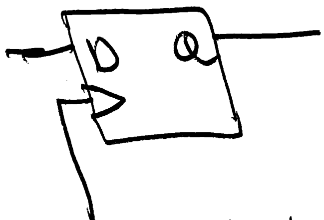
Convolutional Codes:

Memory cells: Electronic circuit capable of holding 1-bit information

Memory cells are nothing but flip-flops.

Ex: D-flip flop



CP → clock pulse

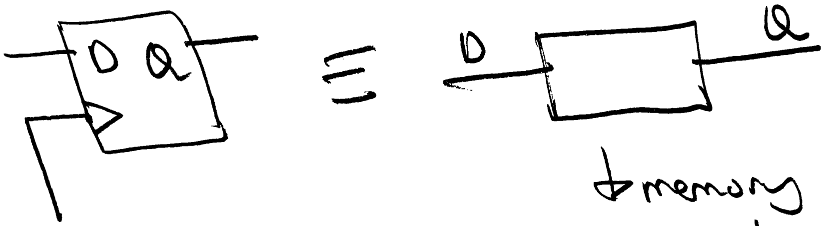
when CP ↑ $Q(t+1) = D$

D	Q(t+1)
0	0
1	1

→ $Q(t+1) = D$

↓

output of flip-flop after clock pulse.

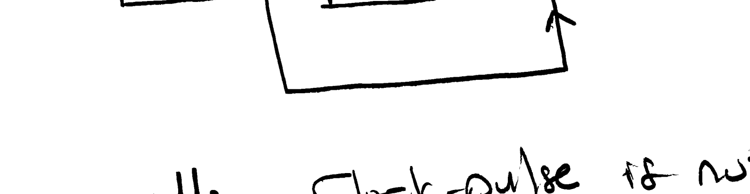


↓ memory cell graphical illustration.

A convolutional encoder consists of memory cells and mod-2 summers.

Ex:

↓ mod-2 summer



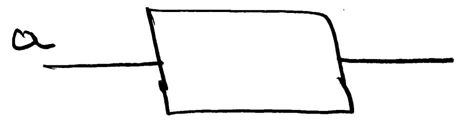
In memory cells Clock-pulse is not shown

we trace the memory cells behaviour

by giving ~~…~~ virtual clocks by ourselves

(2)

Ex:

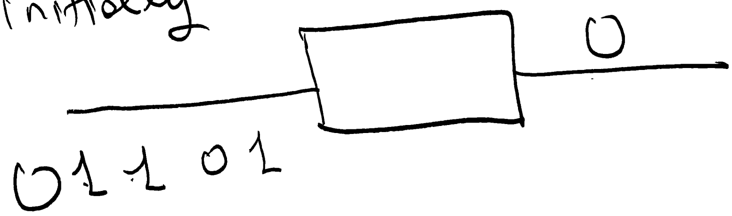


a = 10110 find memory cell output at each clock

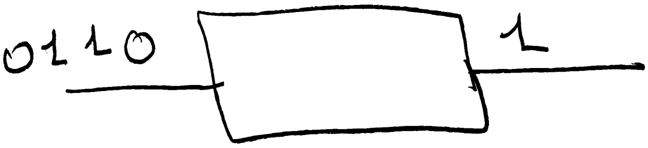
Sln.: a = 10110

↓ first bit to be sent to the cell

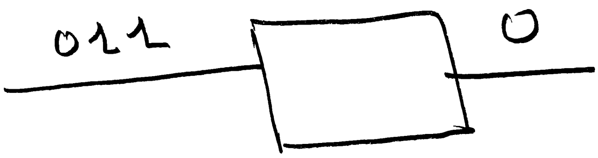
initially



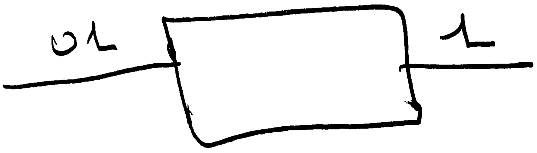
CP1:



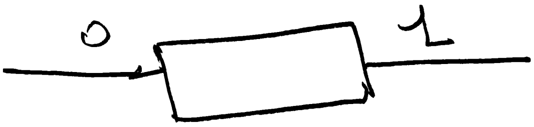
CP2:



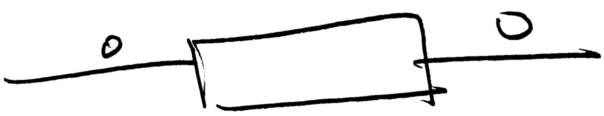
CP3:



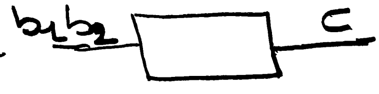
CP4:



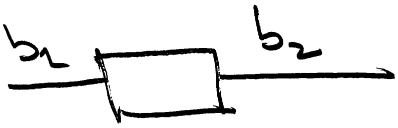
CP5



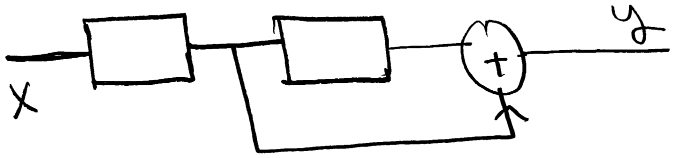
In general



after clock pulse



(3) **Ex:**

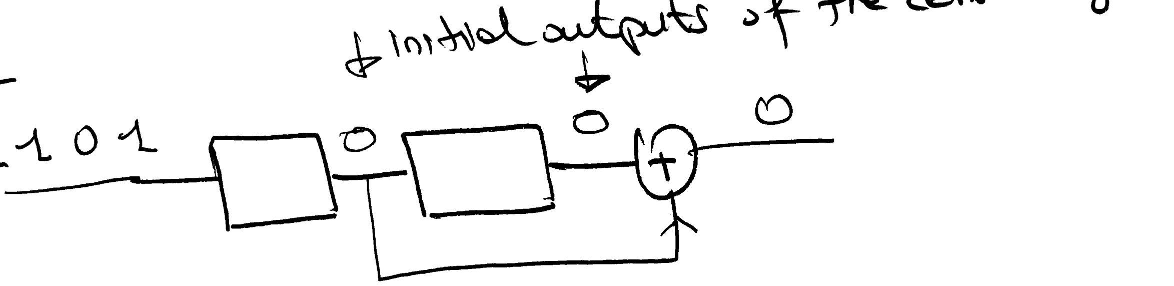


$X = 1011$

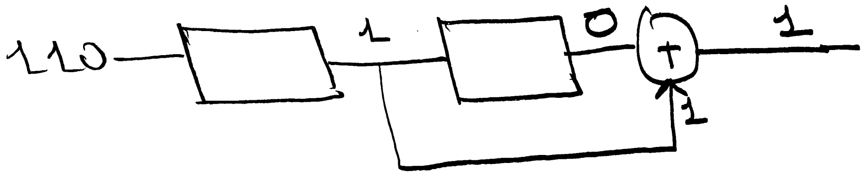
find y

S/n:

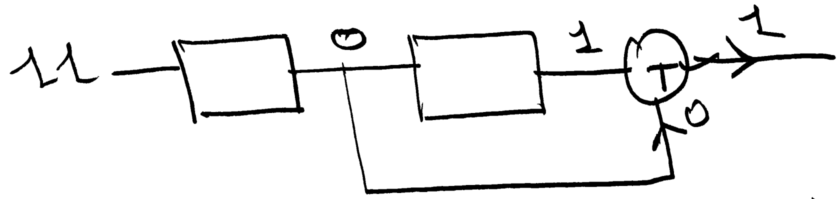
Initially



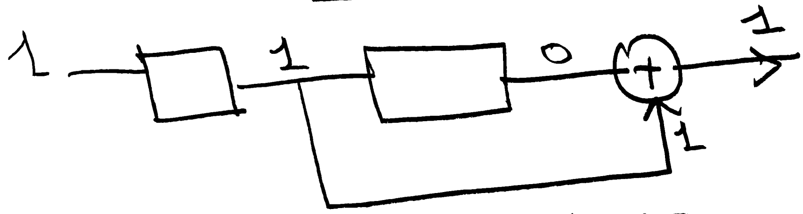
After CP1:



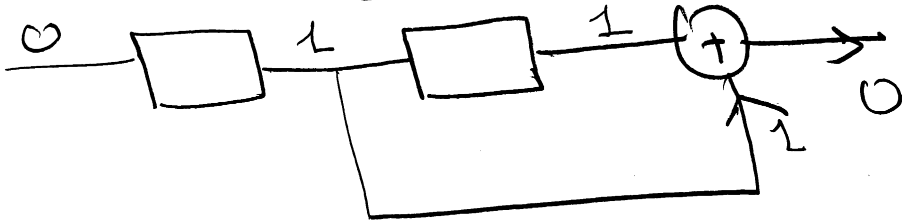
After CP2:



After CP3



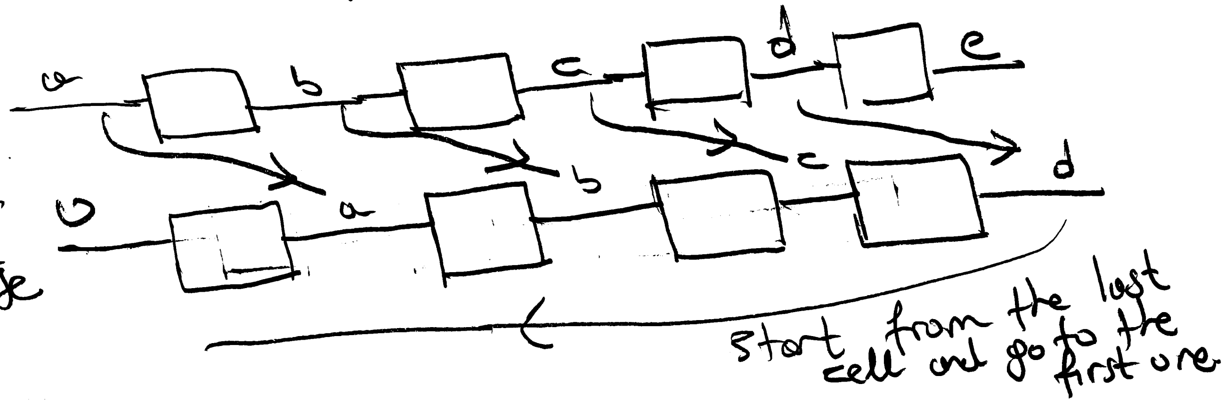
After CP4



Hence output string is

1110

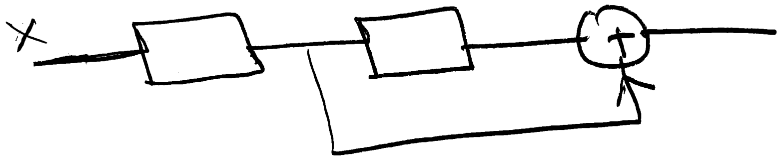
Remarks:



④ Impulse response of a convolutional encoder:

If your input is 1000...0 your output is called impulse response

Ex: Find the impulse response of the following convolutional encoder



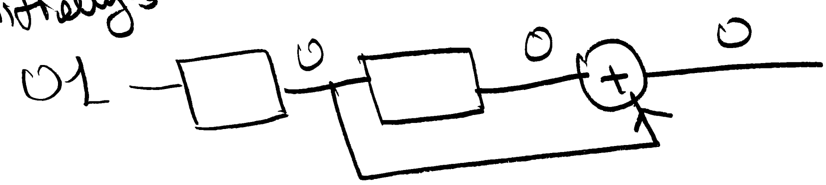
Sln: Input is $x = \underbrace{1\,0}_{}$ only 2-bits

Since convolutional encoder includes 2-memory cells

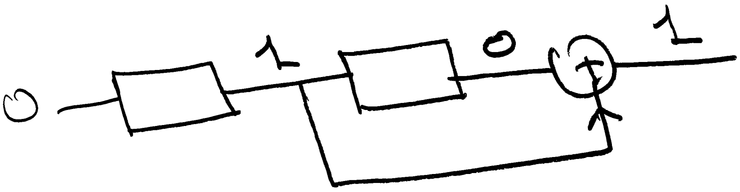
In general

$x = \underbrace{1\,0 \cdots 0}_{m\text{-bits}}$ m is the number of memory cells

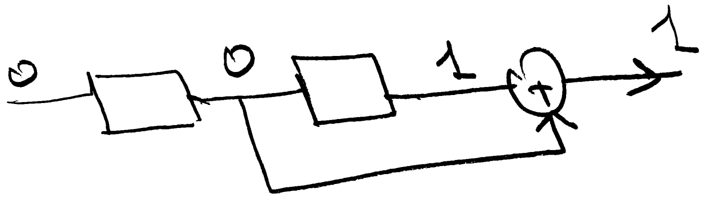
Initially:



After CP1:



After CP2:



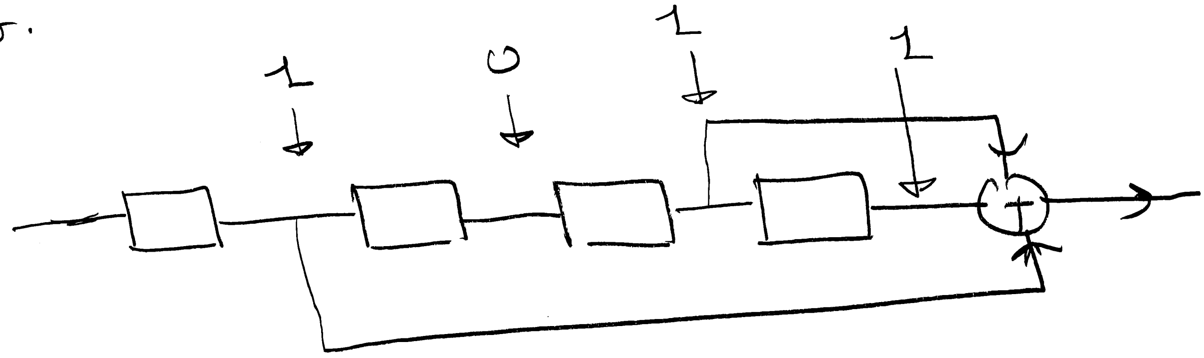
Output is 11

Hence, impulse response is 11

(5) Easy way of finding the impulse response

Check the connection between output of a cell and final output if there is a connection write 1 otherwise write 0, and collect the bits.

Ex:



Impulse response is 1011

Convolution:

x = 1011 Find x * y → convolution of x & y

y = 110

Take one of them and write it

e.g. 1011 → x is taken

reverse the other

011 → y is reversed.

Compute the convolution as follows

1011
011 } → product, and mod 2 sum gives 1

1011
011 } → 1

1011
011 } → 1

$$\left.\begin{array}{l}1011\\ \;\;011\end{array}\right\} \rightarrow 1+1 \Rightarrow 0$$

$$\left.\begin{array}{l}1011\\ \;\;\;\;011\end{array}\right\} \rightarrow 1$$

$$\left.\begin{array}{l}1011\\ \;\;\;\;\;\;011\end{array}\right\} \rightarrow 0$$

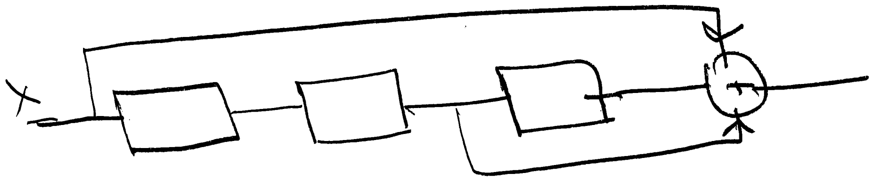
So output is 1 1 1 0 1 0

Remark: If you know the impulse response of a convolutional code, for an arbitrary string x the output of the convolutional encoder can be found as

$$y = x \otimes \underbrace{h}_{\text{impulse response}}$$

This is why they are called as convolutional codes.

Exe:



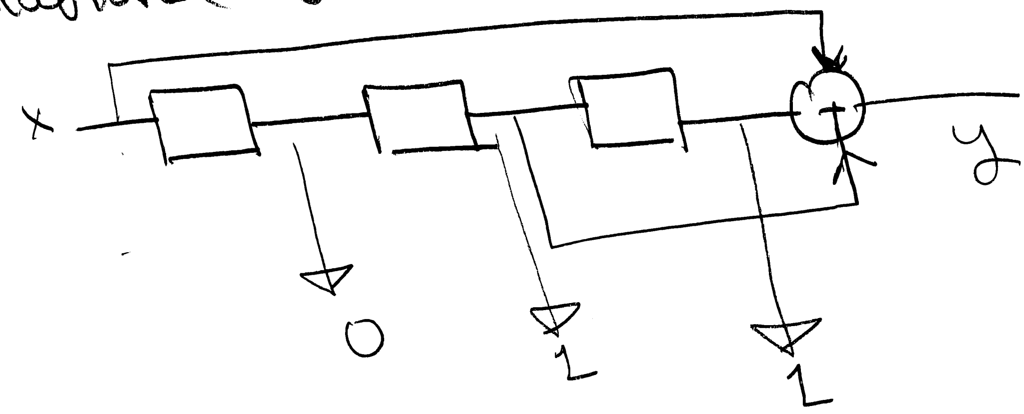
If $x = 11011$ ← first bit, find output of the encoder

Sln:

We can compute the output bit every clock pulse and collect the output bits and find the output string

7

or lets first the impulse response of the convolutional encoder



$h = (011)$ → impulse response.

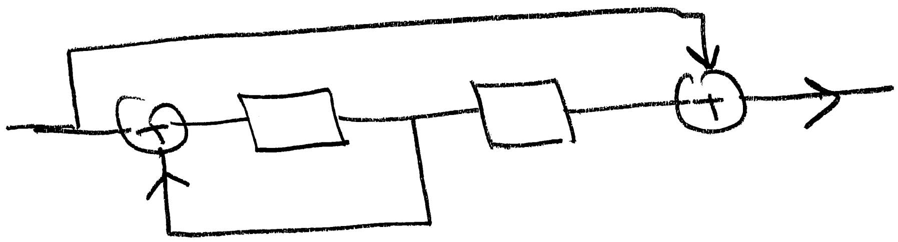
$x = (11011)$ → input

$y = x \otimes h$ →

11011
110 → 0

so on like this

Recursive Convolutional Encoders



If a convolutional encoder includes a feedback it is a recursive convolutional encoder.

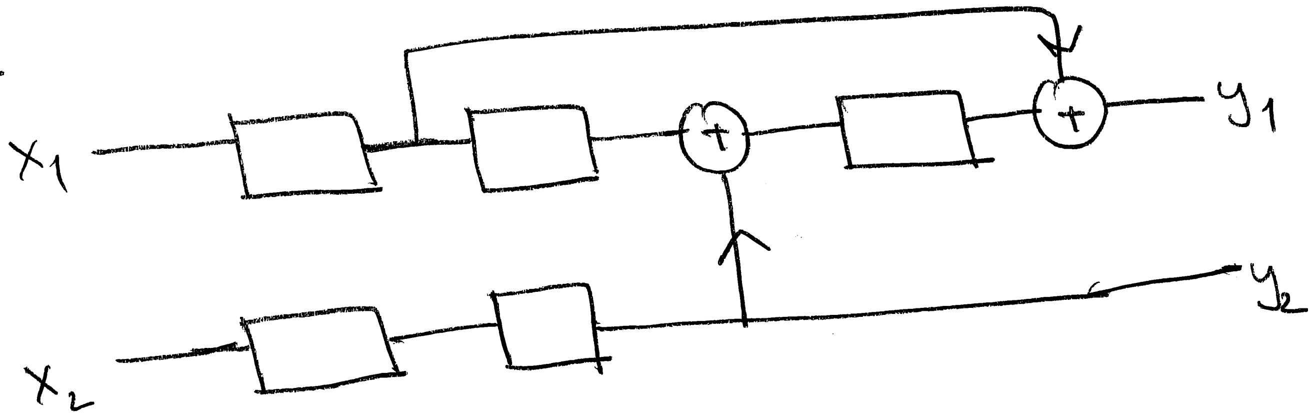
Recursive convolutional encoders may have infinite impulse response

i.e; $h = 10110\ldots$

(8)

A convolutional encoder may have more than one-input stream and more than one output stream

Ex:



In this case an impulse response is defined between every input and every output pair

i.e., impulse responses between

$x_1 - y_1 \rightarrow h_1$

$x_1 - y_2 \rightarrow h_2$

$x_2 - y_1 \rightarrow h_3$

$x_2 - y_2 \rightarrow h_4$

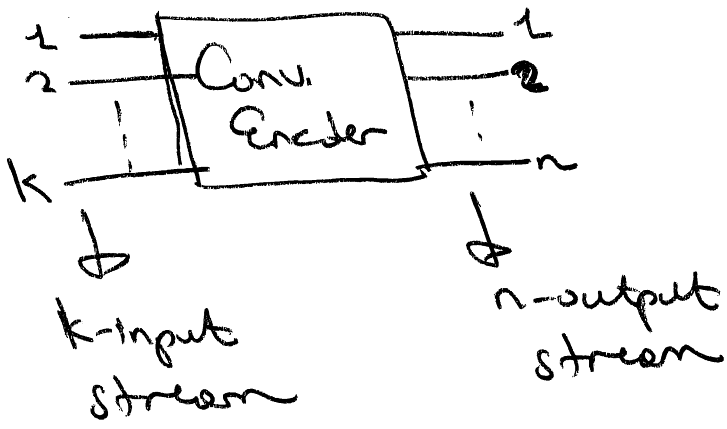
4-impulse responses

$h_1 = 1\ 0\ 1 \rightarrow x_1 = 1000 \rightarrow$ output (y_1) $h_1 = 101$

$h_2 = 0\ 0\ 0 \rightarrow x_1 = 1000 \rightarrow$ output y_2 $h_2 = 000$

$h_3 = 0\ 1 \rightarrow x_2 = 10 \rightarrow$ output y_1 $h_3 = 01$

$h_4 = 0\ 1 \rightarrow x_2 = 10 \rightarrow$ output y_2 $h_3 = 01$



h^{ij} → impulse response between input i and output j

$i = 1 \ldots k$
$j = 1 \ldots n$

In total there are $k \times n$ impulse responses

<u>Remark:</u> If there are m memory cells in a convolutional encoder, the impulse response h^{ij} contains m-bits

h^{ij}_k → kth bit of h^{ij}

Generator matrices for convolutional codes

$c = d \circledast h$ → codeword = dataword $\circledast$ imp. resp.

$c = dG$ → G is constructed from the components of the impulse sequences

(10)

For an (n,k,m) Convolutional code

$k \rightarrow$ number of input ports
$n \rightarrow$ number of output ports
$m \rightarrow$ number of memory cells

G_i is the $k \times n$ matrix

given as

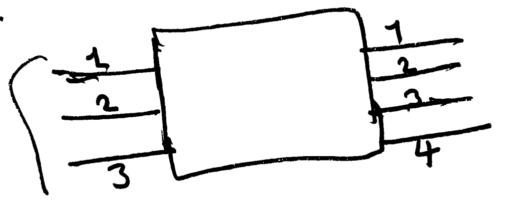
$$G_i = \begin{bmatrix} h_i^{(1,1)} & h_i^{(1,2)} & \cdots & h_i^{(1,n)} \\ h_i^{(2,1)} & h_i^{(2,2)} & \cdots & h_i^{(2,n)} \\ \vdots & & & \\ h_i^{(k,1)} & h_i^{(k,2)} & \cdots & h_i^{(k,n)} \end{bmatrix}$$

the generator matrix G is constructed as

$$G = \begin{bmatrix} G_0 & G_1 & G_2 & \cdots & G_m & \bar{0} & \bar{0} & \cdots \\ \bar{0} & G_0 & G_1 & \cdots & \cdots & G_m & \bar{0} & \cdots \\ \bar{0} & \bar{0} & G_0 & \cdots & \cdots & & G_m & \\ \vdots & & & & & & & \cdots \\ \vdots & & & & & & & \cdots \end{bmatrix}$$

Ex: $(4,3,2)$ convolutional code
$\downarrow n \quad \downarrow k \quad \downarrow m$

input ports $\leftarrow$



n output ports

(11) The impulse responses between input ports and output ports are given as

$$h^{(1,1)} = [1\ 0\ 0] \quad h^{(2,1)} = [0\ 0\ 0] \quad h^{(3,1)} = [0\ 0\ 0]$$

$$h^{(1,2)} = [1\ 0\ 0] \quad h^{(2,2)} = [1\ 1\ 0] \quad h^{(3,2)} = [0\ 1\ 0]$$

$$h^{(1,3)} = [1\ 0\ 0] \quad h^{(2,3)} = [0\ 1\ 0] \quad h^{(3,3)} = [1\ 0\ 1]$$

$$h^{(1,4)} = [1\ 0\ 0] \quad h^{(2,4)} = [1\ 0\ 0] \quad h^{(3,4)} = [1\ 0\ 1]$$

$$G_0 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad G_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

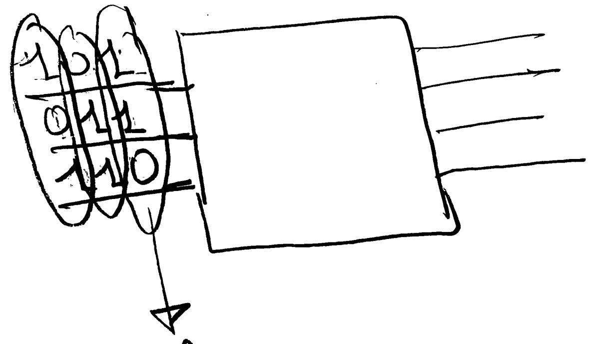
in a similar manner as in G_0, the matrices G_1 & G_3 are formed.

$$G = \begin{bmatrix} G_0 & G_1 & G_2 & \bar{0} & \bar{0} & \cdots \\ \bar{0} & G_0 & G_1 & G_2 & \bar{0} & \cdots \\ \bar{0} & \bar{0} & G_0 & G_1 & G_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \end{bmatrix}$$

Size of G depends on information block length.

(12)

Ex: For $d=(1\,1\,0\,|\,0\,1\,1\,|\,1\,0\,1)$



first bit entering into the convolutional group encoder

$$G=\begin{bmatrix} G_0 & G_1 & G_2 & \bar{0} & \bar{0} \\ \bar{0} & G_0 & G_1 & G_2 & \bar{0} \\ \bar{0} & \bar{0} & G_0 & G_1 & G_2 \end{bmatrix} \quad \text{where} \quad \bar{0}=\begin{bmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{bmatrix}$$

$$G_0=\begin{bmatrix} 1&1&1&1 \\ 0&1&0&1 \\ 0&0&1&1 \end{bmatrix} \qquad G_1=\begin{bmatrix} 0&0&0&0 \\ 0&1&1&0 \\ 0&1&0&0 \end{bmatrix}$$

$$G_2=\begin{bmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&1&1 \end{bmatrix}$$

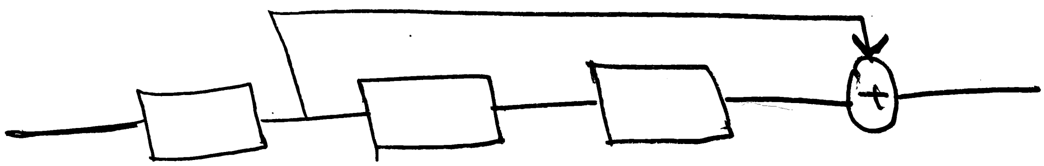
Generator polynomials for convolutional codes

$$v = u * h \rightarrow v(D) = \underbrace{u(D)}_{\text{polynomial for } u}\ \underbrace{h(D)}_{\text{polynomial for } h}$$

$h(D) \rightarrow$ is the generator polynomial of the convolutional code and shown by $g(D)$

13

Ex:



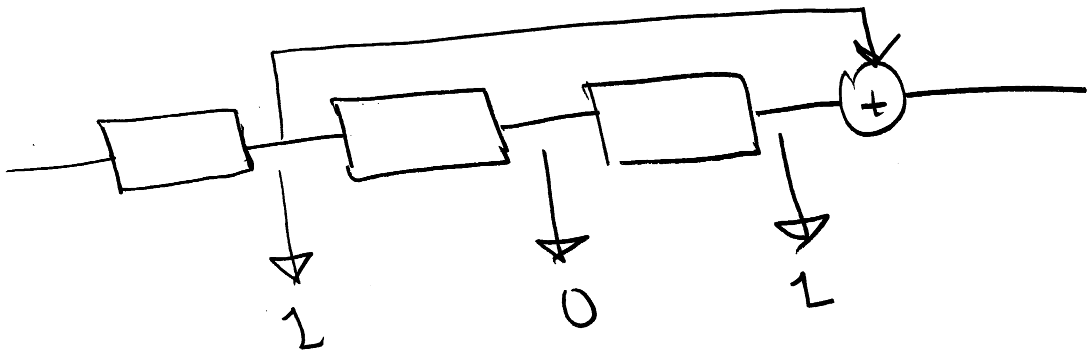
a) $h = ?$ impulse response

b) $h(D) = ?$ (i.e, $g(D) = ?$)

c) encode the bit stream

$d = (1\ 0\ 1\ 1)$ using $g(D)$

Sln:



$h = (1\ 0\ 1)$

$g(D) = h(D) = D^2 + 1$

$d(D) = D^3 + D + 1$

$$\begin{aligned} c(D) &= g(D)\, d(D) \\ &= (D^2+1)(D^3+D+1) \\ &= D^5 + \cancel{D^3} + D^2 + \cancel{D^3} + D + 1 \\ &= D^5 + D^2 + D + 1 \end{aligned}$$

$c = (1\ 0\ 0\ 1\ 1\ 1) \xleftrightarrow[\text{check}]{?} (1\ 0\ 1\ 1) \circledast (1\ 0\ 1)$ ✓

(14)

Remark

If $d=(d_0\ d_1\ d_2 \dots)$

then $d(D) = d_0 + d_1 D + d_2 D^2 + \dots$

OR if you adopt d as

$d=(\dots\ d_2\ d_1\ d_0)$

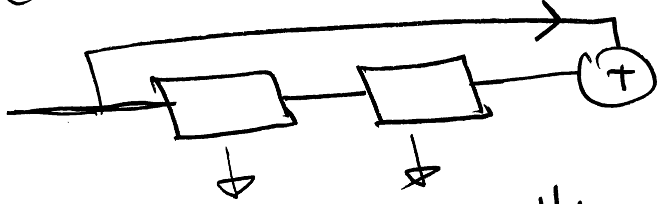
$d(D) = \dots + d_2 D^2 + d_1 D + d_0$

adopt one of them for d and h i.e, g

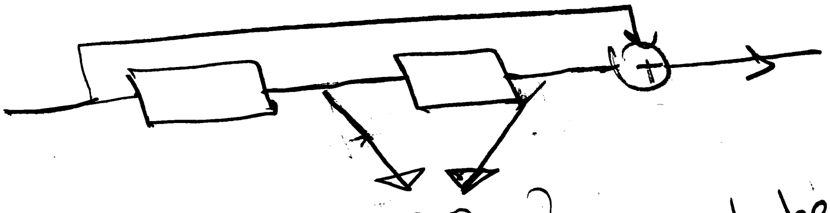
Graphical Representation of Convolutional Codes

States: The content of the memory cells is accepted as state.
(The content of memory cells = output of memory cells)

Ex:

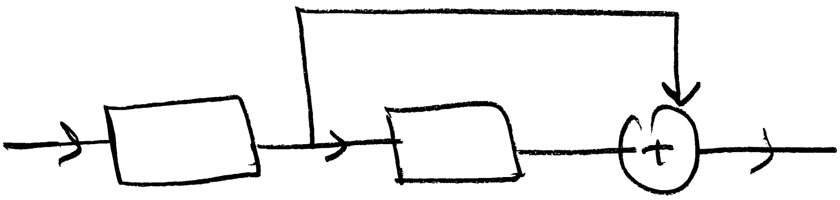


memory cells



can be
00
01
10
11
→ states of the conv. encoder.

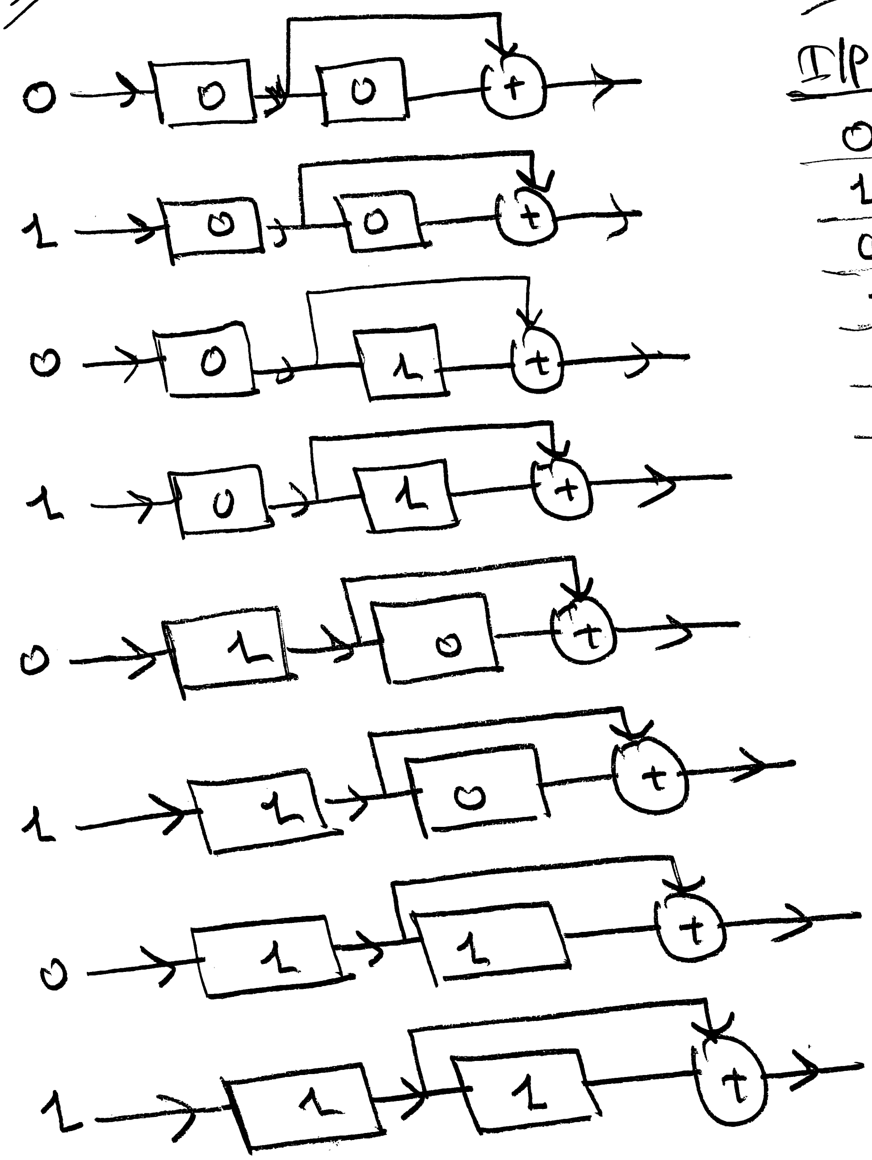
Ex:



CP → Clock pulse

Draw state diagram of the above conv. encoder

S/n:

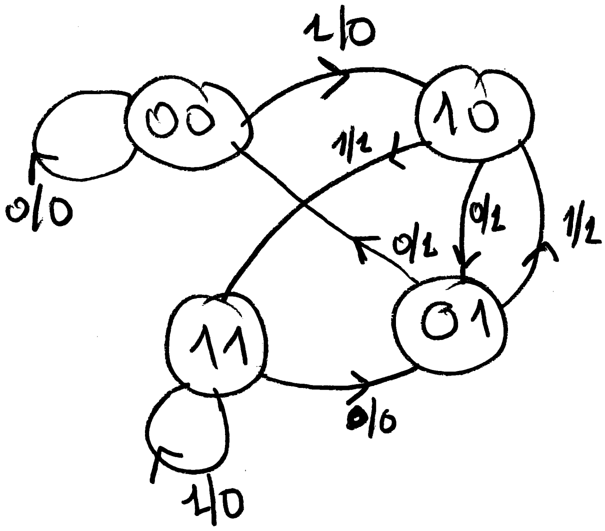


I/P (Before CP)	O/P (Before CP)	Present State (Before CP)	Next State (After CP)
0	0	00	00
1	0	00	10
0	1	01	00
1	1	01	10
0	1	10	01
1	1	10	11
0	0	11	01
1	0	11	11

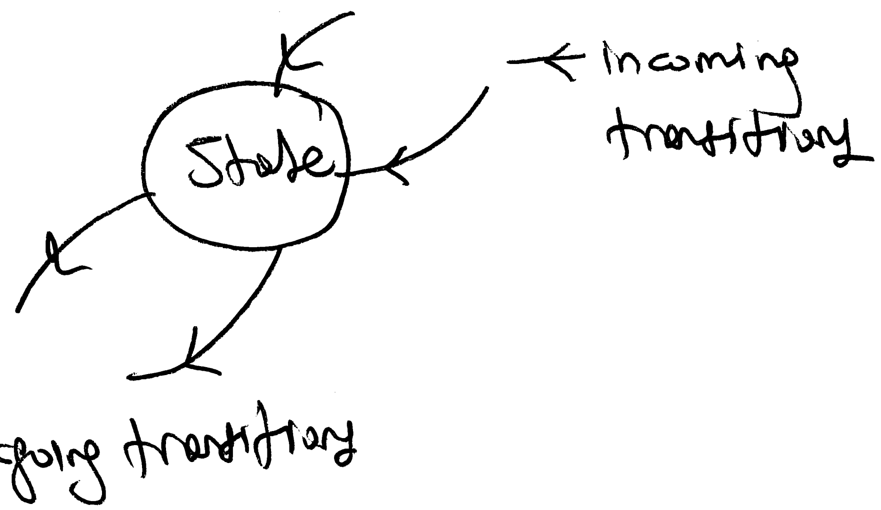
This table can be expressed using state diagram

16

State diagram of the example is drawn as

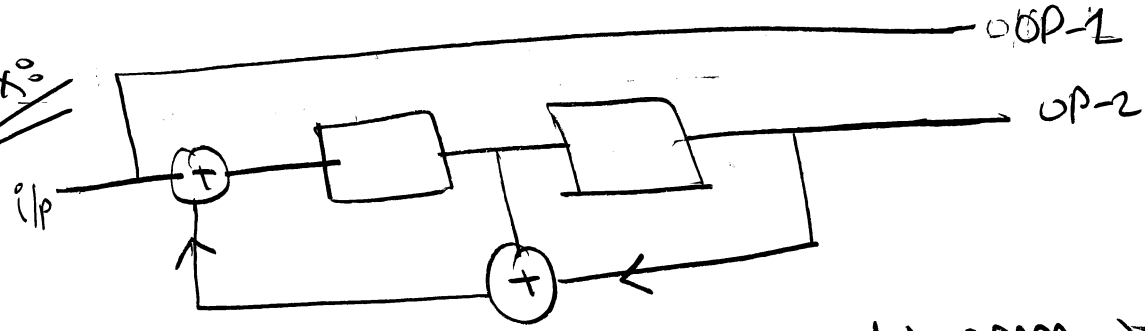


Every state has two incoming transitions and two outgoing transitions



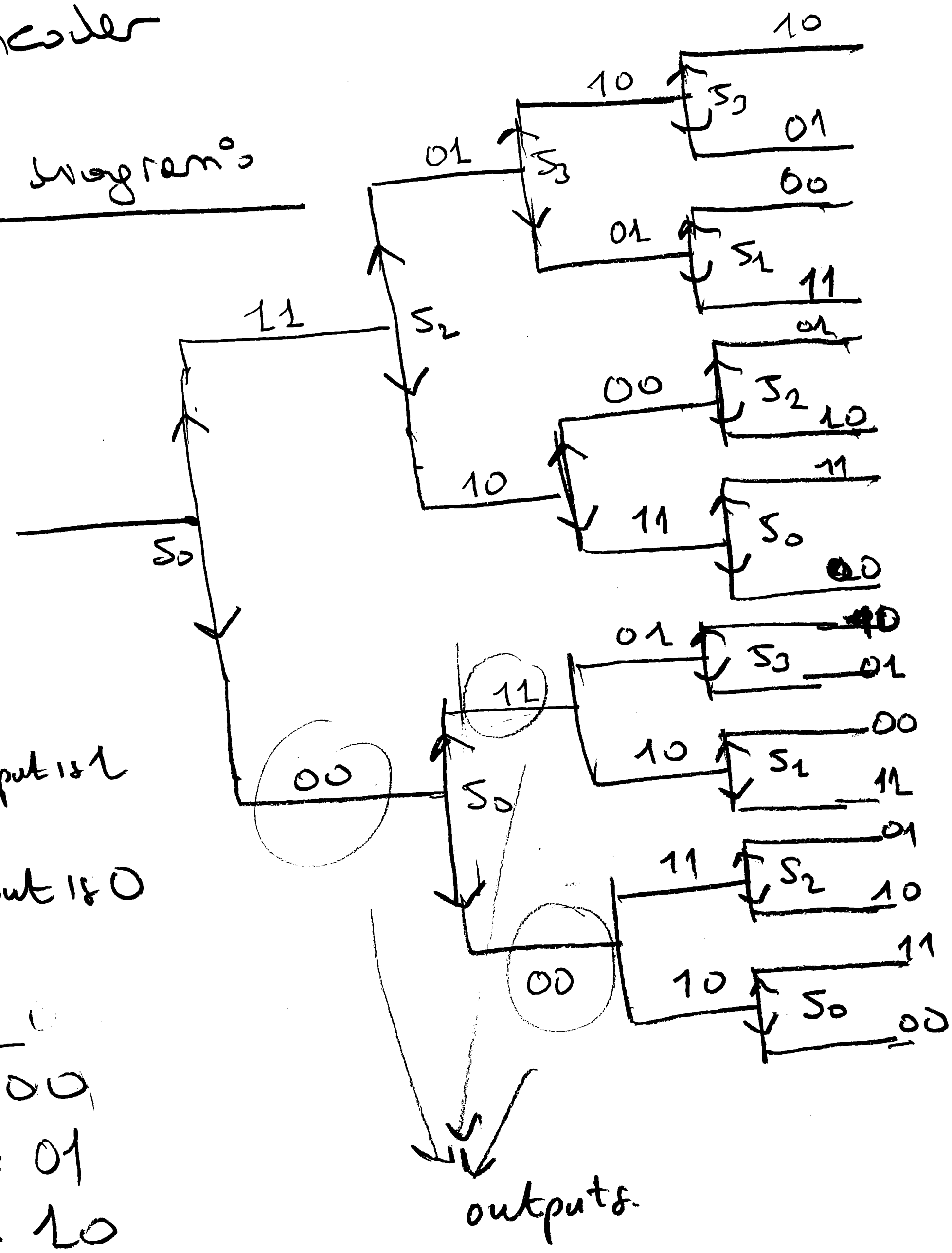
17

Ex:



obtain the state transition diagram of the above recursive convolutional encoder

Tree diagram:



↑ when input is 1

↓ when input is 0

States

$S_0 = 00$

$S_1 = 01$

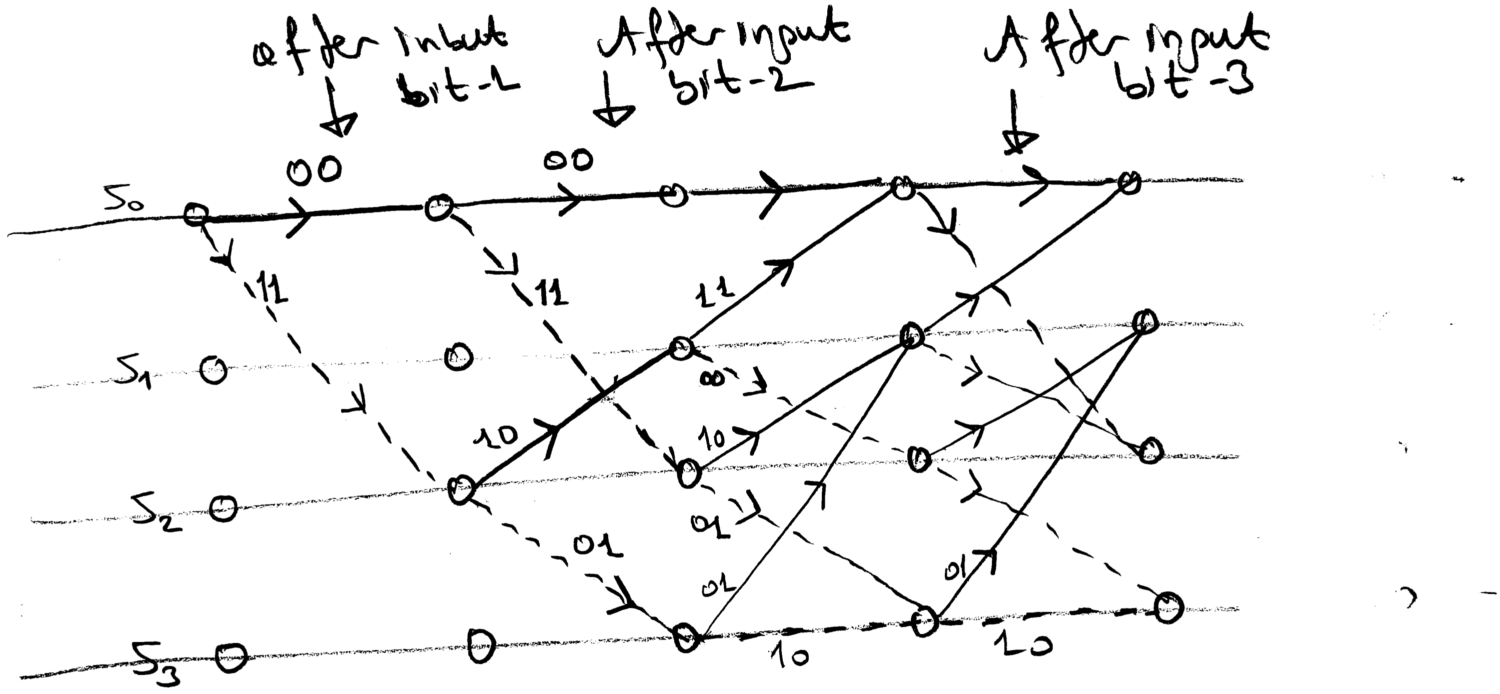
$S_2 = 10$

$S_3 = 11$

(18)

Trellis diagrams

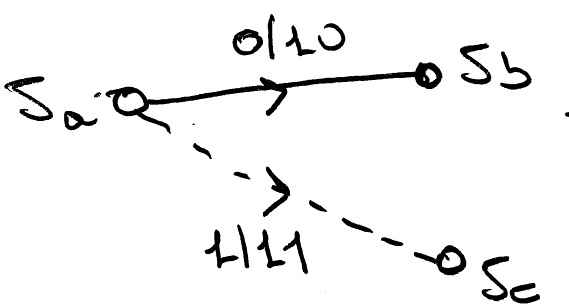
States: 00 → S_0
01 → S_1
10 → S_2
11 → S_3



Initial state is S_0

As a new bit in taken transition to a new state occurs and output bits are formed

In general a state has the following transition graph in trellis diagram



→ Last State is S_a

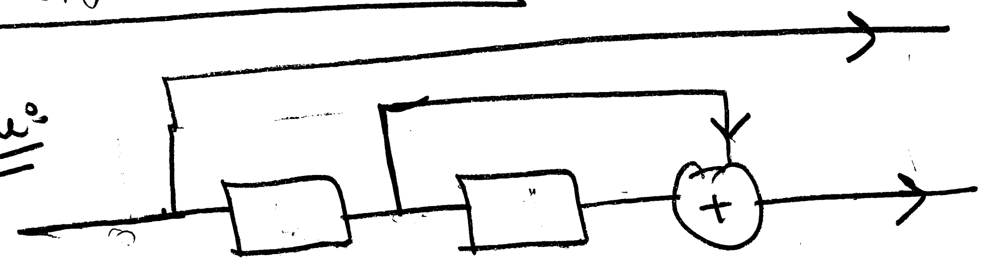
If input is 0 encoder outputs 10 and makes a transition to state S_b

If input is 1, encoder outputs 11 and makes a transition to S_c

(18) Trellis diagram is used to decode the codewords. Viterbi algorithm is used to decode the codewords.

The Viterbi Decoder:

Example:

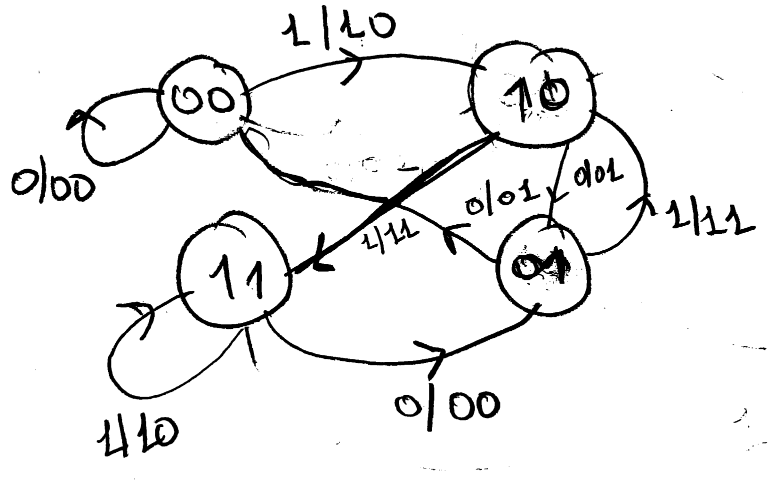


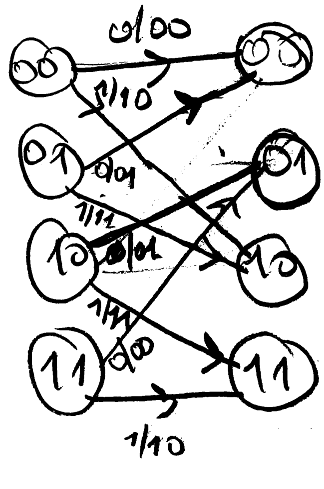
a) Obtain state transition diagram
b) obtain trellis diagram (using state diagram)

c) Encode the vector u=(1011)
d) Let c be the codeword for u in (c)
decode u using Viterbi algorithm.

Sln: a) Encoder has 1-input port and 2 output ports
(2,1) → convolutional encoder.

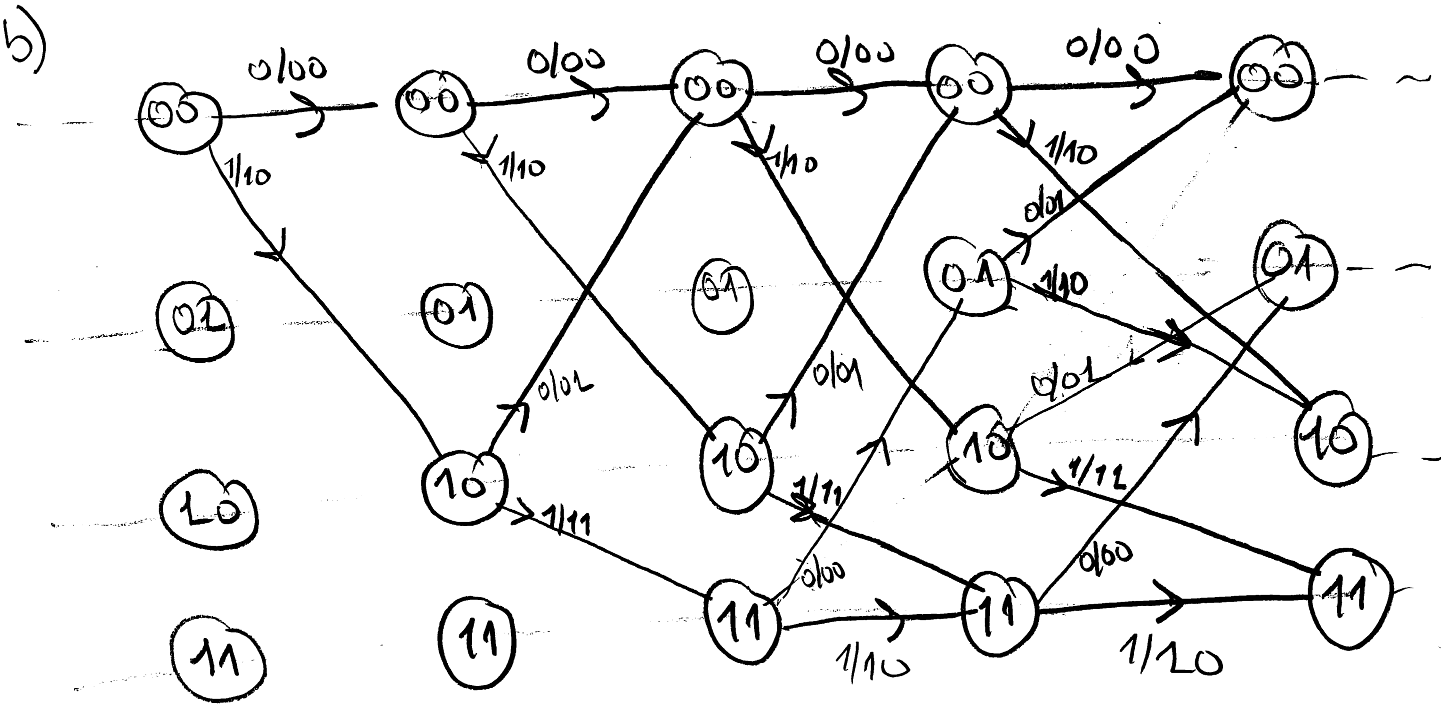
State diagram is as follows





20

b)



c) $w = (1\ 0\ 1\ 1)$

Using the state diagram the codeword is obtained as

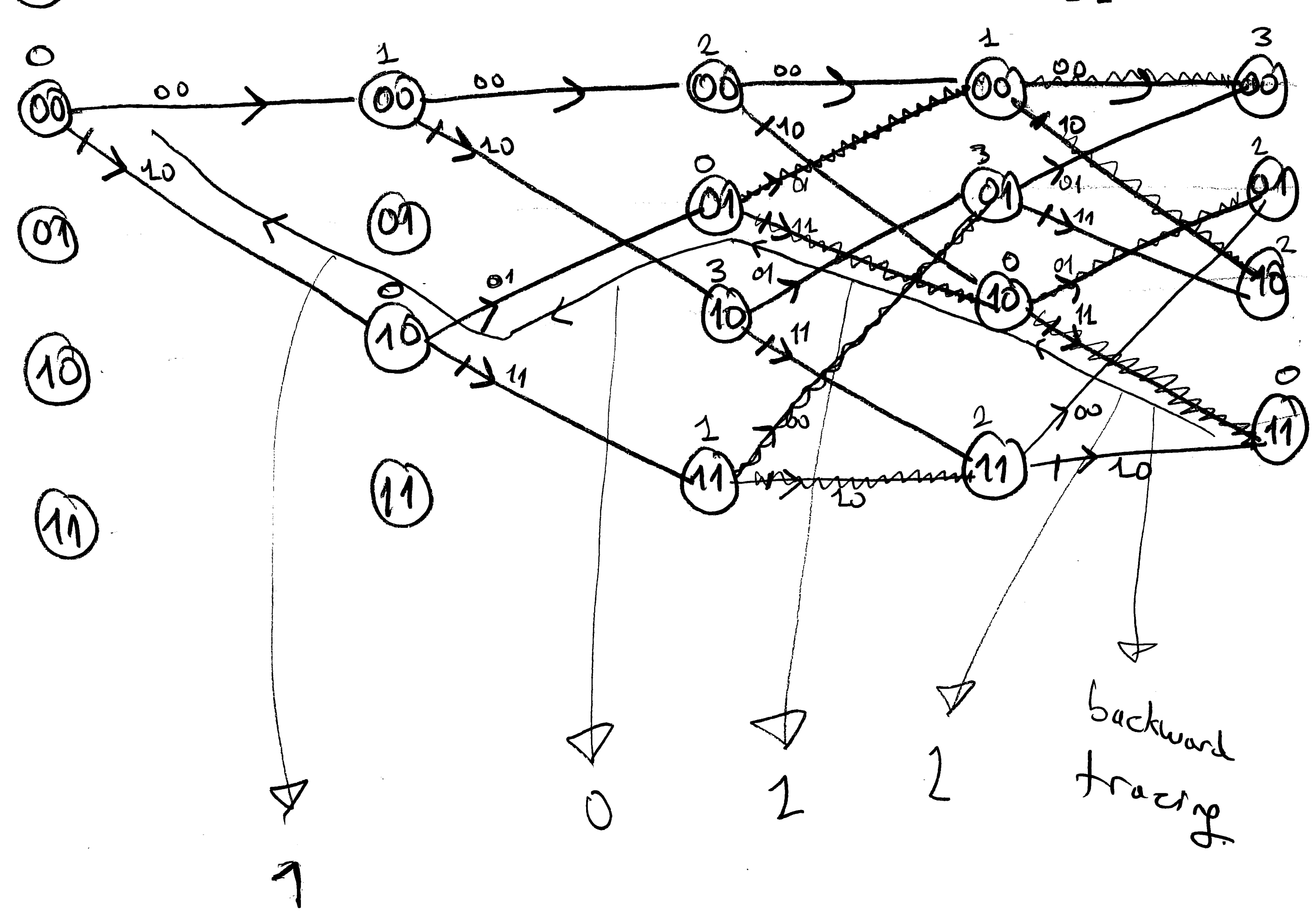
$$w = (1 \quad 0 \quad 1 \quad 1)$$

$$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$$

$$10 \quad 01 \quad 11 \quad 11$$

$c = (10\ 01\ 11\ 11)$

d) Lets decode the codeword c using the Viterbi algorithm.

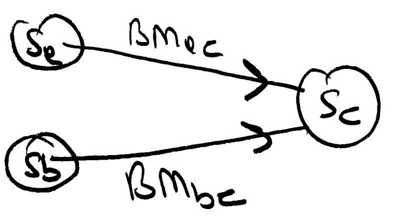


Remarks

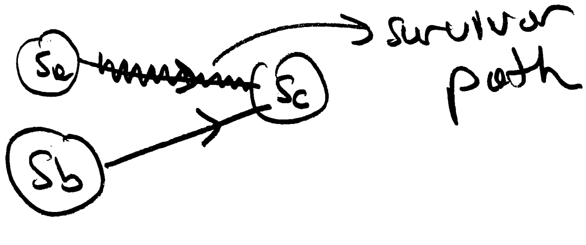
Branch metric = Hamming distance between code-bit pairs and encoder output pairs

Every state has a state metric

State metric (S_c) = = min($S_a + BM_{ac}$, $S_b + BM_{bc}$)

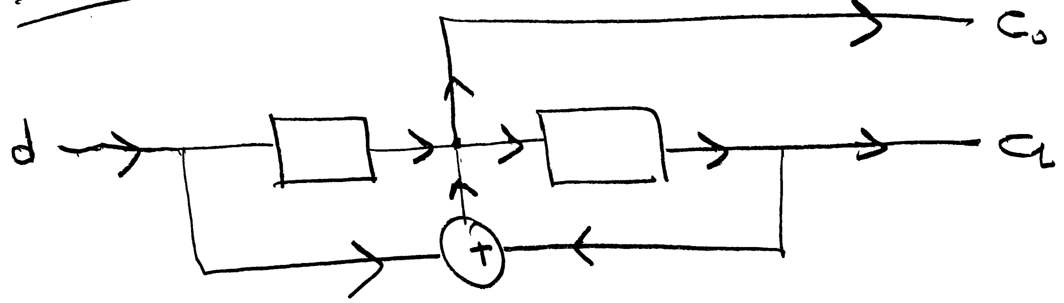


If $S_a + BM_{ac}$ is minimum



22

Exercise:



a) Obtain the state diagram of the above circuit

b) Encode the dataword using state diagram

$d = (11011)$

Let w be the codeword

d) Decode w using the Viterbi decoder

c) Draw trellis diagram using state diagram