MODULE 5
CONVOLUTIONAL CODES

Ms. Anila Kuriakose,
Assistant Professor, Department of ECE, RSET

Contents

- Convolutional Codes

  – encoding – time and frequency domain approaches

- State Tree & Trellis diagrams

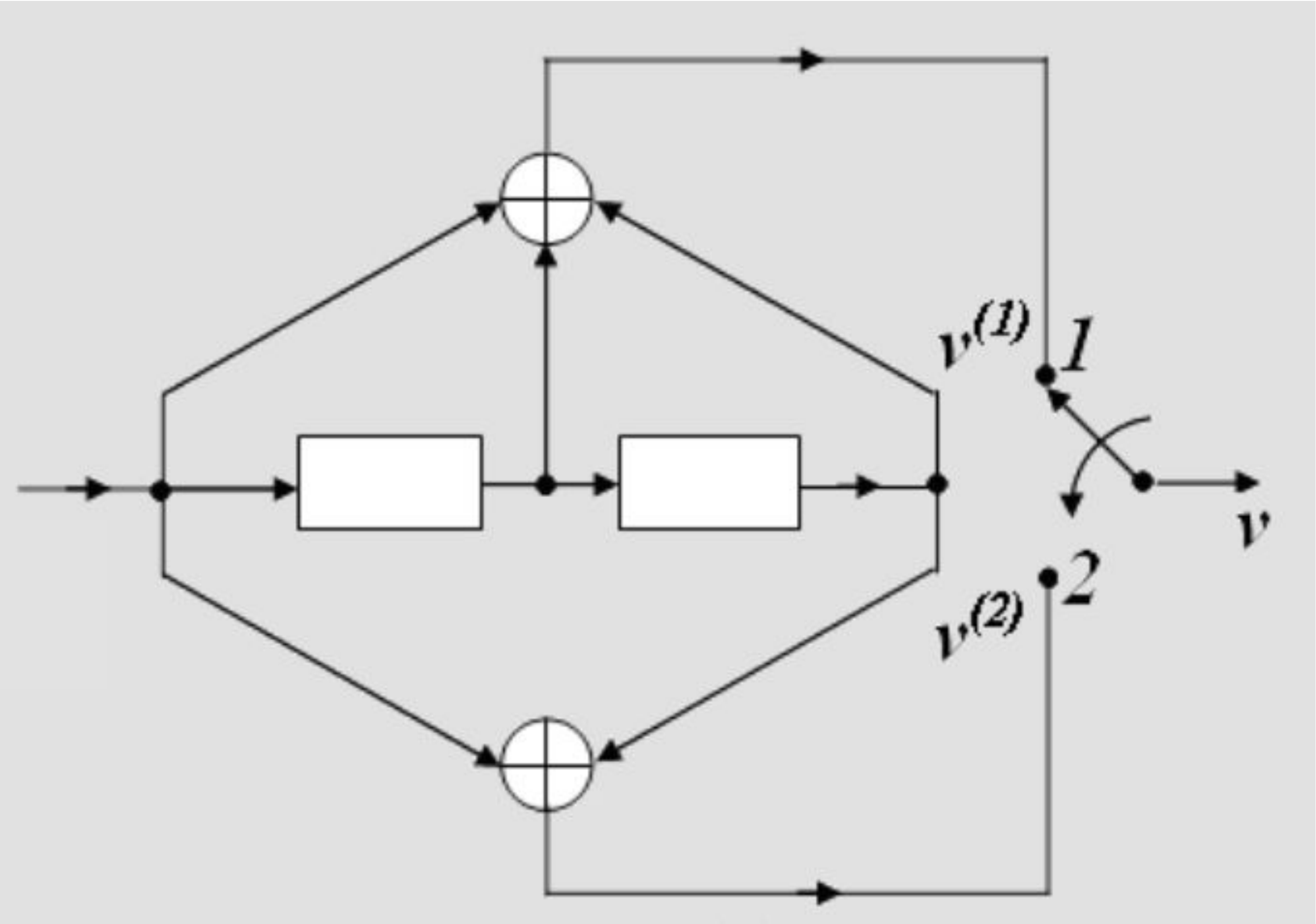
- Transfer function and minimum free distance

- Maximum likelihood decoding of convolutional codes
  - The Viterbi Algorithm.
  - Sequential decoding.

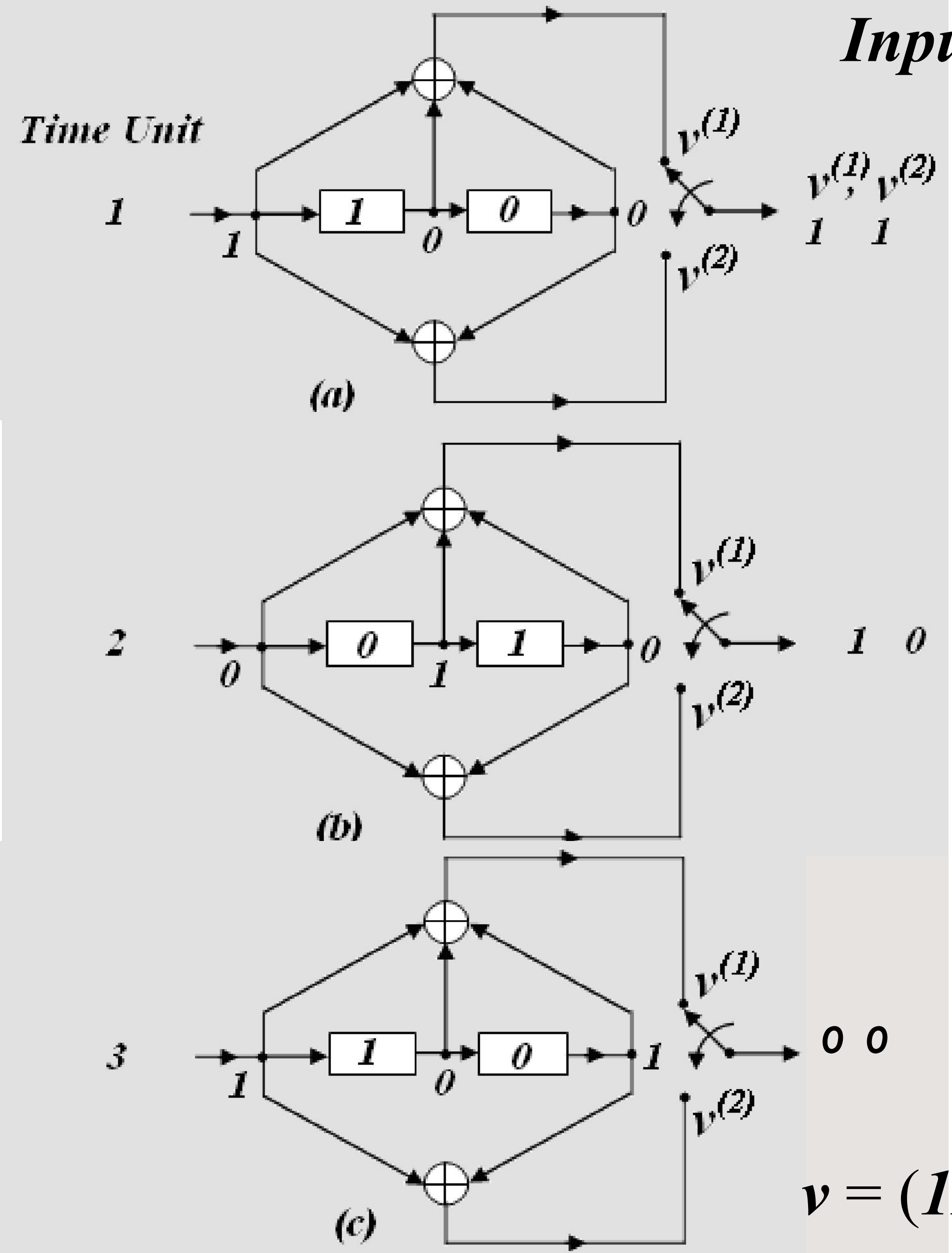
Convolutional Codes

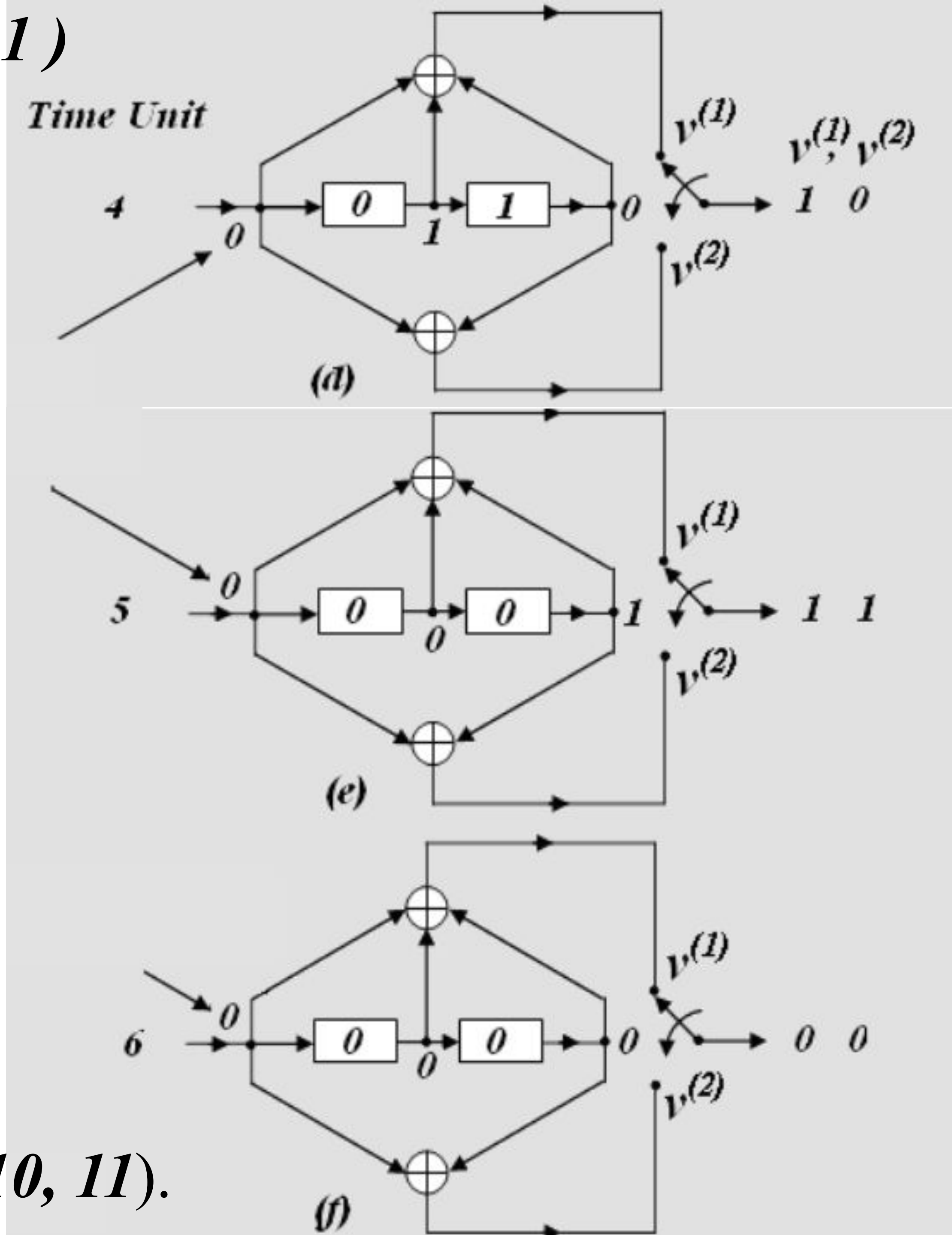
- In block codes the “n” bit code word generated at a particular time instant depends only on the “k” bit message input at that time.
- Block codes - generated by combinatorial logic circuits.
- In convolutional codes, the encoder contains memory.
- The block of ***n***-digits generated by the encoder in a time unit depends on not only on the block of ***k***-data digits with in that time unit, but also on the preceding ‘***m***’ input blocks.
- Usually denoted by (n, k, m) convolutional code. (k-input, n-output, m- memory)
- Rate $= \boldsymbol{\frac{k}{n}}$ and memory m.
- The encoder will have “n” no. of adders & “m” no. of memory registers.
- Obtained by convolution of message and the impulse response of the encoder.

Example of a (2,1,2) convolution encoder



Input u=(1, 0, 1)





$v = (11, 10, 00, 10, 11).$

Important Points to remember …

- The final output is obtained after ***(L + m)*** time units where L is the number of input bits.
- Left most symbols represent earliest transmission.
- Convolution codes suffers from the '*problem of choosing connections*' to yield good distance properties.
- they require ***m***-zeros to be appended to the end of the input sequence for the purpose of '***clearing***' or '***flushing***' or '***re-setting***' of the encoding shift registers off the data bits.
- These added zeros carry no information but have the effect of reducing the code rate below (***k***/***n***).

- The encoding procedure as depicted pictorially earlier, is rather tedious.
- We can approach the encoder in terms of “**Impulse response**” or “**generator sequence**” which merely represents the response of the encoder to a single ‘*1*’ bit that moves through it.

Time Domain approach -Theory

- The "information sequence' ***u* = (*u1*, *u2*, *u3* …….)** enters the encoder one bit at a time starting from ***u1***.
- As the name implies, a convolutional encoder operates by performing convolutions on the information sequence. Specifically, the encoder output sequences, in this case

***v(1)* = {*v1(1)*, *v2(1)*, *v3(1)*… }** and ***v(2)* = {*v1(2)*, *v2(2)*, *v3(2)* … }** are obtained by the discrete convolution of the information sequence with the encoder "impulse responses'.

- The impulse responses are obtained by determining the output sequences of the encoder produced by the input sequence ***u* = (*1*, *0*, *0*, *0*…)**.
- The impulse responses so defined are called ***'generator sequences***' of the code. Since the encoder has a ***m***-time unit memory the impulse responses can last at most **(*m*+ *1*)** time units (That is a total of **(*m*+ *1*)** shifts are necessary for a message bit to enter the shift register and finally come out) and are written as: ***g (i)* = {*g1(i)*, *g2 (i)*, *g3(i)* … *gm+1(i)*}.**

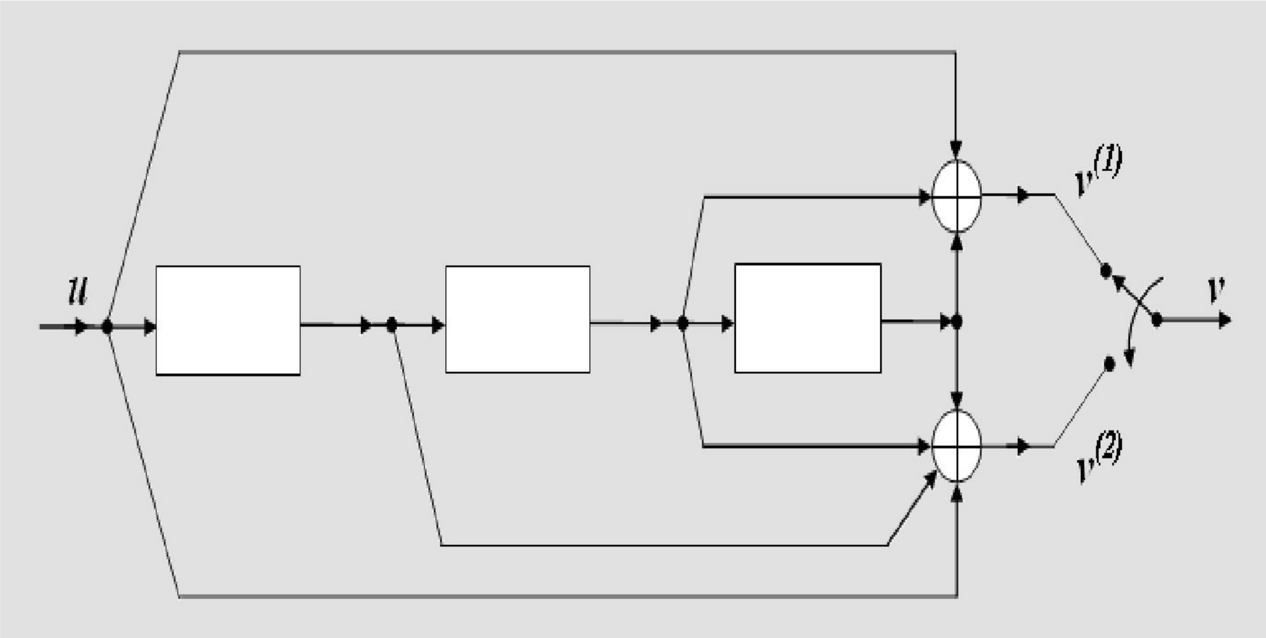
Convolutional Encoding-Time domain approach

- All convolution encoders can be implemented using a "***linear feed forward shift register circuit***".
- Consider an R=1/2 convolutional encoder with memory order m=3
- The impulse response

  is given by

$$g^{(i)} = (g_1^{(i)}, g_2^{(i)}, \dots\dots, g_{m+1}^{(i)})$$

$$g^{(1)} = (1011) \text{ and } g^{(2)} = (1111)$$



- $g^{(1)}$ and $g^{(2)}$ are called **generator sequences** of the encoder

- The encoding equations are given as

  $\boldsymbol{V}^{(1)} = \boldsymbol{u} * \boldsymbol{g}^{(1)}$ and $\boldsymbol{V}^{(2)} = \boldsymbol{u} * \boldsymbol{g}^{(2)}$

- If U=(10111)

  $\boldsymbol{V}^{(1)} = (\mathbf{10111}) * (\mathbf{1011}) = (\mathbf{10000001})$

  $\boldsymbol{V}^{(2)} = (\mathbf{10111}) * (\mathbf{1111}) = (\mathbf{11011101})$

The 2 code words are interlaced and is given by

V=(11,01,00,01,01,01,00,11)

- If U has length "L", then the length of V is n× $(\boldsymbol{m} + \boldsymbol{L}) = \mathbf{2}(\mathbf{3} + \mathbf{5}) = \mathbf{16}$

Generator matrix for Convolutional Encoder

- If the information sequence ***u*** has a finite length, say ***L***, then ***G*** has ***L*** rows and ***v*** has ***n*** x ***(m +L)*** columns (or ***(m +L)*** branch word columns).
- Each branch word is of length '***n***'.
- Thus the Generator matrix ***G***, for the encoders of type shown in Fig as:

$$G = \begin{bmatrix} g_1^{(1)} g_1^{(2)} & g_2^{(1)} g_2^{(2)} & g_3^{(1)} g_3^{(2)} & g_4^{(1)} g_4^{(2)} & \cdots & \cdots & \cdots \\ & g_1^{(1)} g_1^{(2)} & g_2^{(1)} g_2^{(2)} & g_3^{(1)} g_3^{(2)} & g_4^{(1)} g_4^{(2)} & \cdots & \cdots \\ & & g_1^{(1)} g_1^{(2)} & g_2^{(1)} g_2^{(2)} & g_3^{(1)} g_3^{(2)} & g_4^{(1)} g_4^{(2)} & \cdots \end{bmatrix}$$

(Blank places are zeros.)

- The encoding equations in Matrix form is:

$$\boxed{v = u \,.G}$$

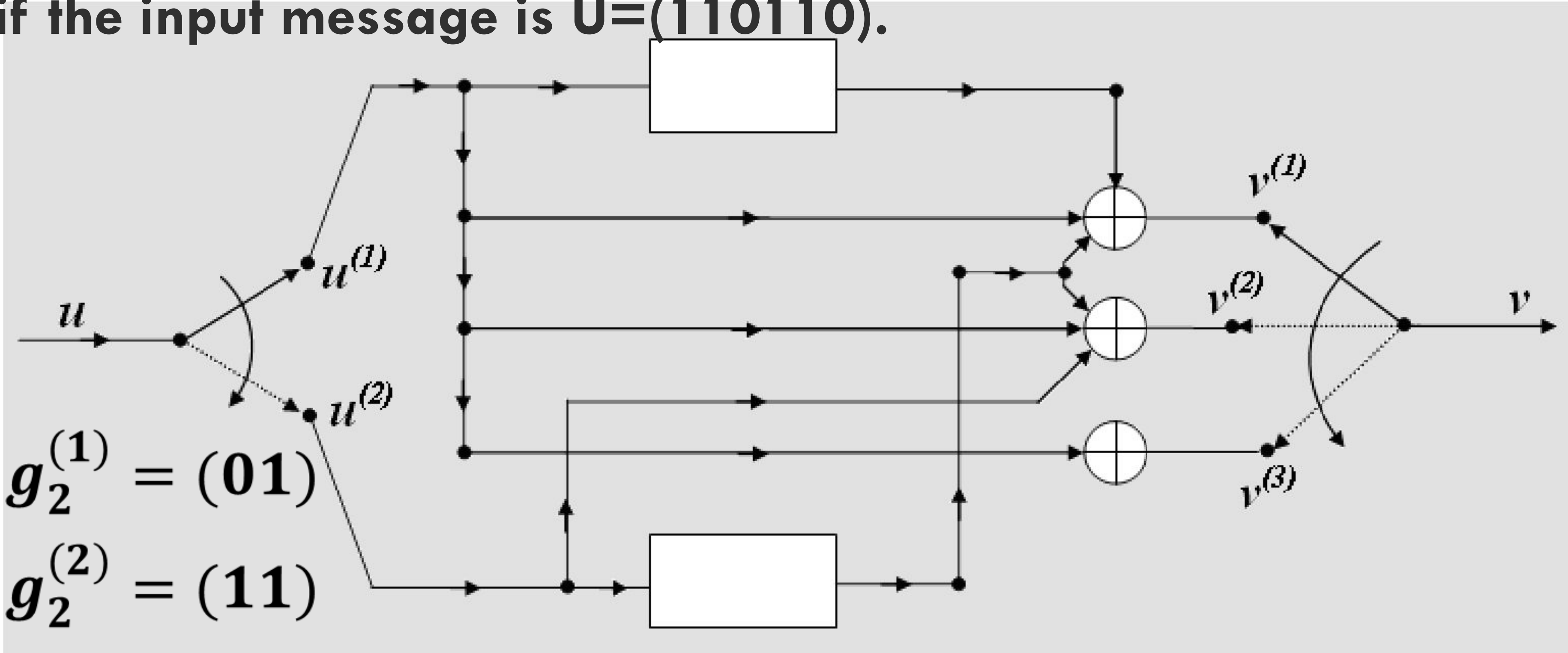
- $\boldsymbol{g}^{(1)} = (\mathbf{1011})$ and $\boldsymbol{g}^{(2)} = (\mathbf{1111})$

- $$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & & & & & & & & \\ & & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & & & & & & \\ & & & & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & & & & \\ & & & & & & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & & \\ & & & & & & & & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$V = U.G = (10111) \quad G = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

V=(11, 01, 00, 01, 01, 01, 00, 11)

Q. A rate 2/3 convolutional encoder with m=1 is given. Obtain the code word obtained if the input message is U=(110110).



Soln:

$$U^{(1)} = (101)$$
$$U^{(2)} = (110)$$

$$g_1^{(1)} = (11) \qquad g_2^{(1)} = (01)$$
$$g_1^{(2)} = (10) \qquad g_2^{(2)} = (11)$$
$$g_1^{(3)} = (10) \qquad g_2^{(3)} = (00)$$

$$V^{(1)} = U^{(1)} * g_1^{(1)} + U^{(2)} * g_2^{(1)}$$
$$V^{(2)} = U^{(1)} * g_1^{(2)} + U^{(2)} * g_2^{(2)}$$
$$V^{(3)} = U^{(1)} * g_1^{(3)} + U^{(2)} * g_2^{(3)}$$

$$V^{(1)} = (101) * (11) + (110) * (01) = (1111) + (0110) = (1001)$$

$$V^{(2)} = (101) * (10) + (110) * (11) = (1010) + (1010) = (0000)$$

$$V^{(3)} = (101) * (10) + (110) * (00) = (1010) + (0000) = (1010)$$

$$V = (101, 000, 001, 100)$$

Length of V=n.(m+L) = 3(1+3)=12

$$G = \begin{bmatrix} 1 & 1 & 1, & 1 & 0 & 0 & & & & & & \\ 0 & 1 & 0, & 1 & 1 & 0 & & & & & & \\ & & & 1 & 1 & 1, & 1 & 0 & 0 & & & \\ & & & 0 & 1 & 0, & 1 & 1 & 0 & & & \\ & & & & & & 1 & 1 & 1, & 1 & 0 & 0 \\ & & & & & & 0 & 1 & 0, & 1 & 1 & 0 \end{bmatrix}$$

Convolutional Encoder-Transform Domain Approach

- Encoding can be done in time domain using convolution as well in transform domain using polynomials

$$v^{(i)}(X) = u(X).\, g^{(i)}(X)$$

Where **u** is the **information polynomial.**

$$\boldsymbol{u = (u_1, u_2, u_3, \dots \dots \dots .)}\ \textbf{and}\ \boldsymbol{u(X) = u_1 + u_2 X + u_3 X^2 + \cdots \dots}$$

And $\boldsymbol{v^{(i)}(X)}$ **is the encoded polynomial**

$$\boldsymbol{v^{(1)}(X) = u(X)\, g^{(1)}(X)} \quad , \quad \boldsymbol{v^{(2)}(X) = u(X)\, g^{(2)}(X)}$$

and $\boldsymbol{g^{(i)}(X)}$ are the generator polynomials of the code.

$$\boldsymbol{g^{(1)}(X)} = g_1{}^{(1)} + g_2{}^{(1)}X + g_3{}^{(1)}X^2 + \ \dots \dots$$

$$\boldsymbol{g^{(2)}(X)} = g_1{}^{(2)} + g_2{}^{(2)}X + g_3{}^{(2)}X^2 + \ \dots \dots$$

- The final codeword can be expressed as:

$$v(X) = v^{(1)}(X^n) + Xv^{(2)}(X^n) + X^2v^{(3)}(X^n) + \ \dots \dots$$

Q) Find the output of convolutional encoder with memory order m=3 using transform domain approach. U=(10111) (previous question)

Soln:

n=2, k=1,m=3

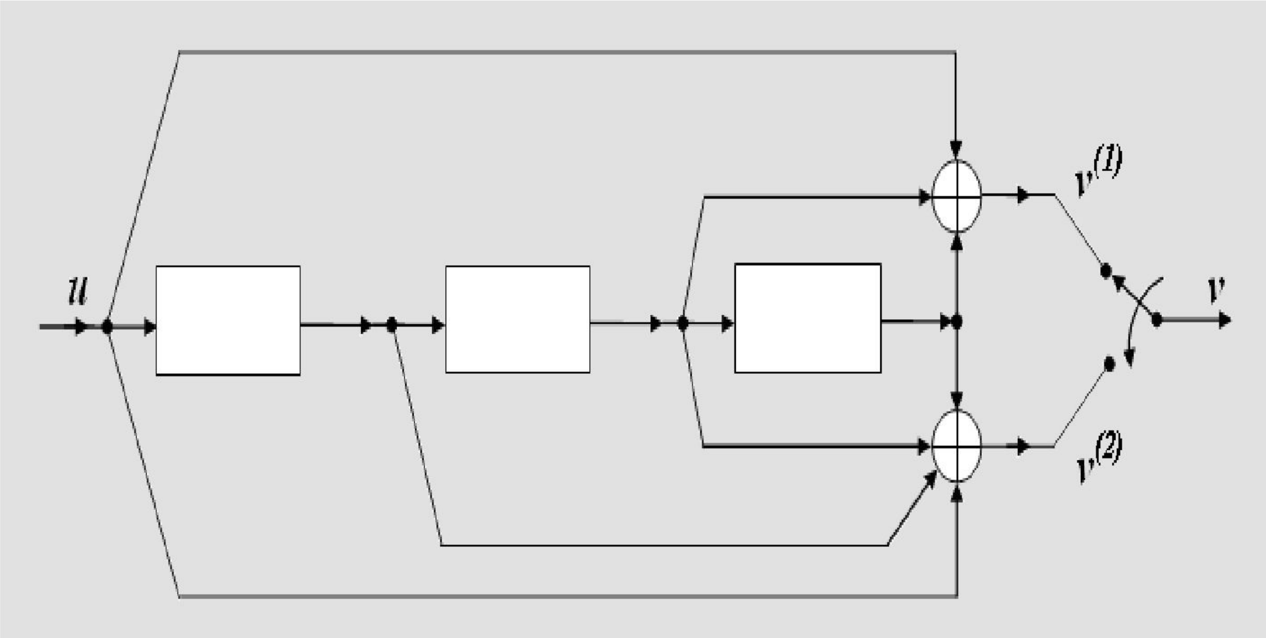
$$g^{(1)} = (1011)$$

$$g^{(2)} = (1111)$$

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

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

$$U(X) = 1 + X^2 + X^3 + X^4$$



$$v^{(1)}(X) = u(X).g^{(1)}(X)$$

$$v^{(2)}(X) = u(X).g^{(2)}(X)$$

$$v^{(1)}(X) = (1 + X^2 + X^3 + X^4).(1 + X^2 + X^3) = 1 + X^7$$

$$v^{(2)}(X) = (1 + X^2 + X^3 + X^4).(1 + X + X^2 + X^3)$$

$$= 1 + X + X^3 + X^4 + X^5 + X^7$$

$$v(X) = [1 + X^7 \qquad 1 + X + X^3 + X^4 + X^5 + X^7]$$

$$v(X) = v^{(1)}(X^2) + X.v^{(2)}(X^2)$$

$$= 1 + X + X^3 + X^7 + X^9 + X^{11} + X^{14} + X^{15}$$

The transfer domain Generator matrix is given by

$$G(X) = [g^{(1)}(X) \quad g^{(2)}(X)] = [1 + X^2 + X^3 \qquad 1 + X + X^2 + X^3]$$

$$v(X) = u(X).G(X)$$

$$= (1 + X^2 + X^3 + X^4).[1 + X^2 + X^3 \qquad 1 + X + X^2 + X^3]$$

$$v(X) = [1 + X^7 \qquad 1 + X + X^3 + X^4 + X^5 + X^7]$$

- According to the formula,
- $v(X) = v^{(1)}(X^2) + Xv^{(2)}(X^2)$
- $= (1 + X^{14}) + (\mathrm{X} + X^3 + X^7 + X^9 + X^{11} + X^{15})$

$$= 1 + \mathrm{X} + X^3 + X^7 + X^9 + X^{11} + X^{14} + X^{15}$$

$$v = (11, 01, 00, 01, 01, 01, 00, 11)$$

Q) What is the constraint length of this encoder?

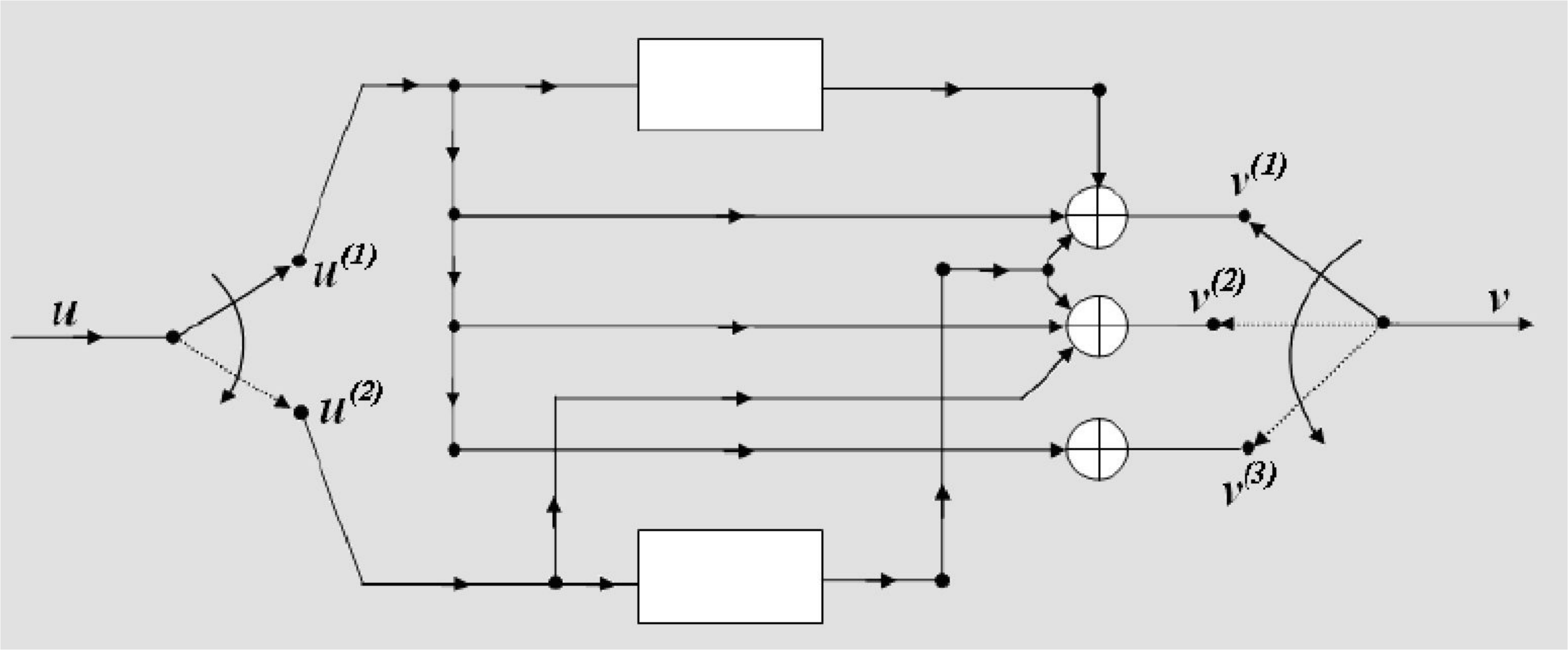
And) Constraint length of an encoder is defined as "***the maximum number of encoder outputs that can be affected by a single information bit***" .

Or

Constraint length is defined as the number of shifts over which a single message bit can influence the encoder output.

$$n_A = n(l + m) = 2(3+1)=8$$

Q) Find the output of the (3, ***2***, ***1***) convolutional encoder for the input U=(1 1 0 1 1 0) as shown in the figure using Transform Domain approach.



- $u^{(1)} = (1\ 0\ 1) \qquad u^{(2)} = (1\ 1\ 0)$

$g_1{}^{(1)}(\mathrm{X}) = 1 + \mathrm{X} \qquad g_1{}^{(2)}(\mathrm{X}) = 1 \qquad g_1{}^{(3)}(\mathrm{X}) = 1$

$g_2{}^{(1)}(\mathrm{X}) = \mathrm{X} \qquad g_2{}^{(2)}(\mathrm{X}) = 1 + \mathrm{X} \qquad g_2{}^{(3)}(\mathrm{X}) = 0$

$$G(X) = \begin{bmatrix} 1+X & 1 & 1 \\ X & 1+X & 0 \end{bmatrix}$$

$$V(\mathrm{X}) = \mathrm{u(X).\,G(X)}$$

$$= \begin{bmatrix} 1+X^2 & 1+X \end{bmatrix} . \begin{bmatrix} 1+X & 1 & 1 \\ X & 1+X & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1+X^3 & 0 & 1+X^2 \end{bmatrix}$$

$\boldsymbol{V^{(1)}(X)}$ $\boldsymbol{V^{(2)}(X)}$ $\boldsymbol{V^{(3)}(X)}$

$$V(X) = \boldsymbol{V^{(1)}}(\boldsymbol{X^3}) + \boldsymbol{XV^{(2)}}(\boldsymbol{X^3}) + \boldsymbol{X^2V^{(3)}}(\boldsymbol{X^3})$$

$$= 1 + X^9 + X^2 + X^8 = 1 + X^2 + X^8 + X^9$$

$$= (1\ 0\ 1, 0\ 0\ 0, 0\ 0\ 1, 1\ 0\ 0)$$

Q) Obtain the generator matrix of a rate ½ systematic convolutional code and draw the encoder circuit. Also obtain the code word if U=(1011). Given $\boldsymbol{g}^{(1)} = (\mathbf{1000})$ $\boldsymbol{g}^{(2)} = (\mathbf{1101})$

Soln:

n=2, k=1, m=3

$$\boldsymbol{U}(\boldsymbol{X}) = \mathbf{1} + \boldsymbol{X}^{2} + \boldsymbol{X}^{3}$$

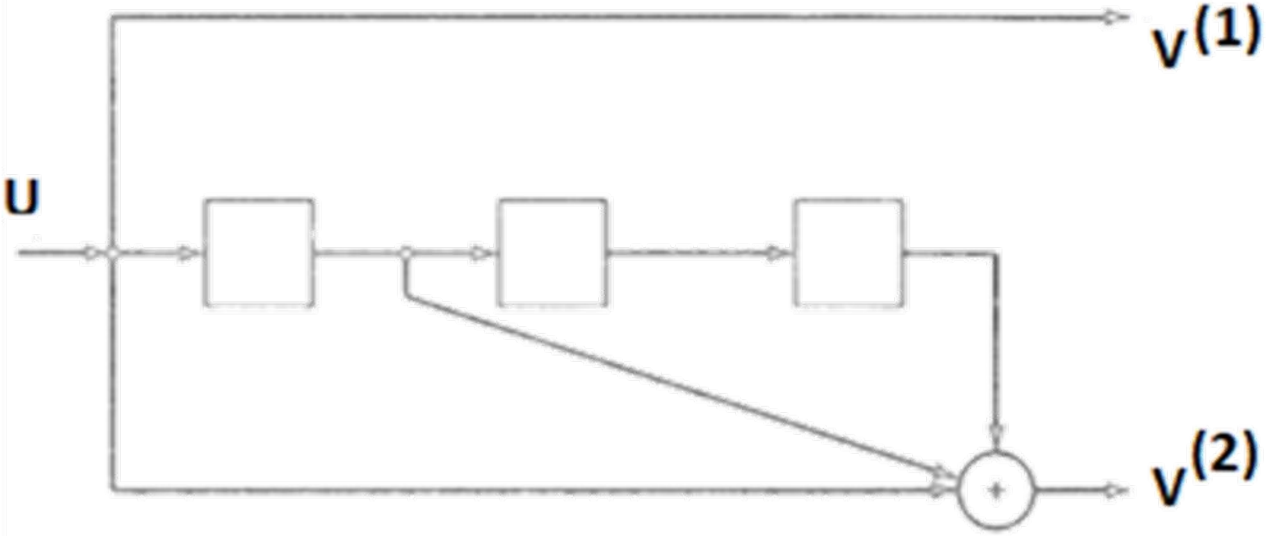
$$\boldsymbol{g}^{(1)}(\boldsymbol{X}) = \mathbf{1}$$

$$\boldsymbol{g}^{(2)}(\boldsymbol{X}) = \mathbf{1} + \boldsymbol{X} + \boldsymbol{X}^{3}$$

$$\boldsymbol{G}(\boldsymbol{X}) = [\boldsymbol{g}^{(1)}(\boldsymbol{X}) \quad \boldsymbol{g}^{(2)}(\boldsymbol{X})]$$

$$= [\ \mathbf{1} \qquad \mathbf{1} + \boldsymbol{X} + \boldsymbol{X}^{3}]$$

$$\boldsymbol{v}(\boldsymbol{X}) = \boldsymbol{u}(\boldsymbol{X}).\boldsymbol{G}(\boldsymbol{X})$$



$$v(X) = [1 + X^2 + X^3].[1 \qquad 1 + X + X^3]$$

$$= [1 + X^2 + X^3 \qquad 1 + X + X^2 + X^3 + X^4 + X^5 + X^6]$$

$$v(D) = v^{(1)}(X^2) + X.v^{(2)}(X^2)$$

$$= 1 + X + X^3 + X^4 + X^5 + X^6 + X^7 + X^9 + X^{11} + X^{13}$$

V=(11, 01, 11, 11, 01, 01, 01)

Structural Properties of Convolutional Codes

- Convolution encoder is a sequential circuit Finite State Machine

- Structural properties of convolutional encoder is portrayed in graphical form using any of the three equivalent diagrams:
  1. State Diagram
  2. Code Tree
  3. Trellis Diagram

State Diagram

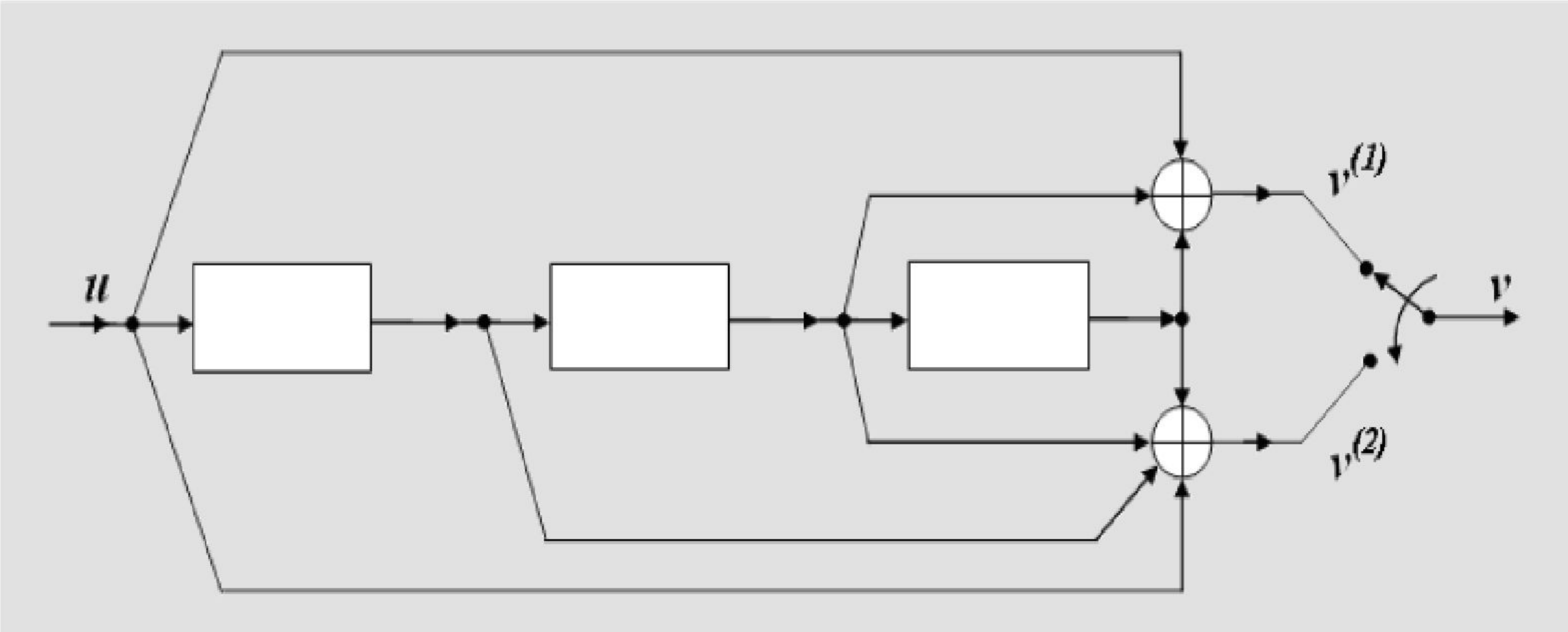
- The state of an encoder is defined as its shift register contents.
- For an ***(n, k, m)*** encoder with k >1the state at time "l" is

σ = $(s_{l-1}, s_{l-2}, \ldots.. s_{l-m})$ =$(u_{l-1}, u_{l-2}, \ldots.. u_{l-m})$

where s_{l-1}represents the contents of the left most delay element.

- 'm' represents the memory order which is defined as the maximum length of any shift register.
- Since there are m memory elements, the no. of states is 2^m
- Each new input block of k bits causes the shift register content to change, ie, there is a transition to a new state.
- Hence there are 2^k branches leaving each state in the state diagram.
- Each branch in the state diagram is labelled with k inputs causing the transition and the n corresponding outputs $(v^{(0)}, v^{(1)}, \ldots \ldots v^{(n-1)})$
- State table is used for constructing the state diagram.

Q) Construct the state table, state transition table and state diagram of the encoder given below.

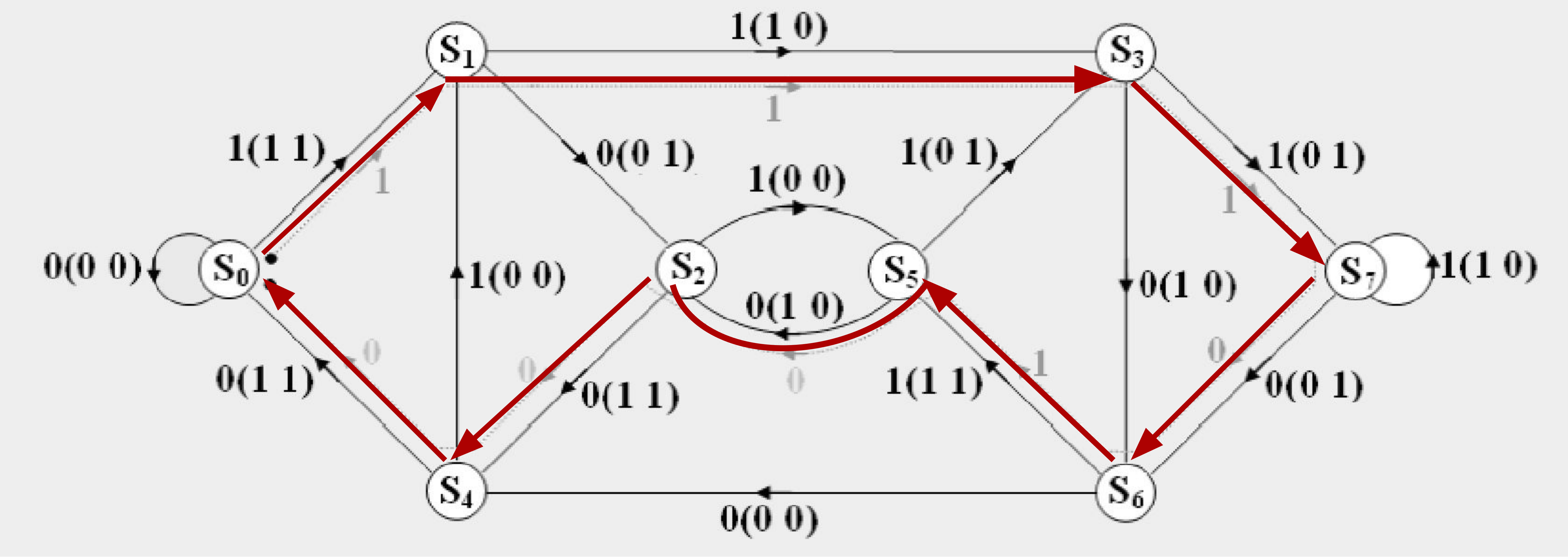


State Table

State	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7
Binary Description	000	100	010	110	001	101	011	111

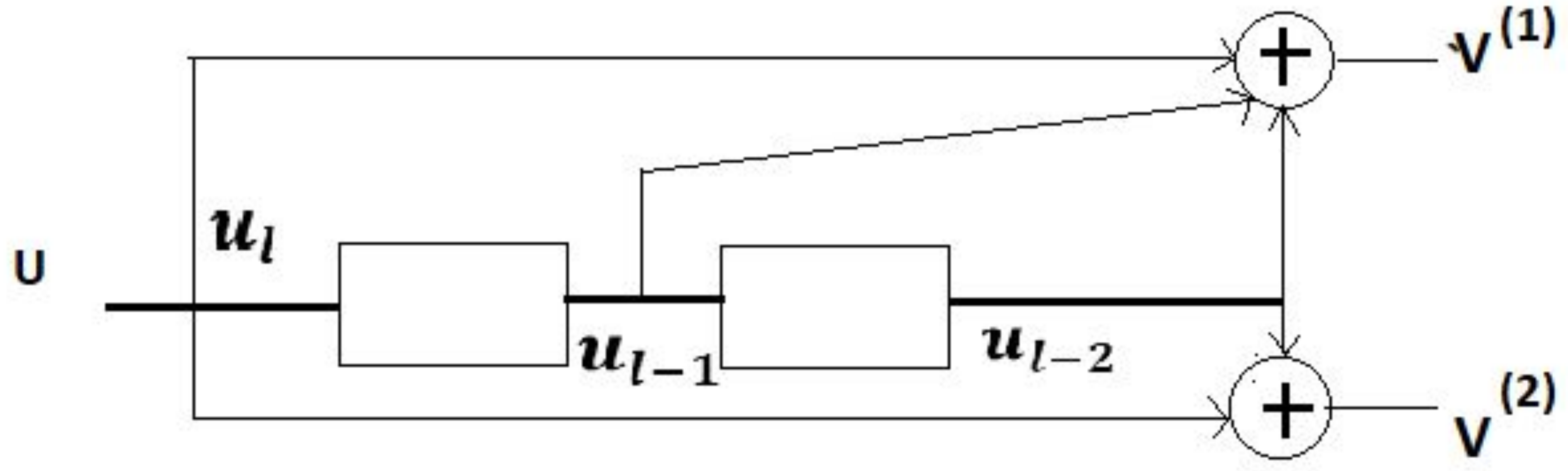
Previous State	Binary Description	Input	Next State	Binary Description	u_l u_{l-1} u_{l-2} u_{l-3}	Output
S_0	0 0 0	0	S_0	0 0 0	0 0 0 0	0 0
		1	S_1	1 0 0	1 0 0 0	1 1
S_1	1 0 0	0	S_2	0 1 0	0 1 0 0	0 1
		1	S_3	1 1 0	1 1 0 0	1 0
S_2	0 1 0	0	S_4	0 0 1	0 0 1 0	1 1
		1	S_5	1 0 1	1 0 1 0	0 0
S_3	1 1 0	0	S_6	0 1 1	0 1 1 0	1 0
		1	S_7	1 1 1	1 1 1 0	0 1
S_4	0 0 1	0	S_0	0 0 0	0 0 0 1	1 1
		1	S_1	1 0 0	1 0 0 1	0 0
S_5	1 0 1	0	S_2	0 1 0	0 1 0 1	1 0
		1	S_3	1 1 0	1 1 0 1	0 1
S_6	0 1 1	0	S_4	0 0 1	0 0 1 1	0 0
		1	S_5	1 0 1	1 0 1 1	1 1
S_7	1 1 1	0	S_6	0 1 1	0 1 1 1	0 1
		1	S_7	1 1 1	1 1 1 1	1 0

State Diagram



If u = (1 1 1 0 1) 0 0 0 11 10 01 01 11 10 11 11

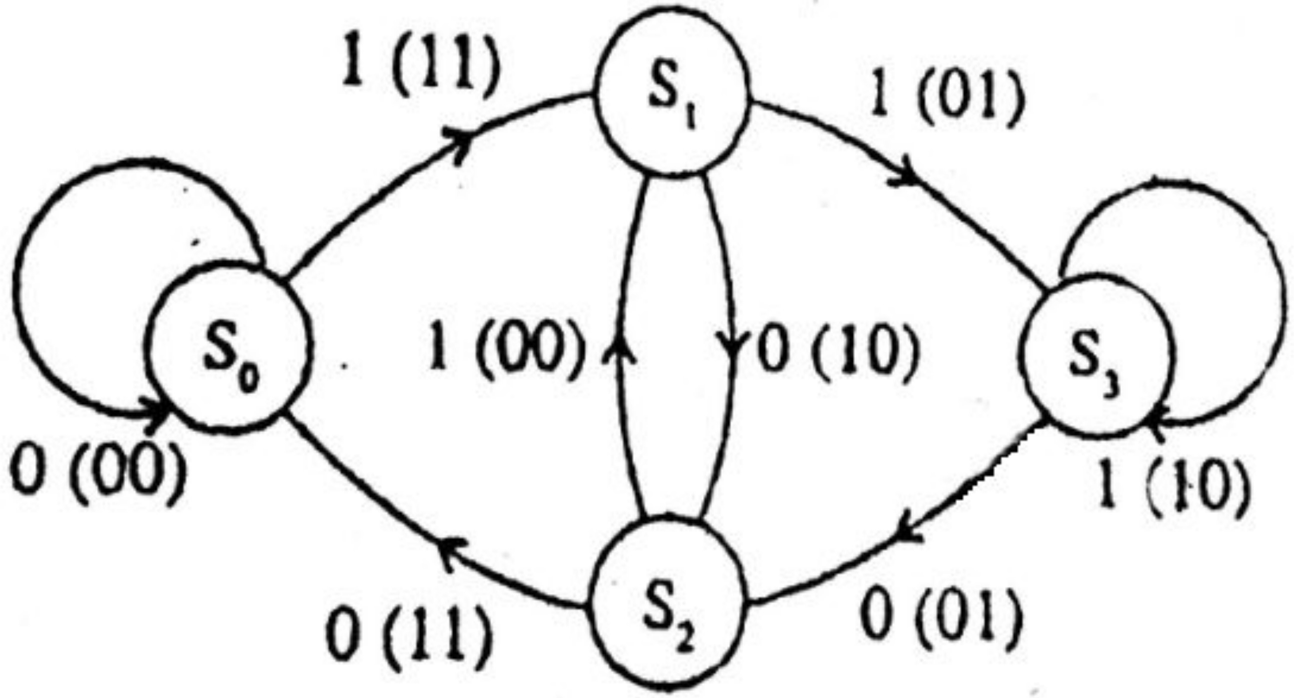
Q. Draw convolutional encoder and its state diagram. Find the convolutional code using state diagram. Given $\boldsymbol{g}^{(1)} = (\mathbf{111})$ $\boldsymbol{g}^{(2)} = (\mathbf{101})$.



State Transition Table

Previous State	Binary Description	Input	Next State	Binary Description		Output
	0 0	0 1		0 0 1 0	0 0 0 1 0 0	0 0 1 1
	1 0	0 1		0 1 1 1	0 1 0 1 1 0	1 0 0 1
	0 1	0 1		0 0 1 0	0 0 1 1 0 1	1 1 0 0
	1 1	0 1		0 1 1 1	0 1 1 1 1 1	0 1 1 0

Consider the input **U=(10111)**



Output= ?

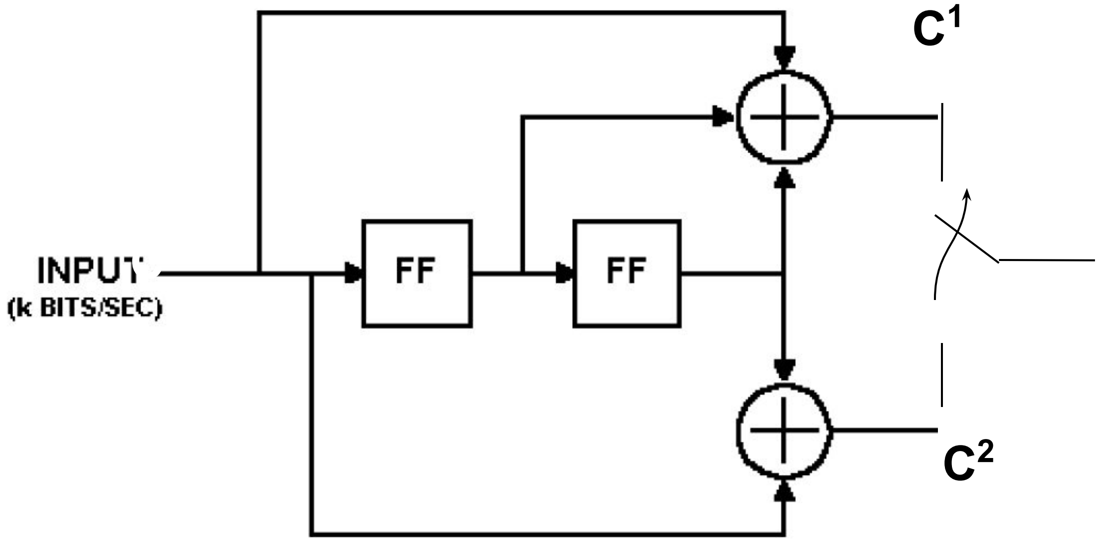
Structural Properties of Convolutional Codes

- Convolution encoder is a sequential circuit Finite State Machine

- Structural properties of convolutional encoder is portrayed in graphical form using any of the three equivalent diagrams:
  1. State Diagram
  2. Code Tree
  3. Trellis Diagram

State Table

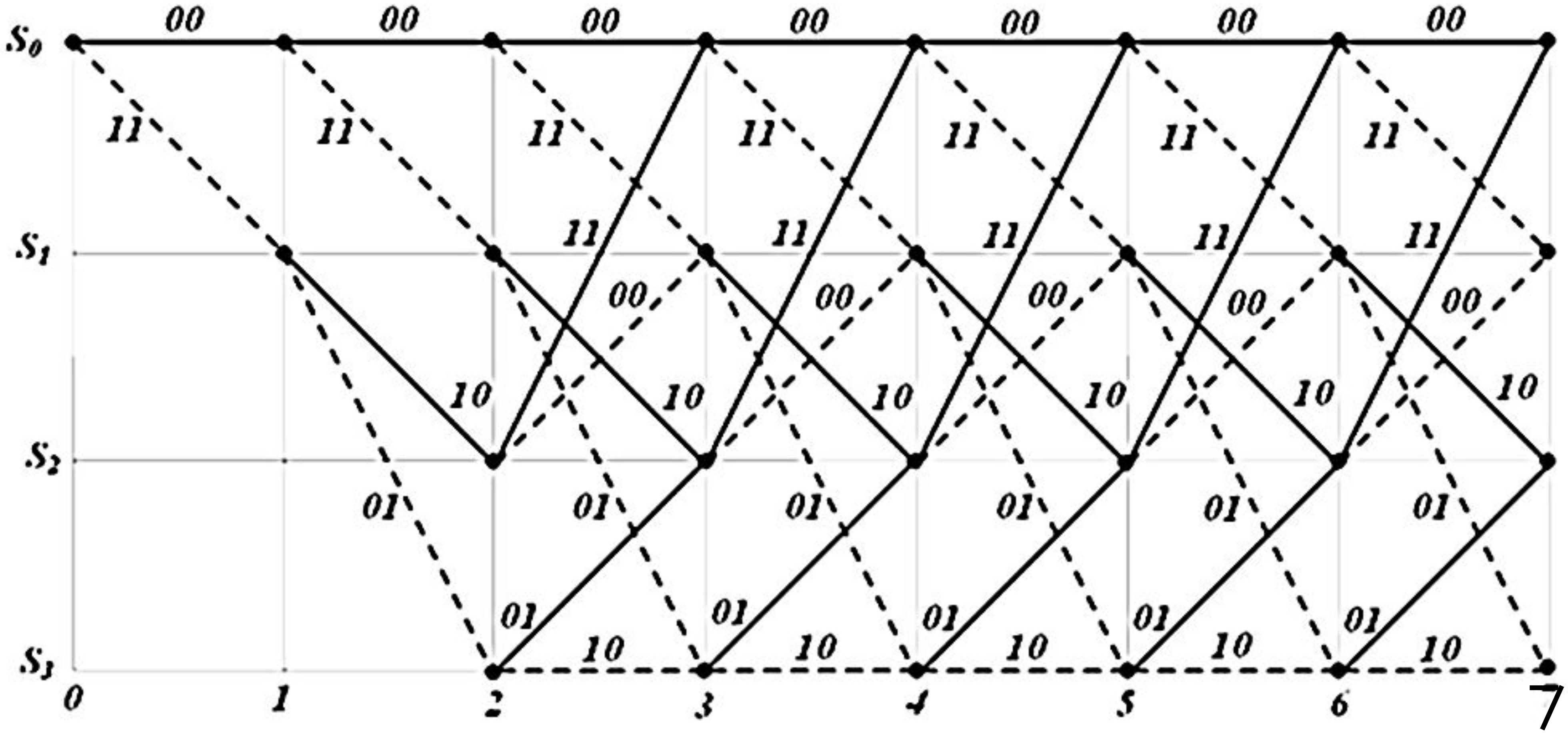
I/P	PS	NS	OUT PUT	
			C^1	C^2
0	00	00	0	0
1	00	10	1	1
0	10	01	1	0
1	10	11	0	1
0	01	00	1	1
1	01	10	0	0
0	11	01	0	1
1	11	11	1	0



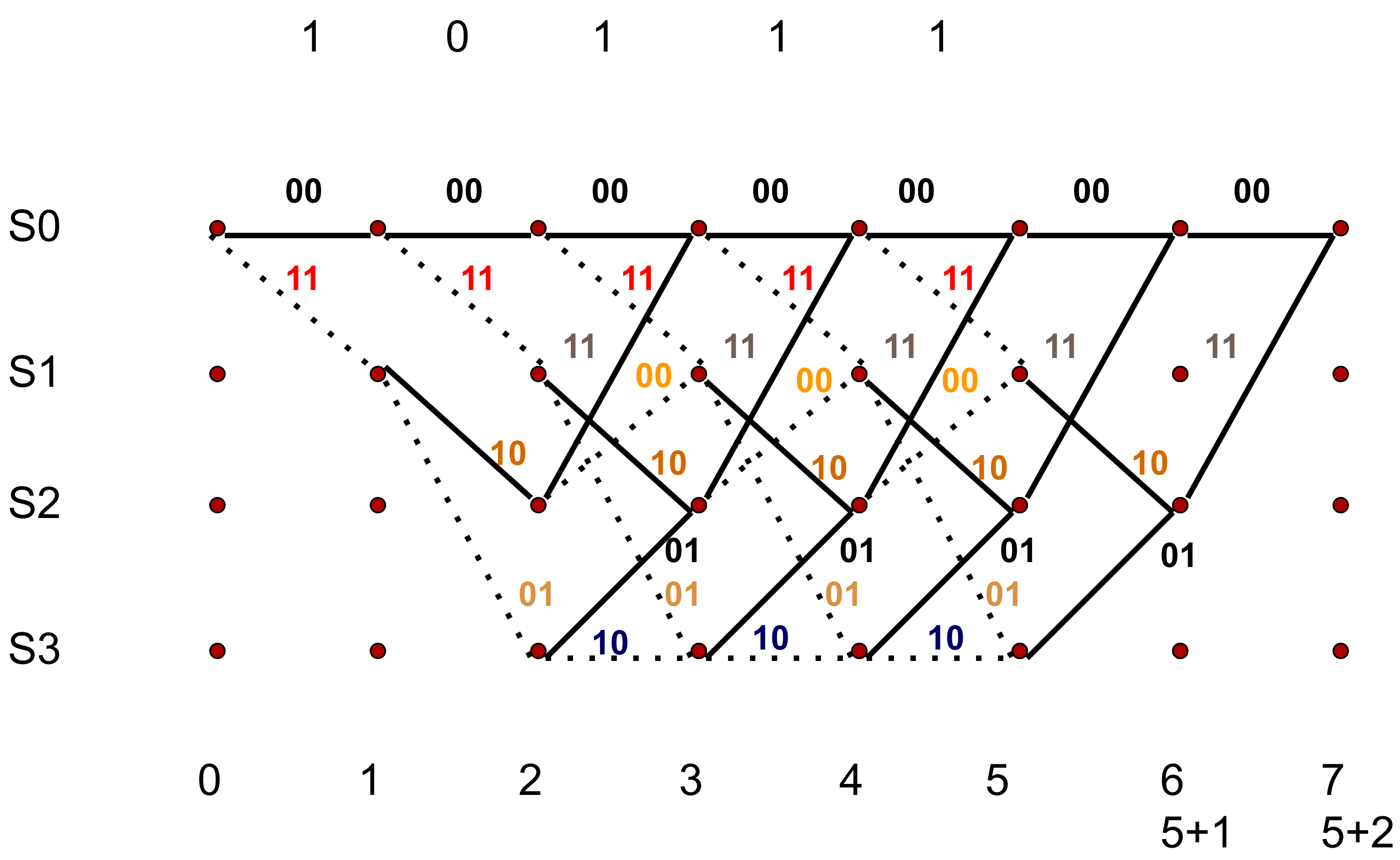
State Transition Table

Previous State	Binary Description	Input	Next State	Binary Description		Output
	0 0	0 1		0 0 1 0	0 0 0 1 0 0	0 0 1 1
	1 0	0 1		0 1 1 1	0 1 0 1 1 0	1 0 0 1
	0 1	0 1		0 0 1 0	0 0 1 1 0 1	1 1 0 0
	1 1	0 1		0 1 1 1	0 1 1 1 1 1	0 1 1 0

Trellis Diagram

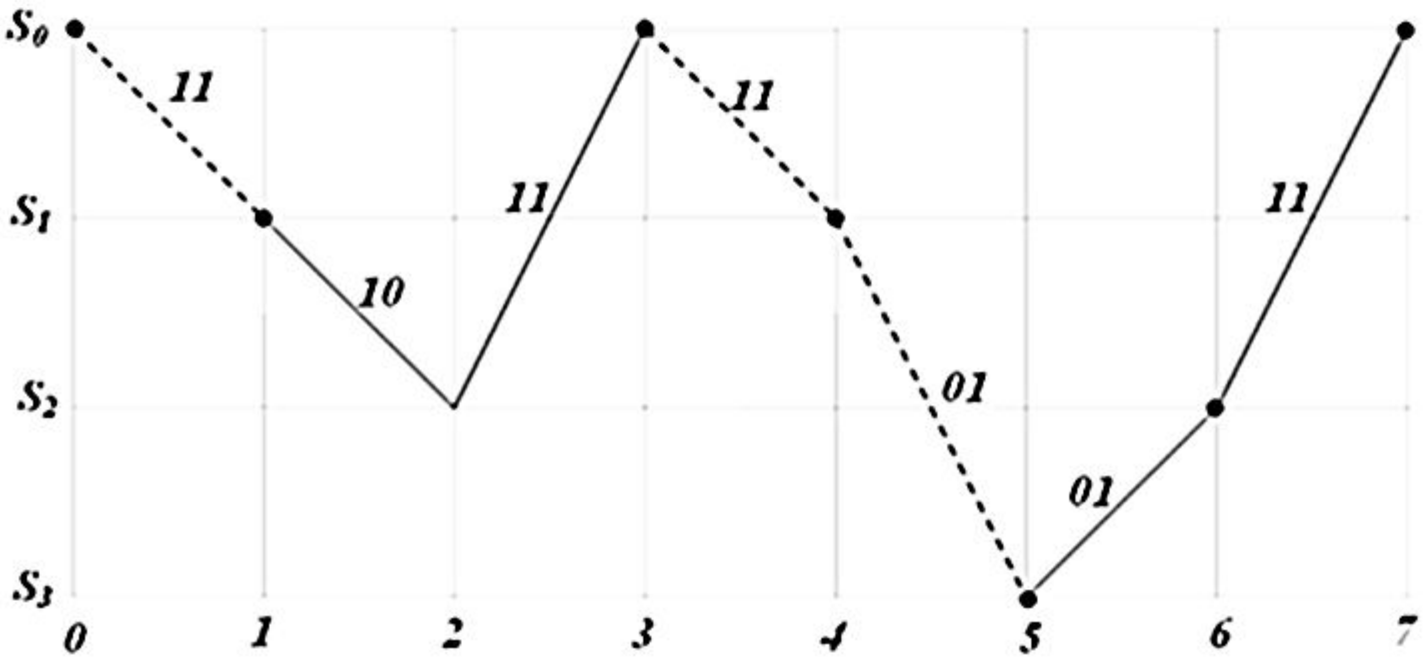


Trellis Diagram (Encoding)

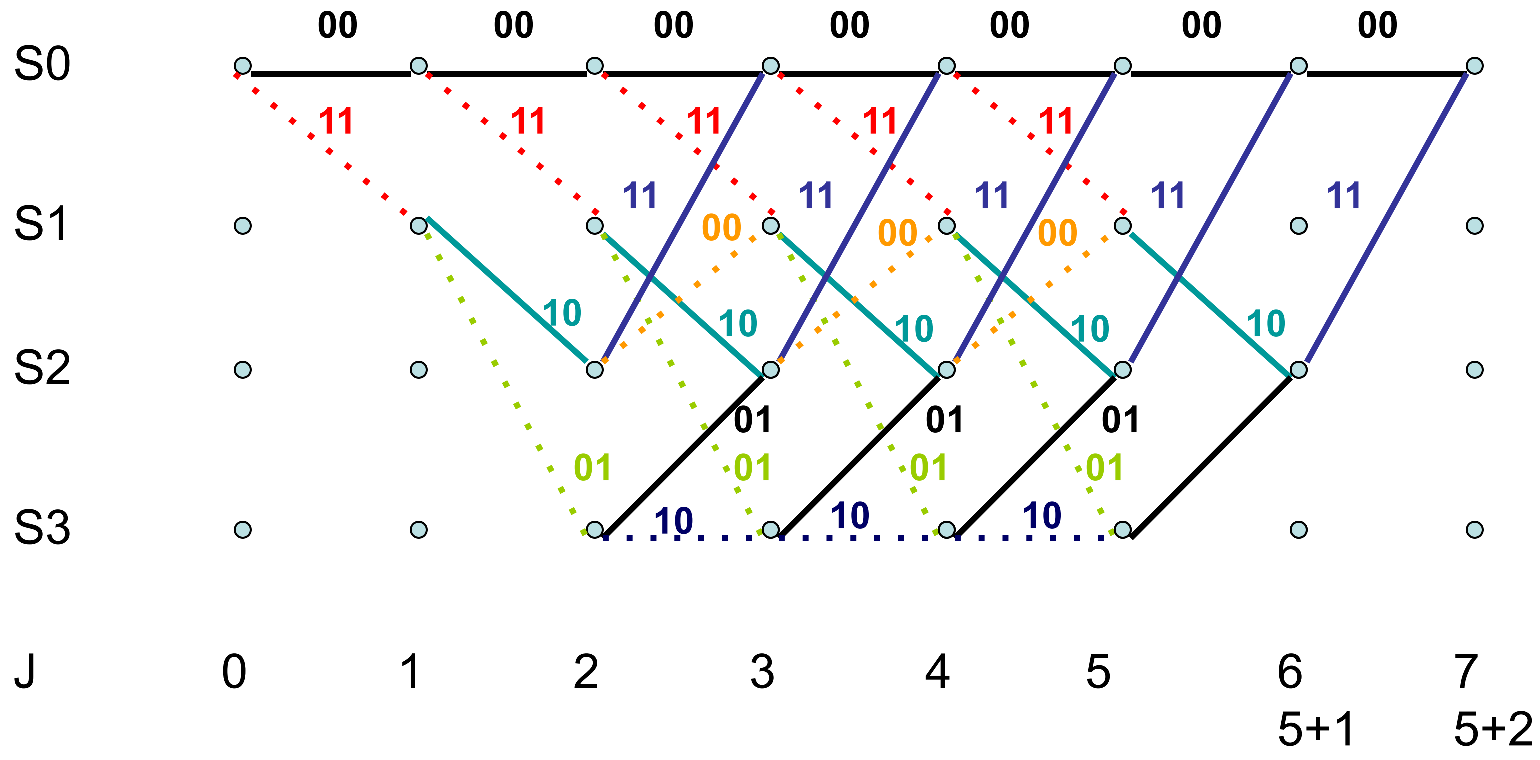


Trellis Diagram

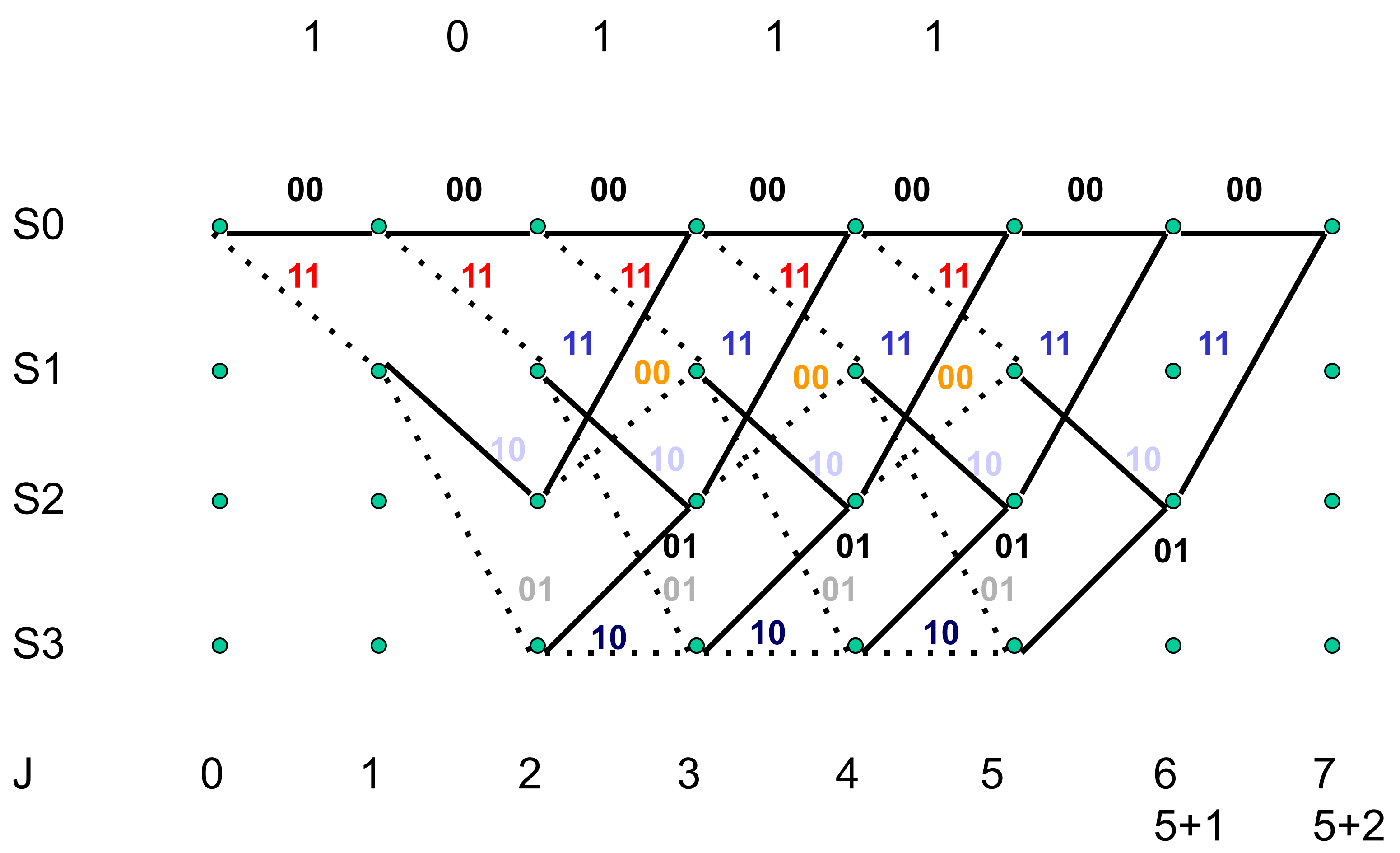
- The reader can readily verify that the encoder output corresponding to the sequence u = (10011) is indeed v = (11, 10, 11, 11, 01, 01, 11) the path followed being as shown in Fig



Trellis Diagram



Trellis Diagram (Encoding)



Viterbi Algorithm

- Assume an (n,k,m) encoder with information sequence length K*=kh. There are 2^k branches leaving and entering each state and 2^{K^*} distinct paths through the trellis corresponding to 2^{K^*} codewords.
- Let u be an information sequence $\boldsymbol{u} = (\boldsymbol{u_0}, \boldsymbol{u_1}, \boldsymbol{u_2}, \dots \dots \dots, \boldsymbol{u_{h-1}})$ of length K*=kh that is encoded into a codeword v= $(\boldsymbol{v_0}, \boldsymbol{v_1}, \boldsymbol{v_2}, \dots \dots \dots, \boldsymbol{v_{h+m-1}})$ of length N=n(h+m)
- Let r $= (\boldsymbol{r_0}, \boldsymbol{r_1}, \boldsymbol{r_2}, \dots \dots \dots, \boldsymbol{r_{h+m-1}})$ be the received codeword over a binary input Discrete memoryless channel (DMC).
- We can alternatively write $\boldsymbol{u} = (\boldsymbol{u_0}, \boldsymbol{u_1}, \boldsymbol{u_2}, \dots \dots \dots, \boldsymbol{u}_{\mathrm{K}*-\boldsymbol{1}})$, v $= (\boldsymbol{v_0}, \boldsymbol{v_1}, \boldsymbol{v_2}, \dots \dots \dots, \boldsymbol{v_{N-1}})$ and r $= (\boldsymbol{r_0}, \boldsymbol{r_1}, \boldsymbol{r_2}, \dots \dots \dots, \boldsymbol{r_{N-1}})$.
- When a decoder receives a sequence, it has to estimate the sequence that was really sent.

- The decoder will be optimum if it choses the estimate that maximizes the log-likelihood function, given by:

$$\log P(r \mid v)$$

- where r is the received bits sequence and v is the code vector applied by the encoder to the input of a discrete memoryless channel. P(r/v) stands for channel transition probability.
- This is called a Maximum Likelihood Detector.
- For a DMC

$$P(\mathbf{r}|\mathbf{v}) = \prod_{l=0}^{h+m-1} P(\mathbf{r}_l|\mathbf{v}_l) = \prod_{l=0}^{N-1} P(r_l|v_l)$$

$\therefore$

$$\log P(\mathbf{r}|\mathbf{v}) = \sum_{l=0}^{h+m-1} \log P(\mathbf{r}_l|\mathbf{v}_l) = \sum_{l=0}^{N-1} \log P(r_l|v_l)$$

- The decoding algorithm uses two metrics: the branch metric (BM) and the path metric (PM).
- The branch metric is a measure of the "distance" between what was transmitted and what was received – log $p(r_l / v_l)$ also denoted as $M(r_l / v_l)$
- The path metric is a value associated with a state in the trellis (i.e., a value associated with each node). It corresponds to the Hamming distance over the most likely path from the initial state to the current state in the trellis - log $p(r/v)$ also denoted as $M(r / v)$
- By "most likely", we mean the path with smallest Hamming distance between the initial state and the current state, measured over all possible paths between the two states. The path with the smallest Hamming distance minimizes the total number of bit errors.

$$M(\mathbf{r}|\mathbf{v}) = \sum_{l=0}^{h+m-1} M(\mathbf{r}_l|\mathbf{v}_l) = \sum_{l=0}^{h+m-1} \log P(\mathbf{r}_l|\mathbf{v}_l) = \sum_{l=0}^{N-1} M(r_l|v_l) = \sum_{l=0}^{N-1} \log P(r_l|v_l).$$

- Path metric for the first ‘t’ branches of a path is given by

$$M([\mathbf{r}|\mathbf{v}]_t) = \sum_{l=0}^{t-1} M(\mathbf{r}_l|\mathbf{v}_l) = \sum_{l=0}^{t-1} \log P(\mathbf{r}_l|\mathbf{v}_l) = \sum_{l=0}^{nt-1} M(r_l|v_l) = \sum_{l=0}^{nt-1} \log P(r_l|v_l).$$

Steps in Viterbi Algorithm

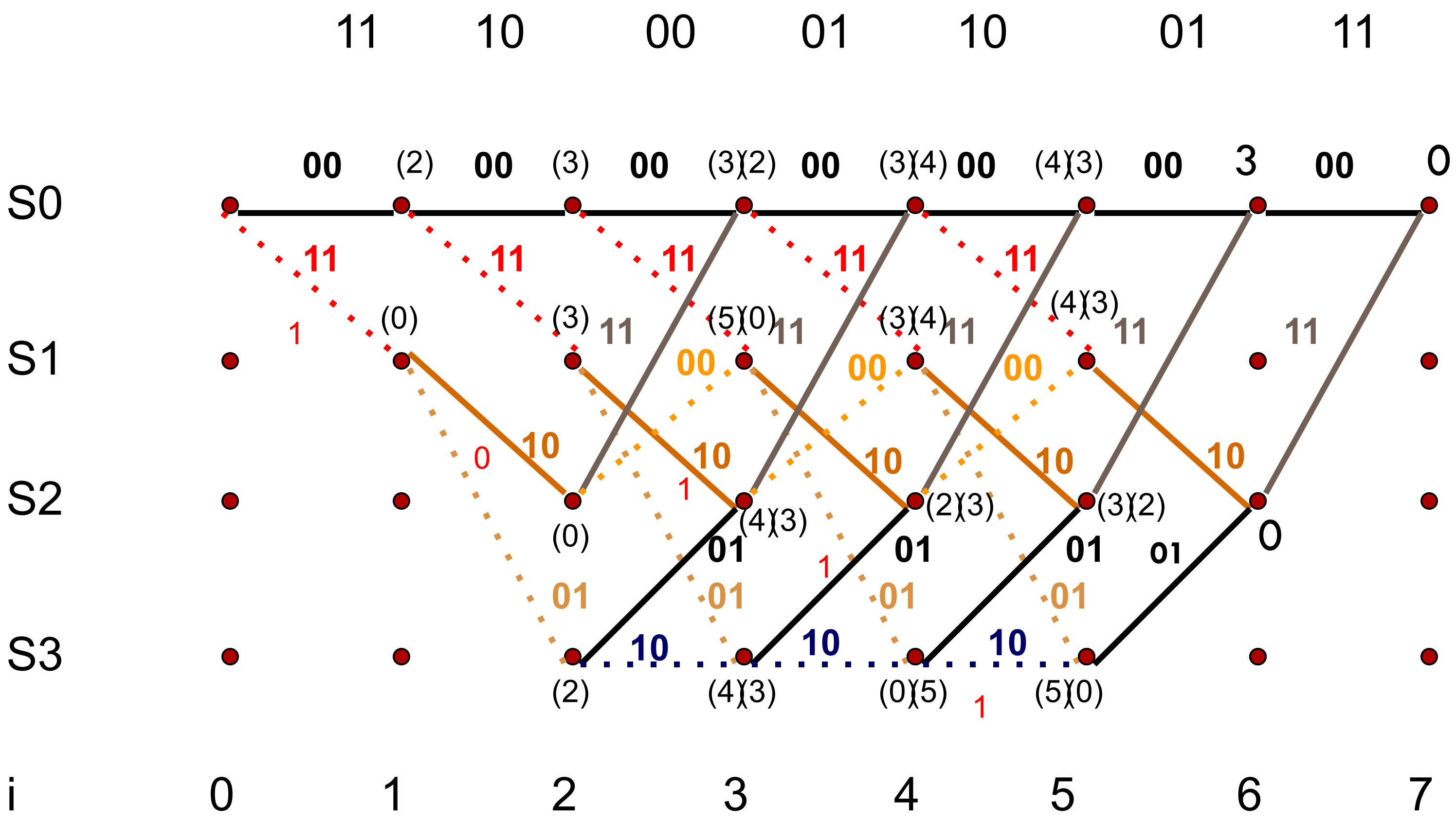
Step 1. Beginning at time unit $t = m$, compute the partial metric for the single path entering each state. Store the path (the survivor) and its metric for each state.

Step 2. Increase t by 1. Compute the partial metric for all 2^k paths entering a state by adding the branch metric entering that state to the metric of the connecting survivor at the previous time unit. For each state, compare the metrics of all 2^k paths entering that state, select the path with the largest metric (the survivor), store it along with its metric, and eliminate all other paths.

Step 3. If $t < h + m$, repeat step 2; otherwise, stop.

Decoding using Viterbi Algorithm

Consider the received code word for a (2,1,2) convolutional encoder to be (11,10,00,01,10,01,11). Check for error , obtain the transmitted code word & obtain the message bits using Viterbi algorithm.



Note: Use presentation mode to see all transitions