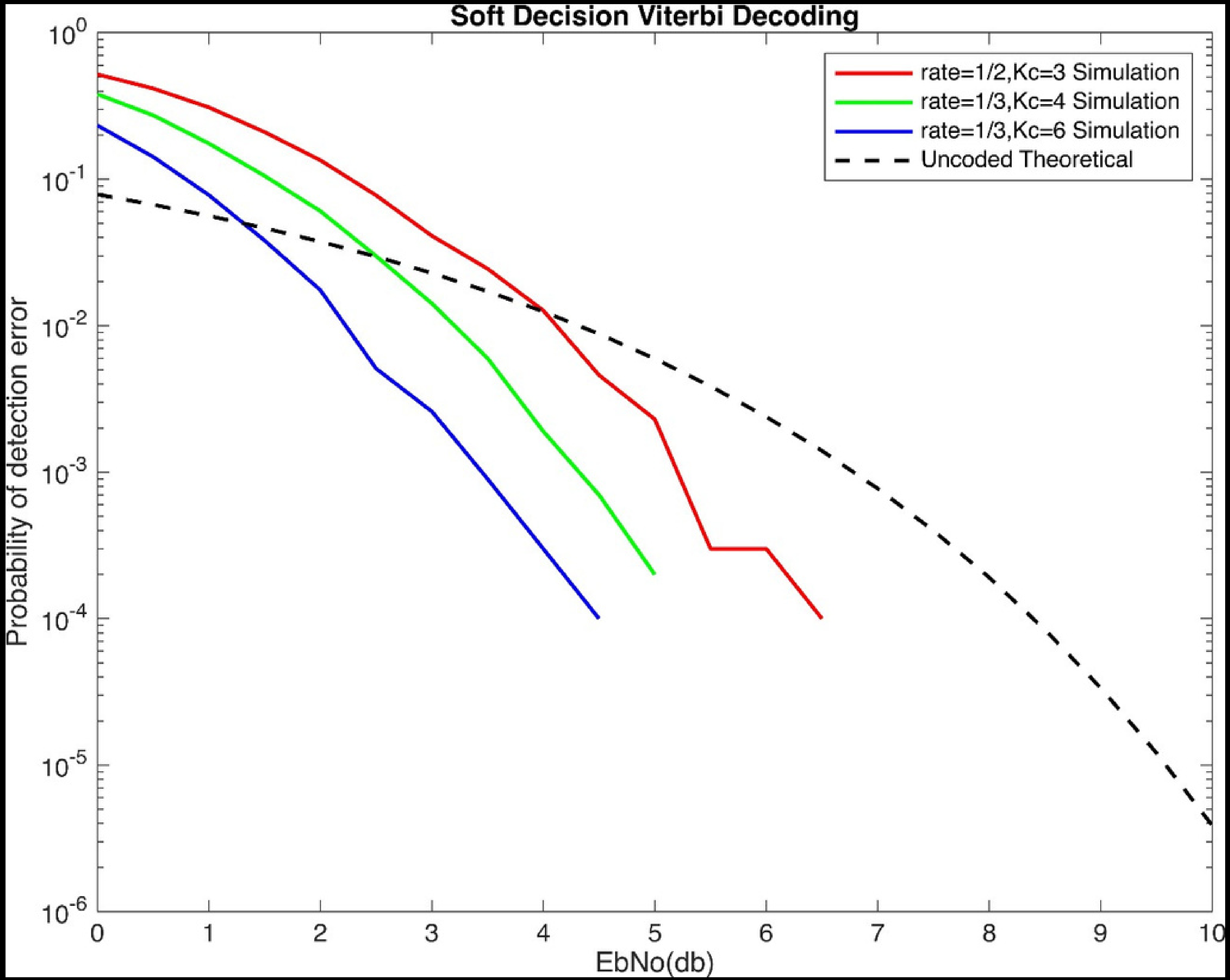


Prob. of Detection Error vs EbNo(dB)

Hard decision viterbi decoding

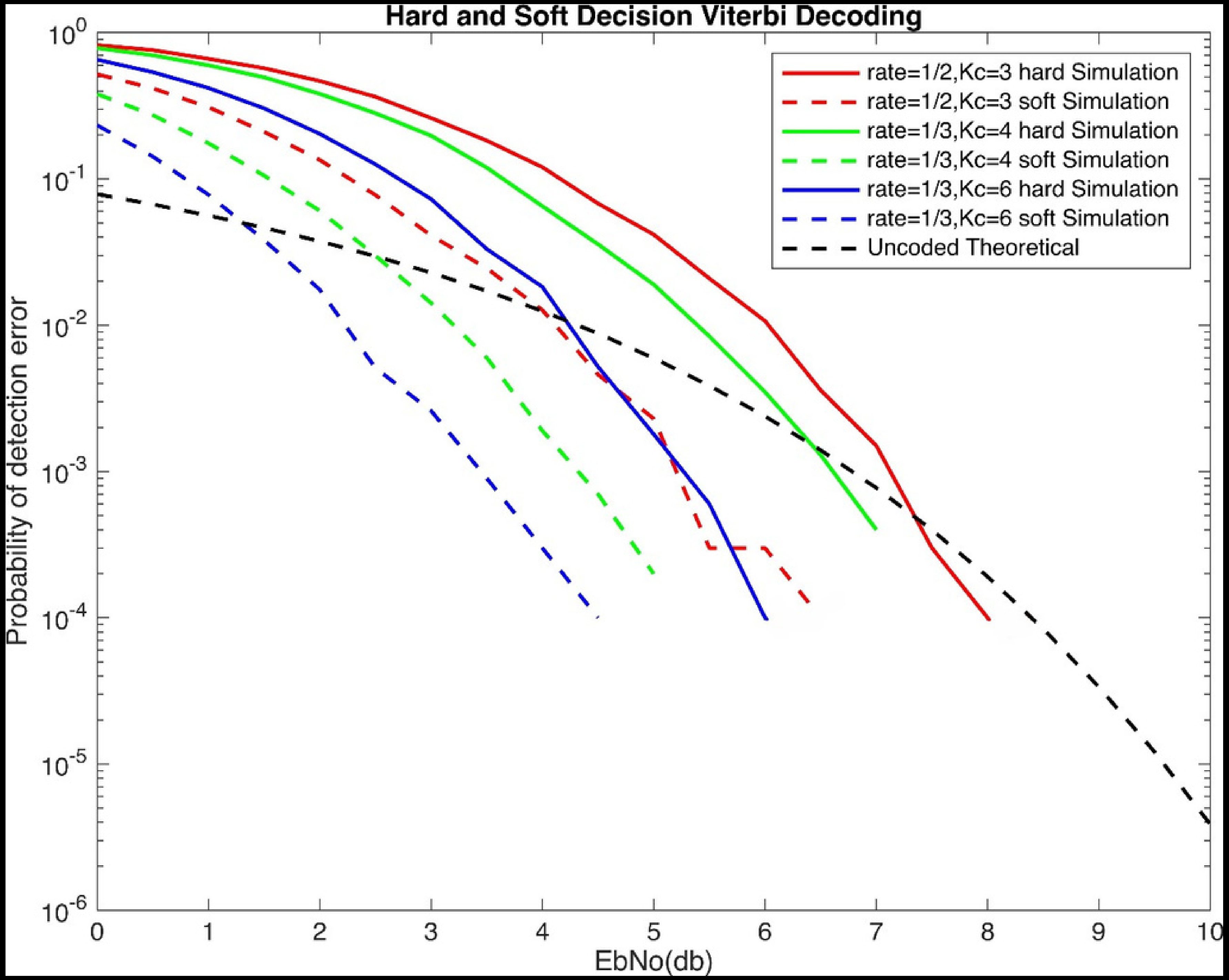


Soft decision viterbi decoding



Prob. of Detection Error vs EbNo(dB)

Hard and soft decision viterbi decoding



Performance of SDD and HDD

- First event error probability where the incorrect path merge with the Correct path for the first time

$$P_2(d) = Q\left(\sqrt{2\gamma_b R_c d}\right)$$

- We sum up the error probability over all possible incorrect path and get upper bound on first event error probability

$$P_e \le \sum_{d=d_{free}}^{\infty} a_d Q\left(\sqrt{2\gamma_b R_c d}\right)$$

- Probability of bit error rate

$$P_b < \sum_{d=d_{free}}^{\infty} B_d Q\left(\sqrt{2\gamma_b R_c d}\right)$$

Summary

- While communicating , we got noise in the data .
- So, to overcome from this problem we use some different types of coding schemes and convolution code is one of them.
- With the use of convolutin coding scheme we can remove most of the errors from the message.
- While encoding, our message will be in form of bits so to transmit it through channel we have to modulate it and after receiving of message we can demodulate it.
- In part of convolution decoding , we have two options:
  1. Hard decision decoding (HDD)
  2. Soft decision decoding (SDD)
- Convolution coding scheme is more suitable for digital communication system because it works well in continuous transmission of data.

Conclusion

- Increasing the SNR leads to increase the performance of both soft and hard Viterbi decoding.
- The performance of the soft Viterbi decoding is better than the Hard Viterbi decoding.
- In hard decision decoding, received symbols are demodulated to bits using a threshold value. This leads to uncertainty about choosing the incorrect bits.
- In soft decision decoding, rather than using demodulation, received symbols are used directly, so there is a low chance of uncertainty.
- Bit Error Rate(BER) and Probability of Detection Error (PDE) will be high for Lower value of SNR and vice-versa, will be low for high value of SNR.
- If the constraint length is increased, the error correction capacity of the convolution code will also be significantly increased.
- But, It will increase the time complexity due to high computational steps as increasing the constraint length.

Bibliography

- Book: Proakis-digital-communications-4th-ed
- Transfer function of convolution codes video lecture by Prof .Subrahmanya K N
- Wikipedia - Convolution coding
- Sciencedirect - Convolution coding
- Convolution Codes ppt of Matthew C. Valenti - Lane Department of Computer Science and Electrical Engineering, West Virginia University

Team Members

202201227 - Nishank Kansara

202201228 - Jaimin Prajapati

202201233 - Harsh Baraiya

202201234 - Anshu Dhankecha

202201241 - Divyesh Ramani

202201242 - Dev Davda

202201250 - Parth Prajapati

202201251 - Mehul Vagh

202201256 - Ayush Pandita

202201258 - Nishant Italiya

Thank you!