Modul #11

TE3223
SISTEM KOMUNIKASI 2

CONVOLUTIONAL CODE

Program Studi S1 Teknik Telekomunikasi
Departemen Teknik Elektro - Sekolah Tinggi Teknologi Telkom
Bandung – 2007

Convolutional codes

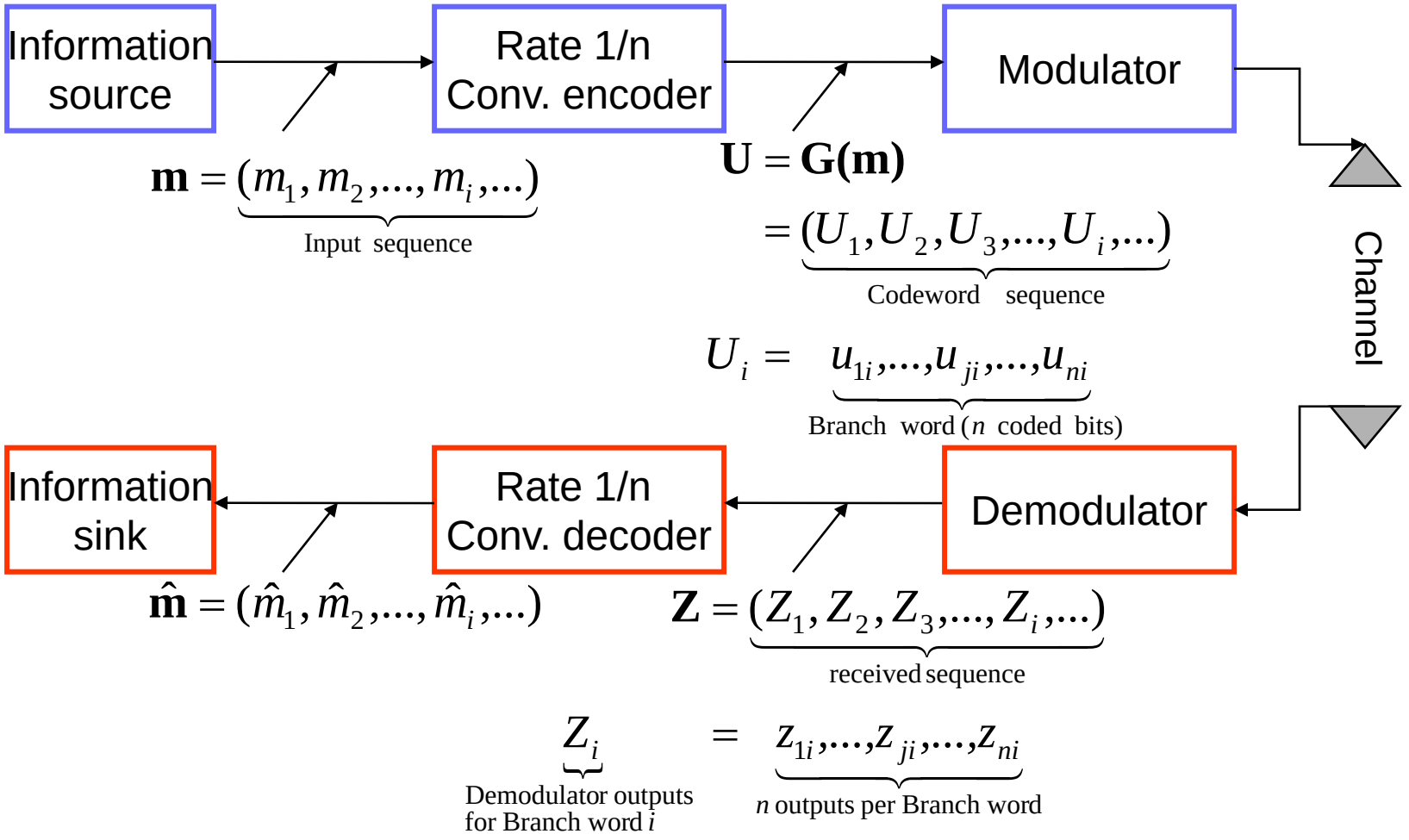
- Pada channel coding menggunakan kode konvolusi :
  - Mengkodekan data stream ke dalam sebuah codeword.
  - Tidak memerlukan pengubahan data stream kedalam suatu blok yang ukurannya tetap.

- Perbedaan mendasar antara kode blok dan kode konvolusi dalam mendesain dan mengevaluasi :
  - Block codes didasarkan pada teknik algebra / kombinasi
  - Convolutional codes didasarkan pada teknik konstruksi

Convolutional codes - cont'd

- Kode konvolusi di spesifikasikan dengan 3 parameter yaitu (n, k, K) atau $(k/n, K)$
  - $R_c = k/n$ adalah coding rate (laju pengkodean), menyatakan jumlah bit data per per coded bit.
    - Pada prakteknya dipilih $k=1$
  - K menyatakan constraint length dari encoder dimana encoder mempunyai K-1 elemen memori (shift register)

Block diagram of the DCS



Proses encoder

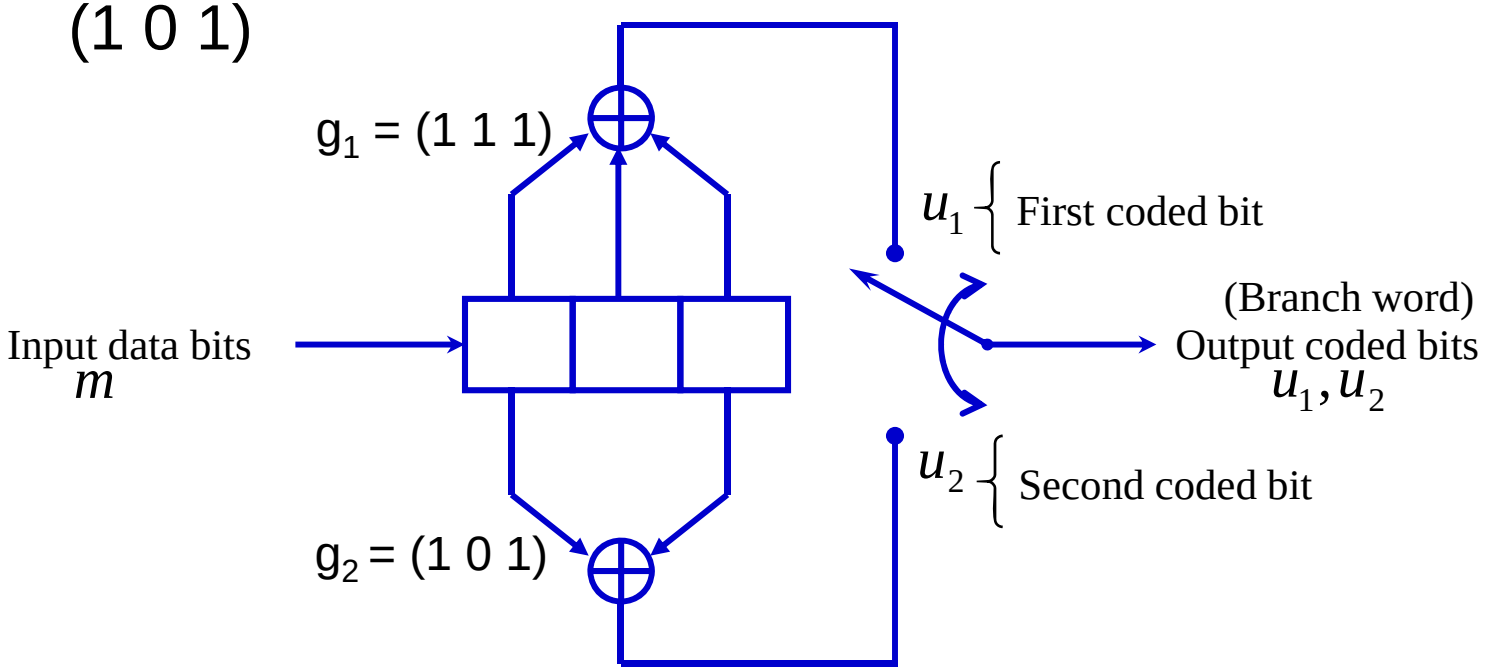
- Inisialisasi memori shift register sebelum proses encoding dengan mengisi semua shift register dengan bit nol (all-zero)
- Tambahkan bit nol diakhir data bit setelah proses encoding agar semua isi shift register menjadi nol. Bit nol yang ditambahkan ini disebut *tail-zero bit* atau bit tail

- Laju pengkodean efektif :
  - L adalah jumlah data bit dan diasumsikan $k=1$:

$$R_{eff} = \frac{L}{n(L+K-1)} < R_c$$

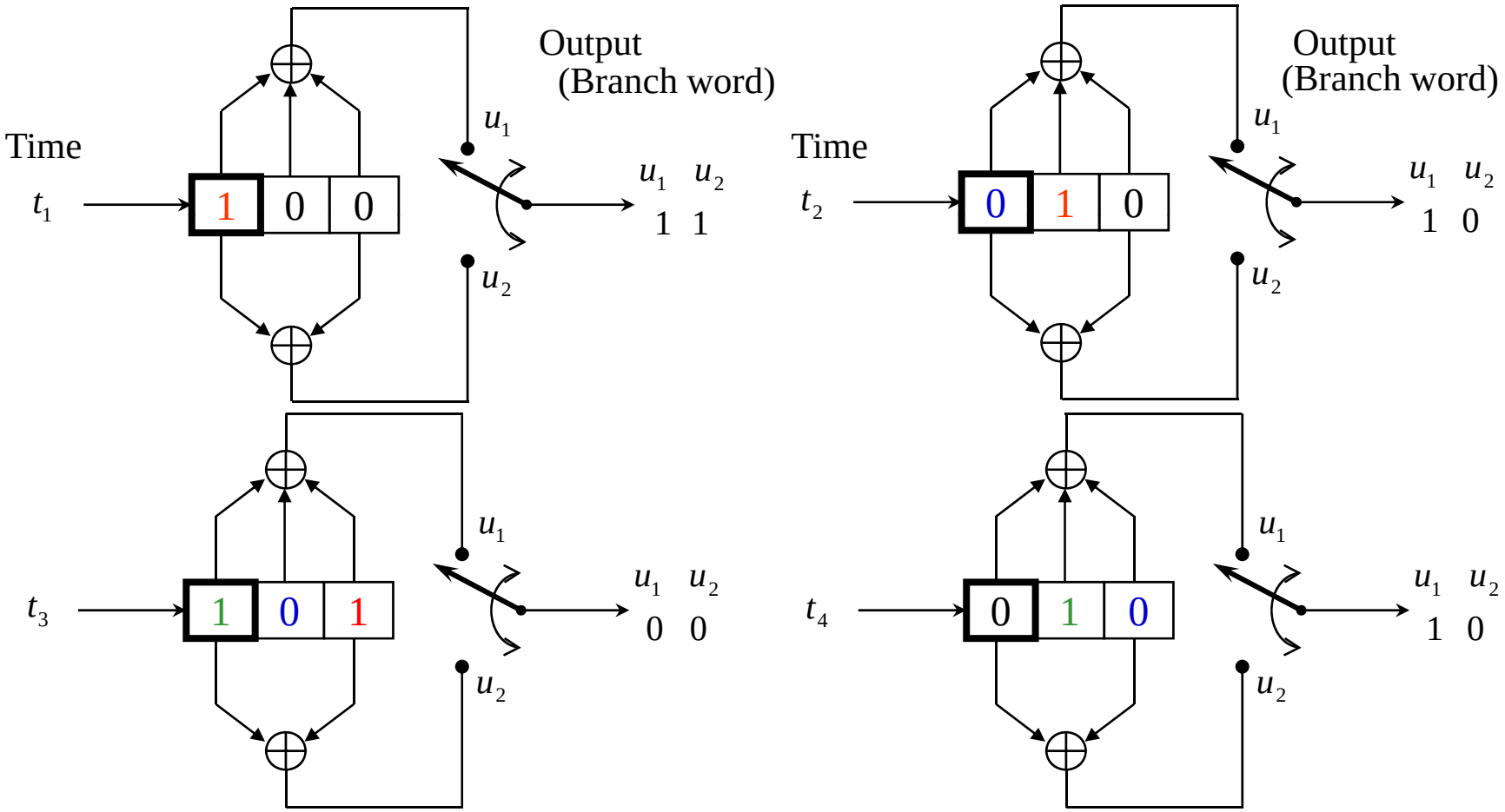
Encoder Konvolusi dengan rate ½

- Convolutional encoder (rate ½, K=3)
  - 3 shift-register dimana isi register yang pertama merupakan data input
  - $k = 1$ dan $n = 2$
  - Contoh Implementasi dengan $g_1 = (1\ 1\ 1)$ dan $g_2 = (1\ 0\ 1)$

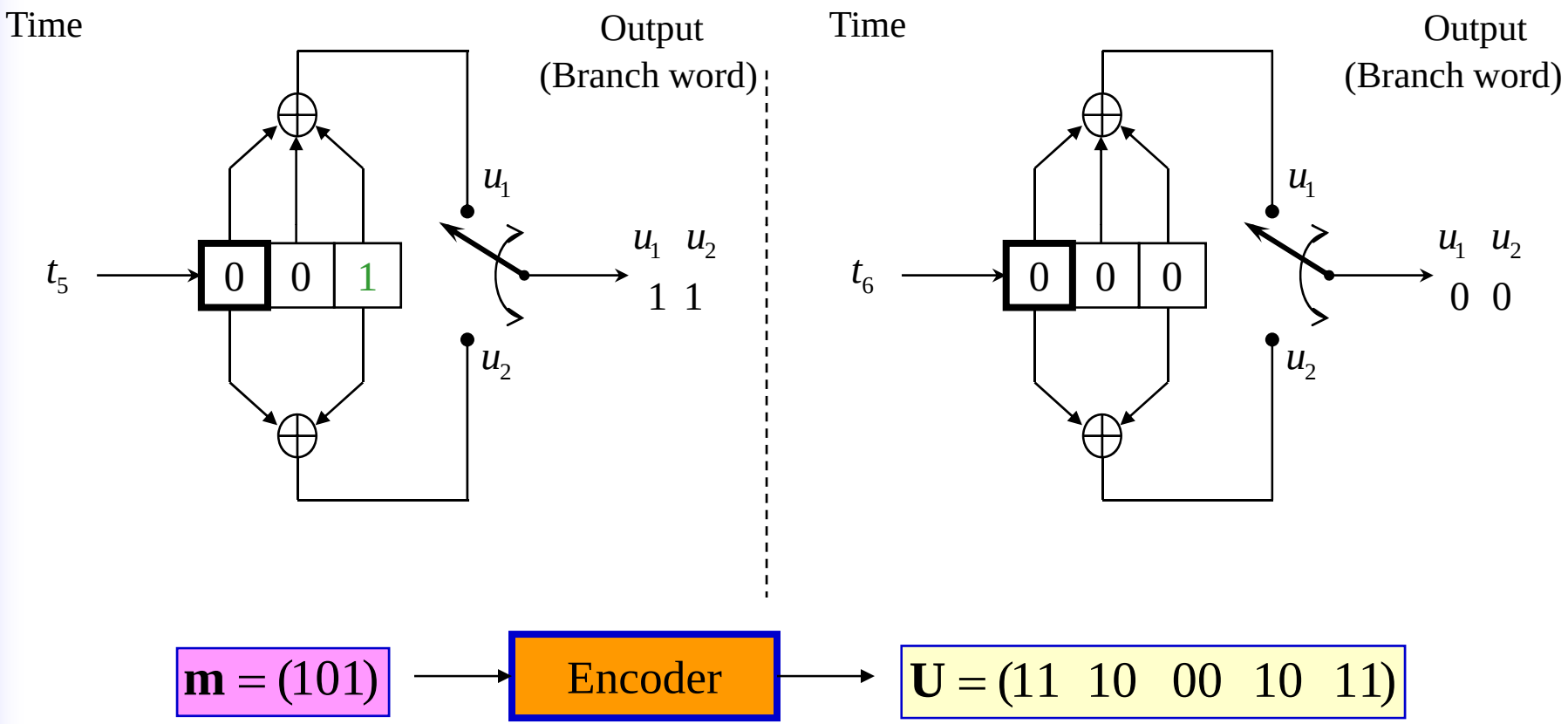


Contoh encoder konvolusi dengan rate = 1/2

Misal : Data Input m = (1 0 1)



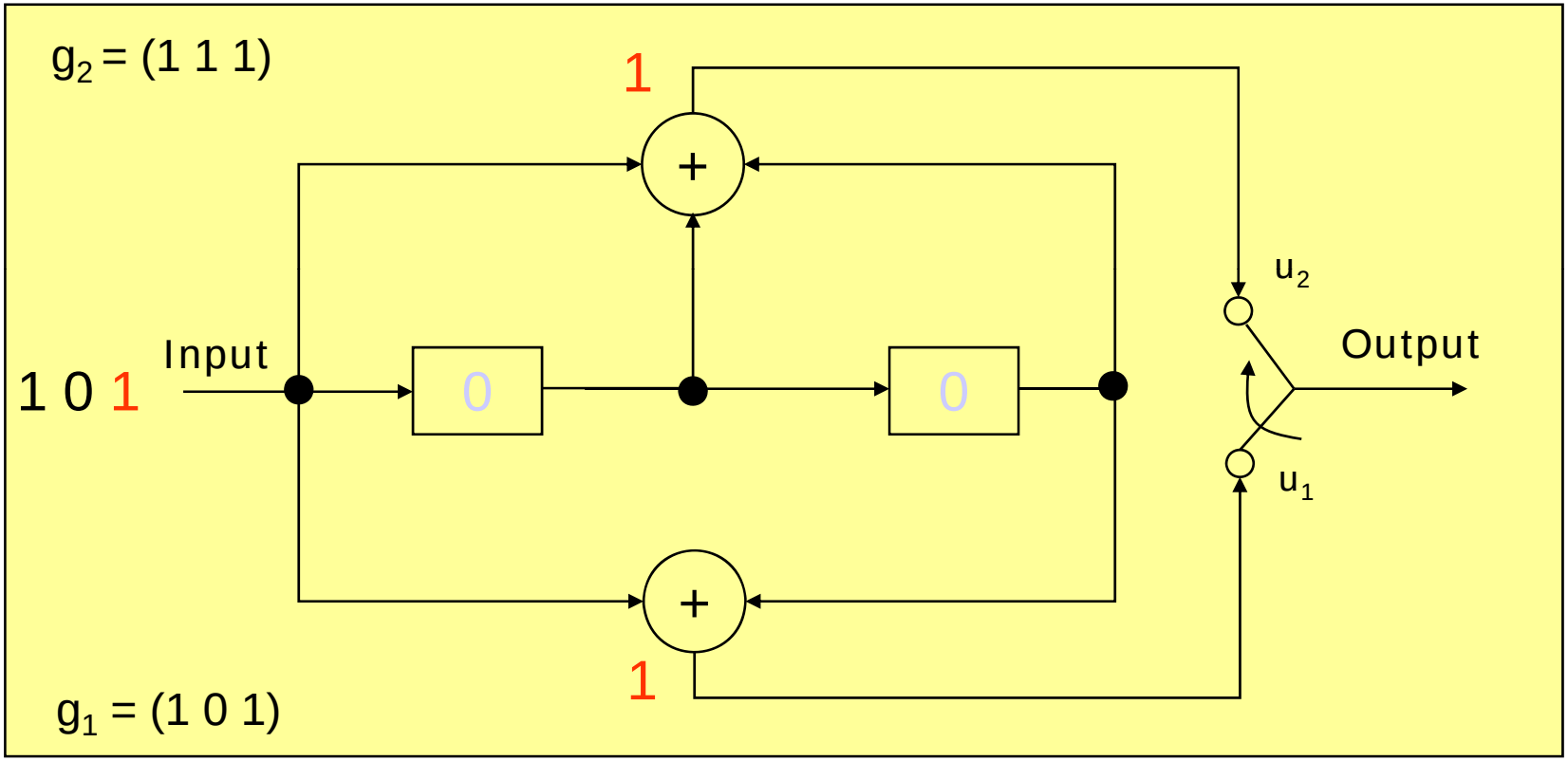
Contoh encoder konvolusi rate = ½ .....cont



Convolutional Encoder: Example

Contoh Implementasi dengan $g_1 = (1\ 0\ 1)$ dan $g_2 = (1\ 1\ 1)$

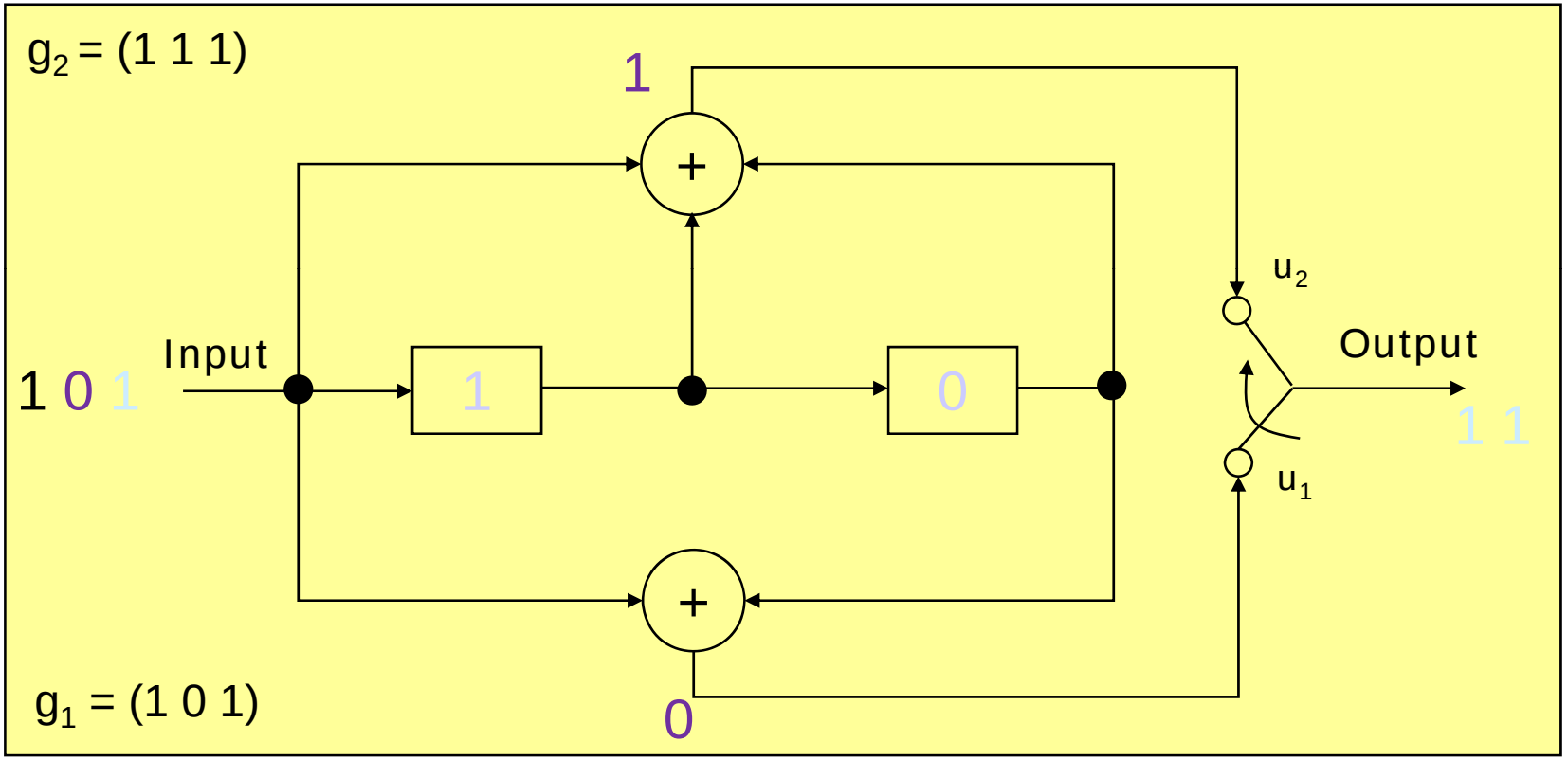
Rate ½ Convolutional Encoder



Convolutional Encoder: Example

Contoh Implementasi dengan $g_1 = (1\ 0\ 1)$ dan $g_2 = (1\ 1\ 1)$

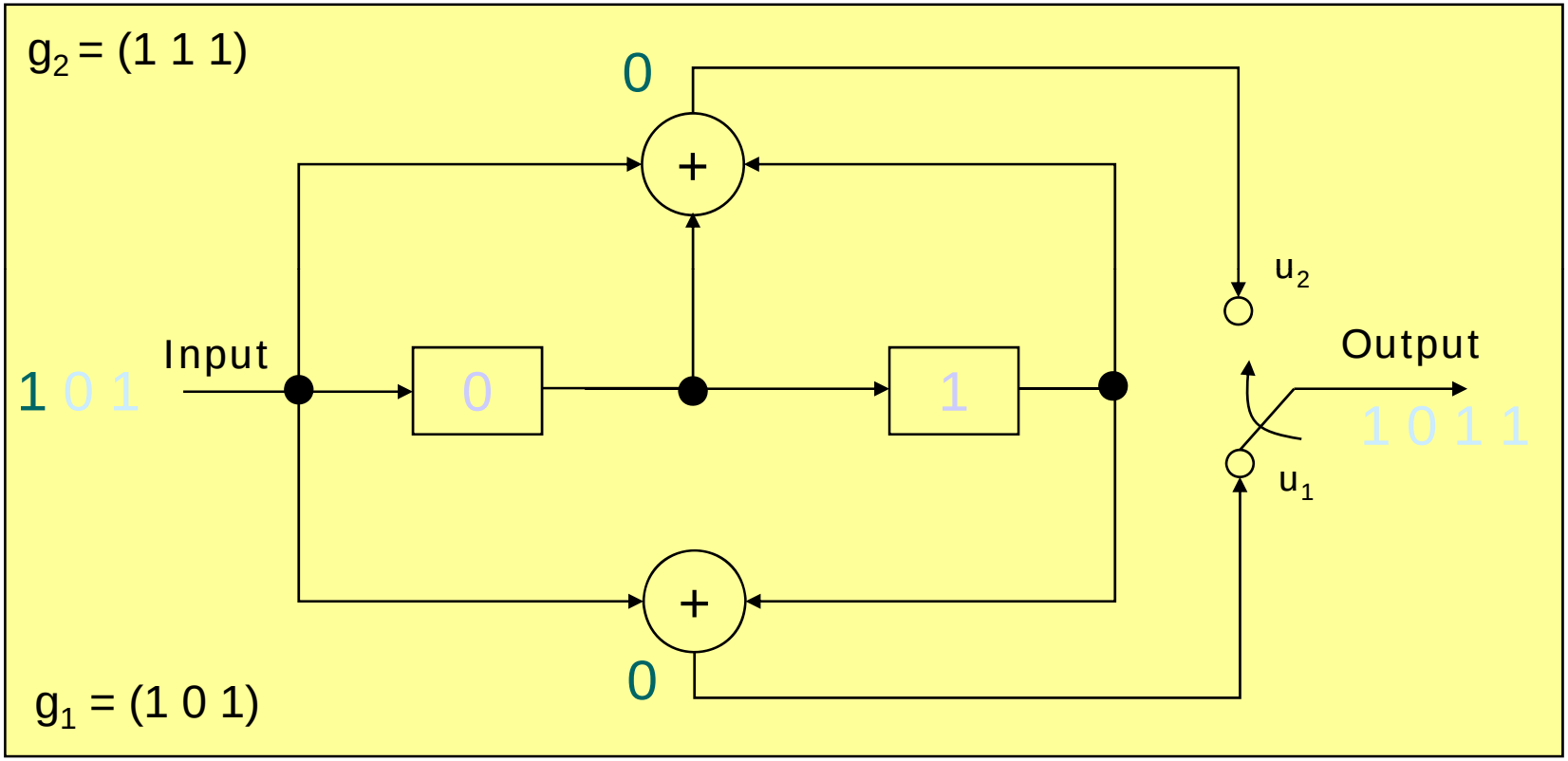
Rate ½ Convolutional Encoder



Convolutional Encoder: Example

Contoh Implementasi dengan $g_1 = (1\ 0\ 1)$ dan $g_2 = (1\ 1\ 1)$

Rate ½ Convolutional Encoder



Convolutional Encoder: Example

Contoh Implementasi dengan $g_1 = (1\ 0\ 1)$ dan $g_2 = (1\ 1\ 1)$

Rate ½ Convolutional Encoder

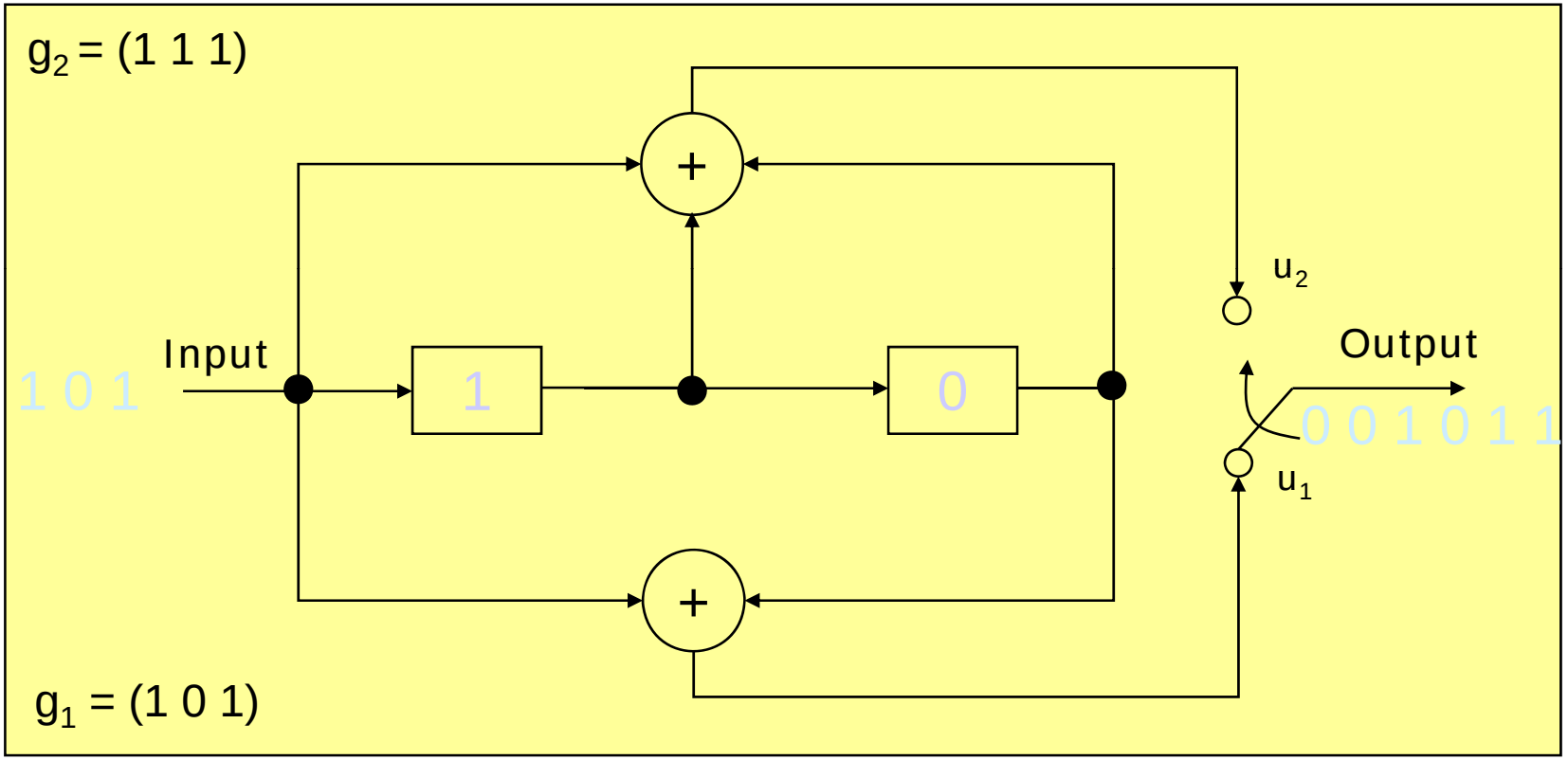
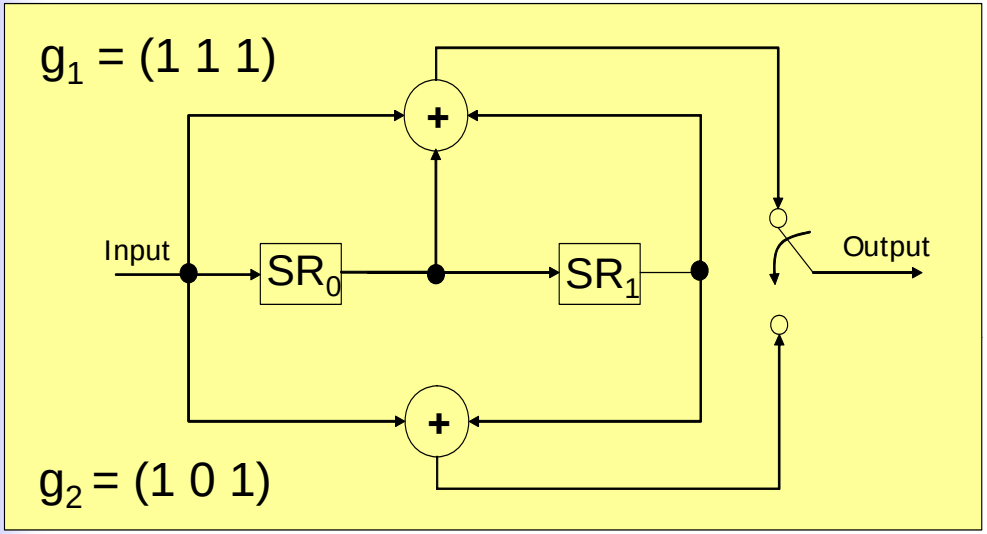


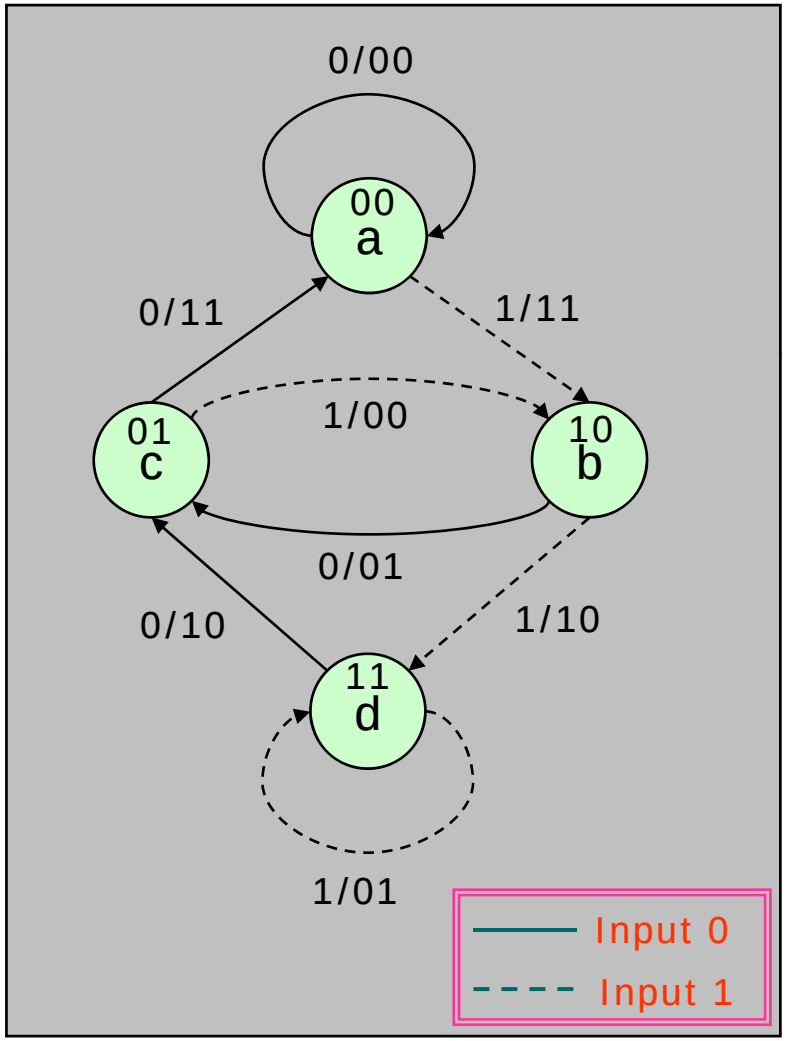
Diagram State

- Menyatakan perubahan isi / kondisi (state) dari memori shift register
- Isi memory shift register selanjutnya akan diprediksi berdasarkan data input yang masuk dan isi memory saat ini
- Terdapat 2^{K-1} state
- Diagram state juga berisi semua state dan semua kemungkinan transisi antar state

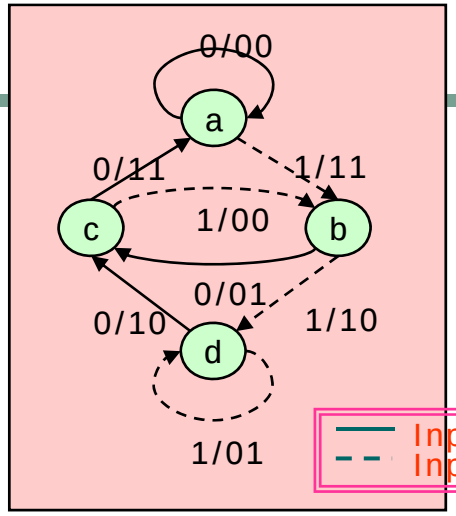
State Diagram Representation of Convolutional Codes

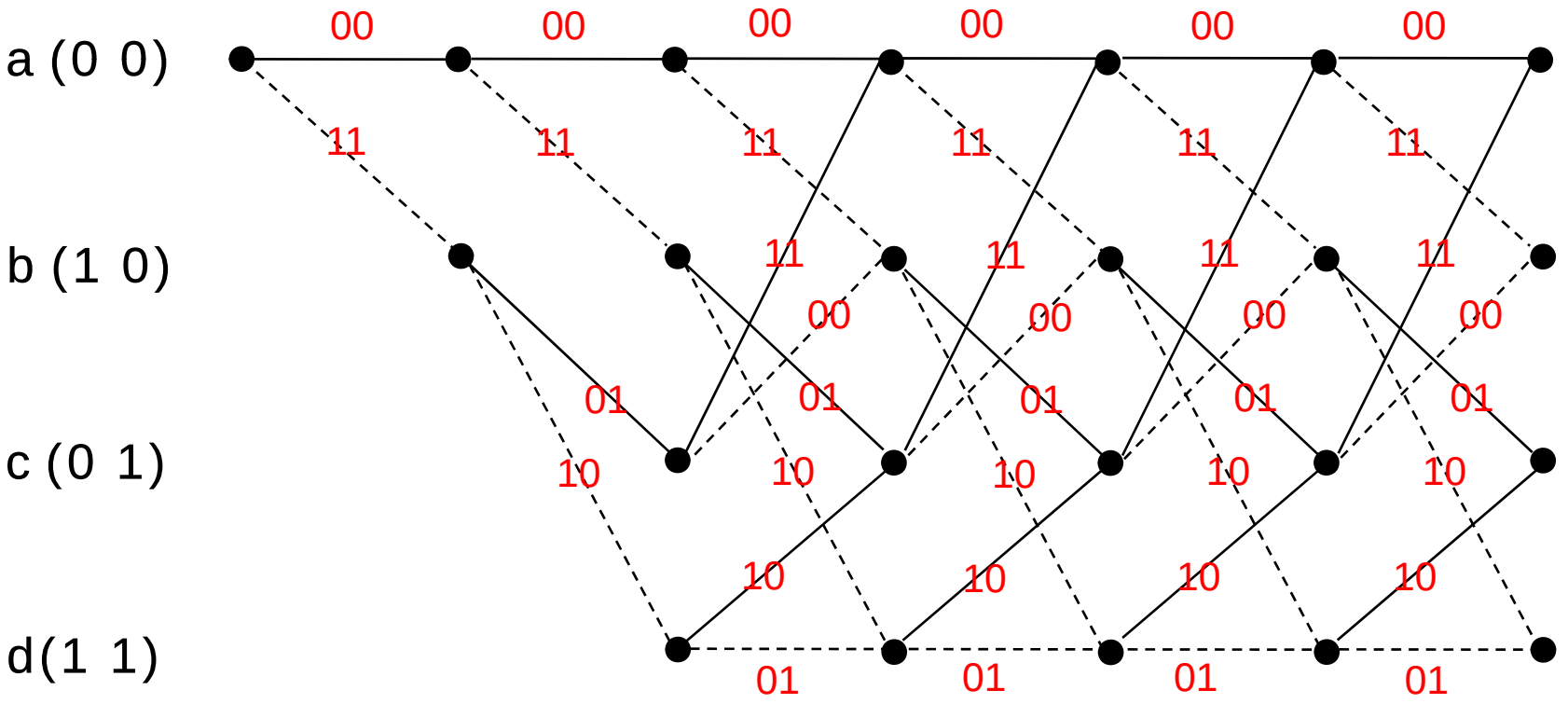


States	(SR_0	SR_1)
a	0	0
b	1	0
c	0	1
d	1	1

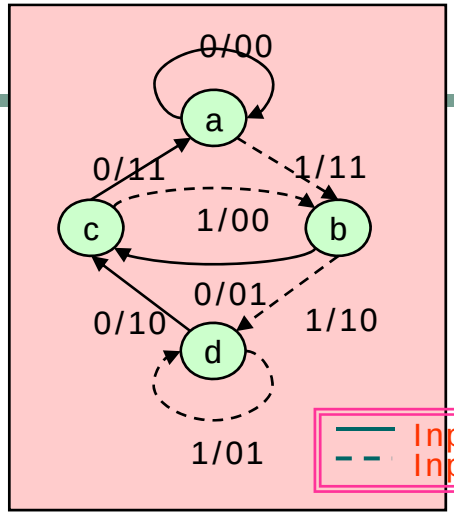


Trellis Representation of Convolutional Codes

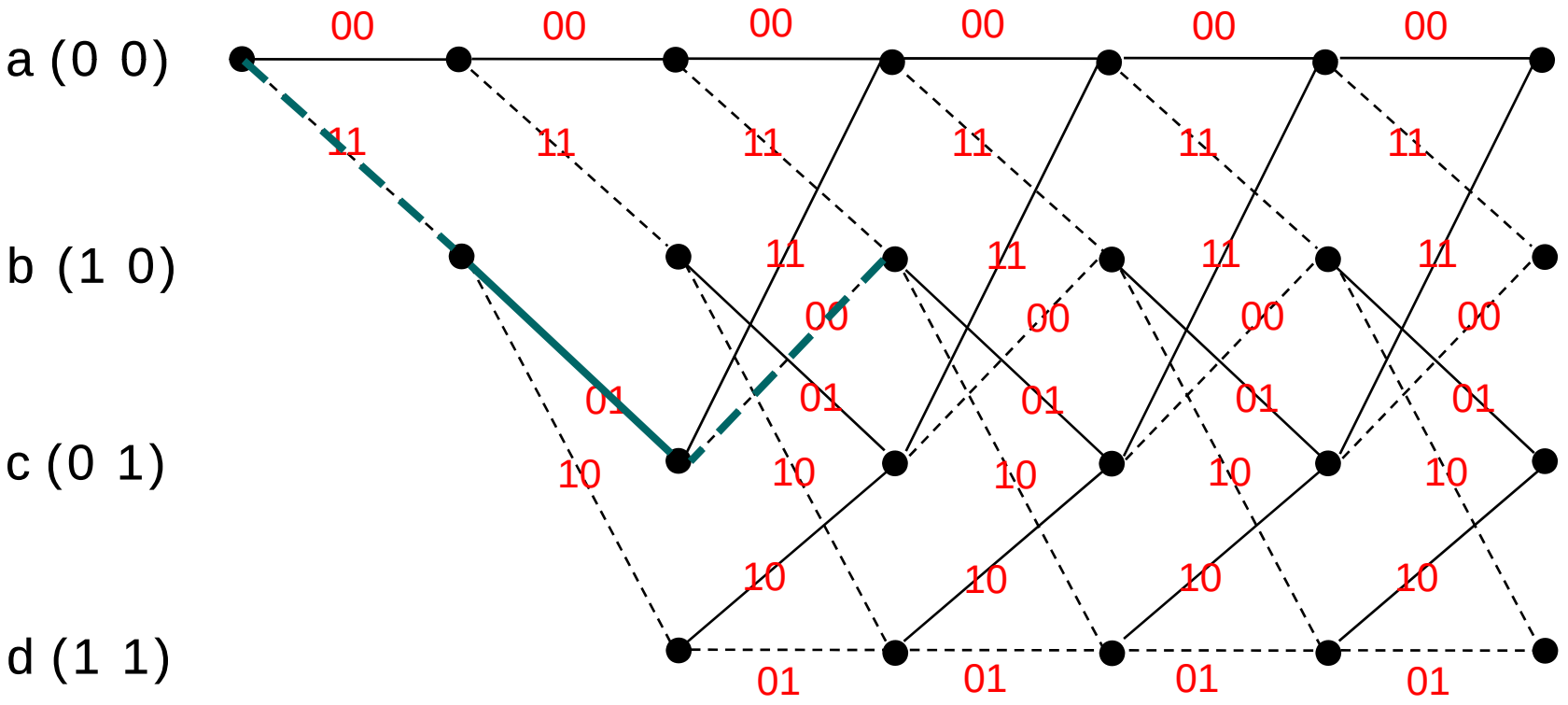




Trellis Representation of Convolutional Codes



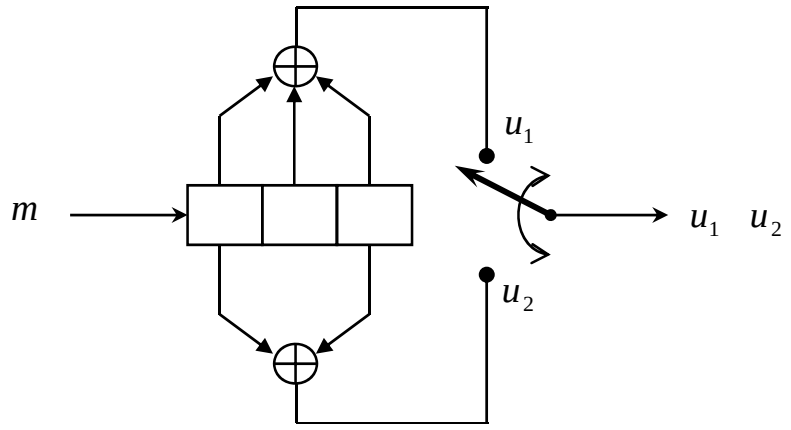
Input: 101➔ Output: 110100



Representasi Encoder Menggunakan Vektor

- Didefinisikan vektor untuk menyatakan encoder konvolusi.
- Vektor terdiri dari K element dimana masing – masing vektor menyatakan penjumlahan modulo-2.
- Elemen ke – i dari masing – masing vektor = “1” jika isi register ke - i tersebut terhubung ke penjumlahan modulo-2. Sedangkan elemen ke – i = “0” jika tidak terhubung ke penjumlahan modulo-2
  - Contoh :

$$\mathbf{g}_1 = (111)$$
$$\mathbf{g}_2 = (101)$$



Representasi Encoder Menggunakan Vektor – cont'd

- Representasi terhadap respon impuls
  - Respon encoder terhadap sebuah bit "1"
    - Contoh :

Input sequence : 1 0 0

Output sequence : 11 10 11

Register contents	Branch word u_1	u_2
100	1	1
010	1	0
001	1	1

Input **m**	Output				
1	11	10	11		
0		00	00	00	
1			11	10	11
Modulo-2 sum:	11	10	00	10	11

Representasi Encoder Menggunakan Polynomial

- Didefinisikan n buah generator polynomial yang masing – masing menyatakan penjumlahan modulo-2.
- Masing –masing polynomial mempunyai derajat *K-1* atau kurang
- Masing – masing polynomial juga menyatakan hubungan shift register yang berhubungan dengan penjumlahan modulo - 2
  - Contoh :

$$\mathbf{g}_1(X) = g_0^{(1)} + g_1^{(1)}.X + g_2^{(1)}.X^2 = 1 + X + X^2$$

$$\mathbf{g}_2(X) = g_0^{(2)} + g_1^{(2)}.X + g_2^{(2)}.X^2 = 1 + X^2$$

Output encoder menjadi :

$$\boldsymbol{U_1}(X) = \boldsymbol{m}(X)\,\boldsymbol{g_1}(X) \quad \textit{dan} \quad \boldsymbol{U_2}(X) = \boldsymbol{m}(X)\,\boldsymbol{g_2}(X)$$

Representasi Polynomial –cont'd

Untuk lebih detail :

$$\mathbf{m}(X)\mathbf{g}_1(X) = (1+X^2)(1+X+X^2) = 1+X+X^3+X^4$$

$$\mathbf{m}(X)\mathbf{g}_2(X) = (1+X^2)(1+X^2) = 1+X^4$$

$$\mathbf{m}(X)\mathbf{g}_1(X) = 1+X+0.X^2+X^3+X^4$$

$$\mathbf{m}(X)\mathbf{g}_2(X) = 1+0.X+0.X^2+0.X^3+X^4$$

$$\mathbf{U}(X) = (1,1)+(1,0)X+(0,0)X^2+(1,0)X^3+(1,1)X^4$$

$$\mathbf{U} = 11 \quad 10 \quad 00 \quad 10 \quad 11$$

Optimum decoding

- If the input sequence messages are equally likely, the optimum decoder which minimizes the probability of error is the ***Maximum likelihood*** decoder.

- ML decoder, selects a codeword among all the possible codewords which maximizes the likelihood function where $p(\mathbf{Z}|\mathbf{U}^{(m')})$ is the received sequence and $\mathbf{Z}$ is one of the possible codewords: $\mathbf{U}^{(m')}$

2^L codewords to search!!!

➢ML decoding rule:

Choose $\mathbf{U}^{(m')}$ if $p(\mathbf{Z}|\mathbf{U}^{(m')}) = \max_{\text{over all } \mathbf{U}^{(m)}} p(\mathbf{Z}|\mathbf{U}^{(m)})$

Soft and hard decisions

- In hard decision:
  - The demodulator makes a firm or hard decision whether one or zero is transmitted and provides no other information for the decoder such that how reliable the decision is.
  - Hence, its output is only zero or one (the output is quantized only to two level) which are called “hard-bits”.
- Decoding based on hard-bits is called the “hard-decision decoding”.

Soft and hard decision-cont'd

- In Soft decision:
  - The demodulator provides the decoder with some side information together with the decision.
  - The side information provides the decoder with a measure of confidence for the decision.
  - The demodulator outputs which are called soft-bits, are quantized to more than two levels.
- Decoding based on soft-bits, is called the "soft-decision decoding".
- On AWGN channels, 2 dB and on fading channels 6 dB gain are obtained by using soft-decoding over hard-decoding.

The Viterbi algorithm

- The Viterbi algorithm performs Maximum likelihood decoding.
- It find a path through trellis with the largest metric (**maximum correlation or minimum distance**).
  - It processes the demodulator outputs in an iterative manner.
  - At each step in the trellis, it compares the metric of all paths entering each state, and keeps only the path with the largest metric, called the survivor, together with its metric.
  - It proceeds in the trellis by eliminating the least likely paths.
- It reduces the decoding complexity to $L2^{K-1}$!

The Viterbi algorithm - cont'd

- Viterbi algorithm:

A. Do the following set up:

- For a data block of L bits, form the trellis. The trellis has $L+K-1$ sections or levels and starts at time t_1 and ends up at time t_{L+K} .
- Label all the branches in the trellis with their corresponding **branch metric**.
- For each state in the trellis at the time t_i which is denoted by $S(t_i) \in \{0,1,...,2^{K-1}\}$, define a parameter $\Gamma(S(t_i), t_i)$

B. Then, do the following:

The Viterbi algorithm - cont'd

1. Set $\Gamma(0, t_1) = 0$ and $i = 2$.
2. At time t_i , compute the partial path metrics for all the paths entering each state.
3. Set $\Gamma\big(S(t_i), t_i\big)$ equal to the best partial path metric entering each state at time t_i .

   Keep the survivor path and delete the dead paths from the trellis.
4. If $i < L + K$, increase i by 1 and return to step 2.

C. Start at state zero at time t_{L+K} . Follow the surviving branches backwards through the trellis. The path thus defined is unique and correspond to the ML codeword.

Maximum Likelihood Decoding of Convolutional Codes

Maximum Likelihood Decoding:

What is the transmitted sequence that will most likely result in the received sequence at the decoder side?

Viterbi Decoding of Convolutional Codes:

Maximum likelihood decoding algorithm

An algorithm that finds the closest codeword to a given received sequence

Hard Decision: **Closest ➔ Minimum Hamming Distance**

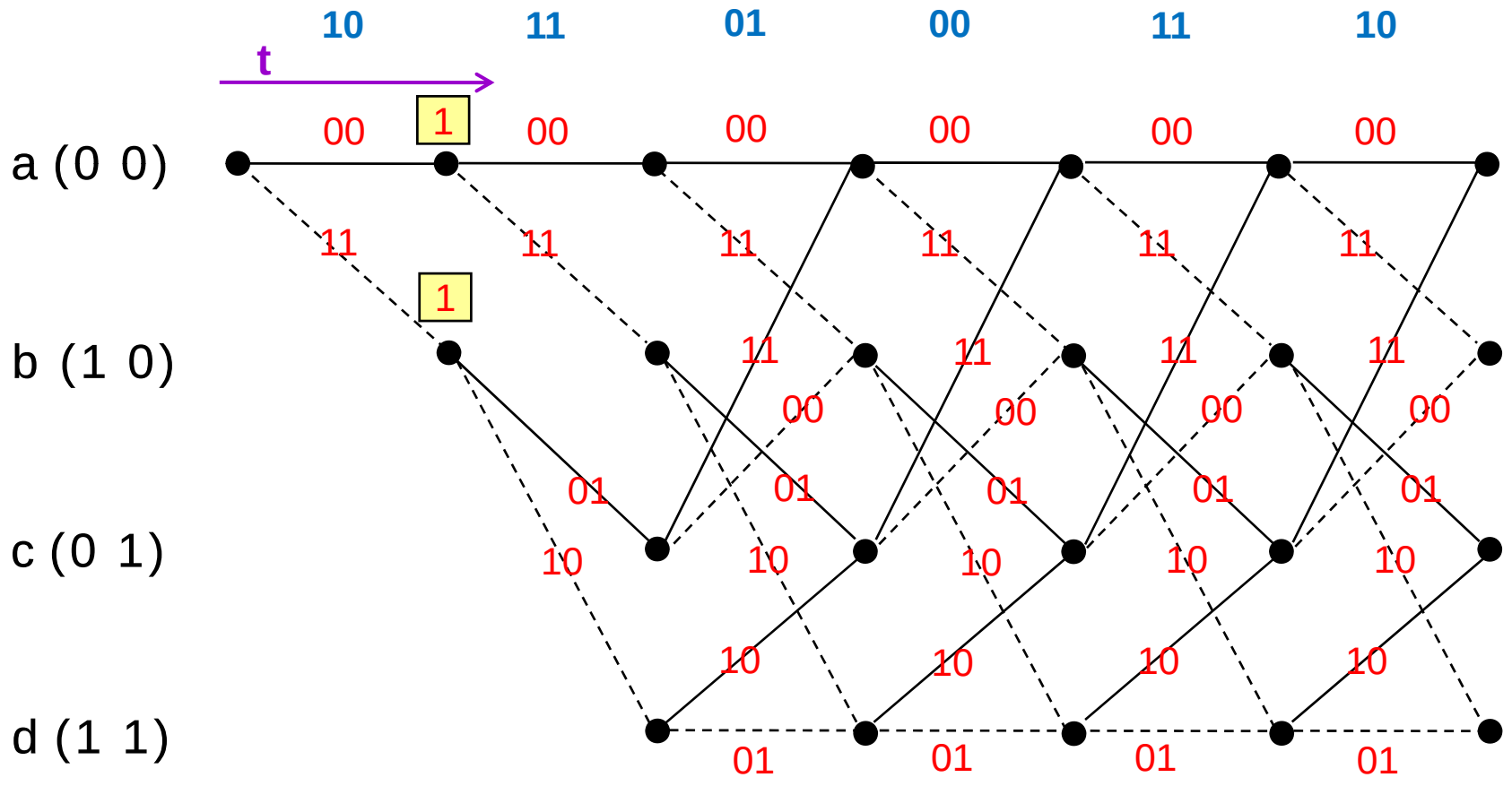
Soft Decision : Closest ➔ Minimum Voltage Separation

Hard Decision:

- The receiver makes a firm hard decision whether one or zero is received
- The receiver provides no information to the decoder characterizing reliability of its decision
- The input to the decoder is only zero or one

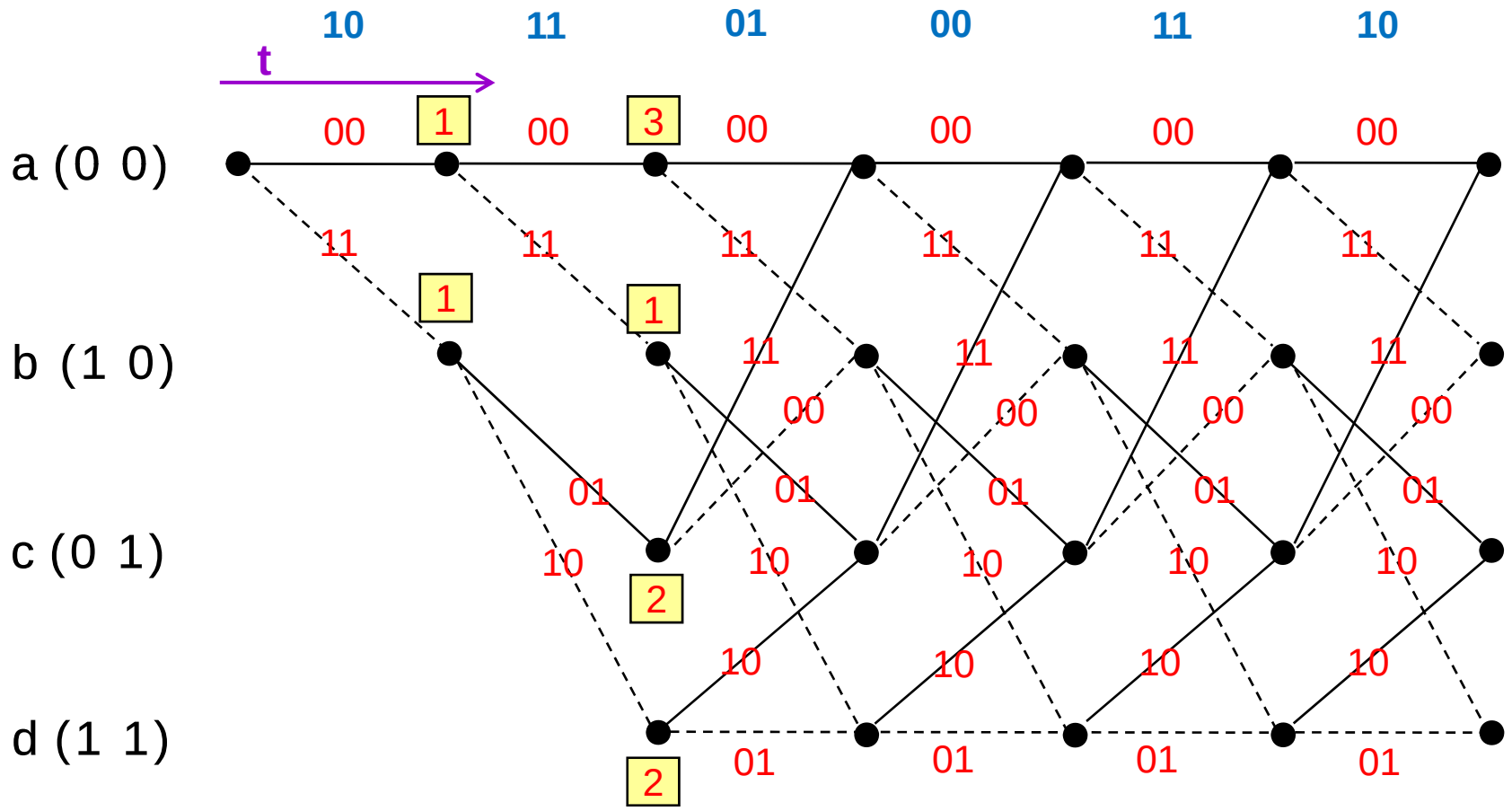
Viterbi Decoder Hard Decision

Assume **received** (Hard Decision) vector is t ← 01 11 00 10 11 01



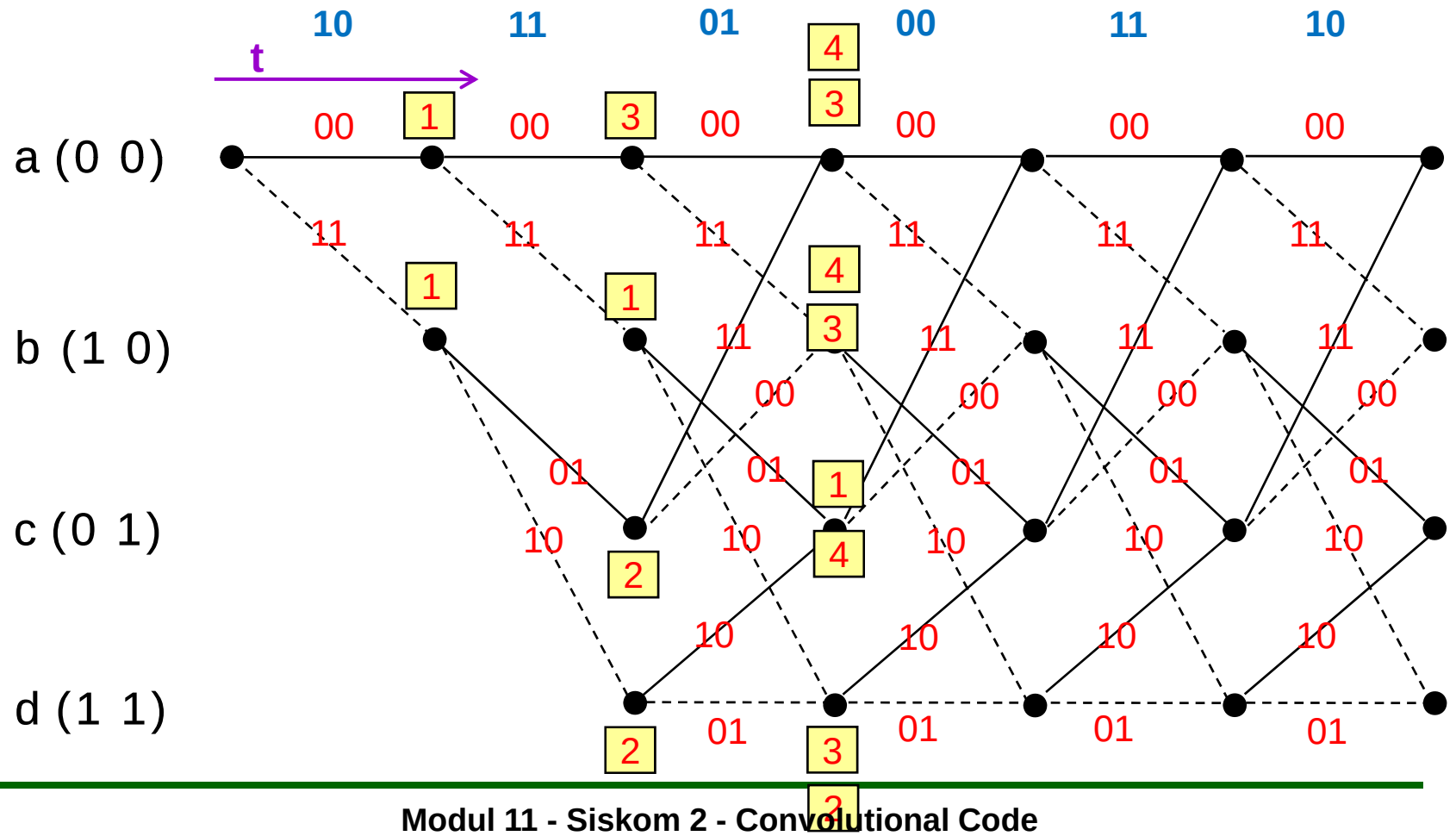
Viterbi Decoder Hard Decision

Assume **received** (Hard Decision) vector is t ← 01 11 00 10 11 01



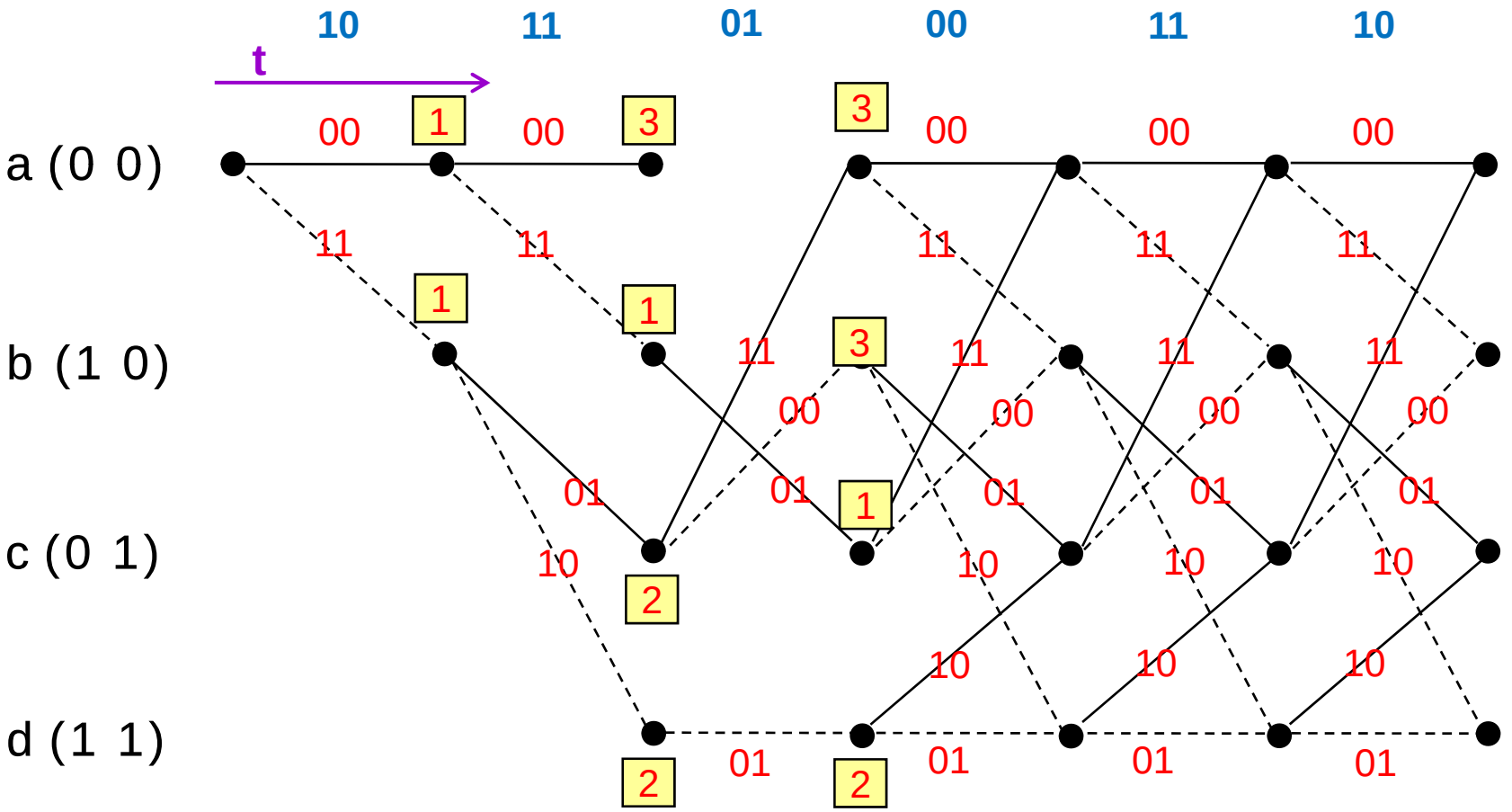
Viterbi Decoder Hard Decision

Assume **received** (Hard Decision) vector is t ← 01 11 00 10 11 01



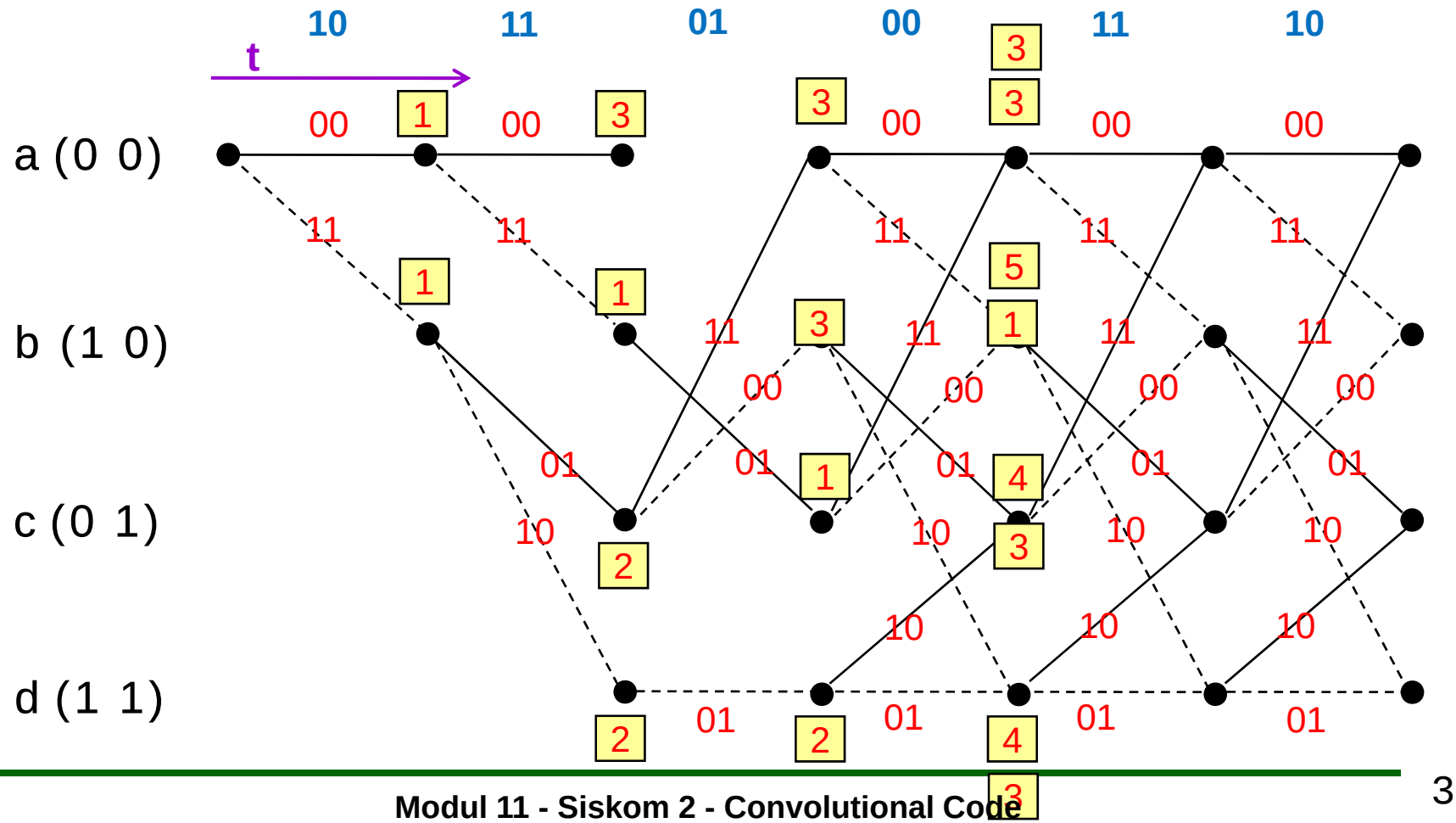
Viterbi Decoder Hard Decision

Assume **received** (Hard Decision) vector is t ← 01 11 00 10 11 01



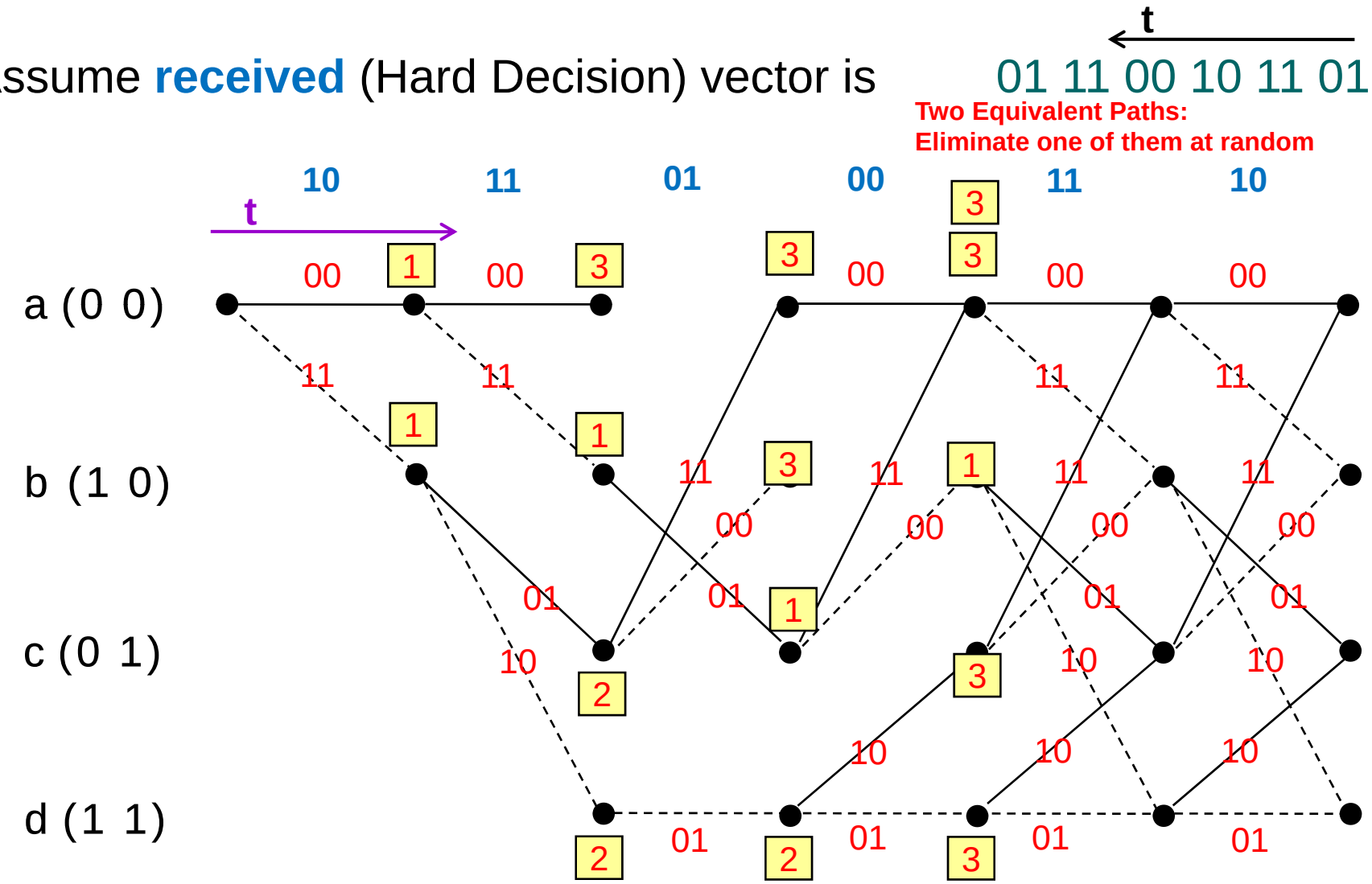
Viterbi Decoder Hard Decision

Assume **received** (Hard Decision) vector is t ← 01 11 00 10 11 01



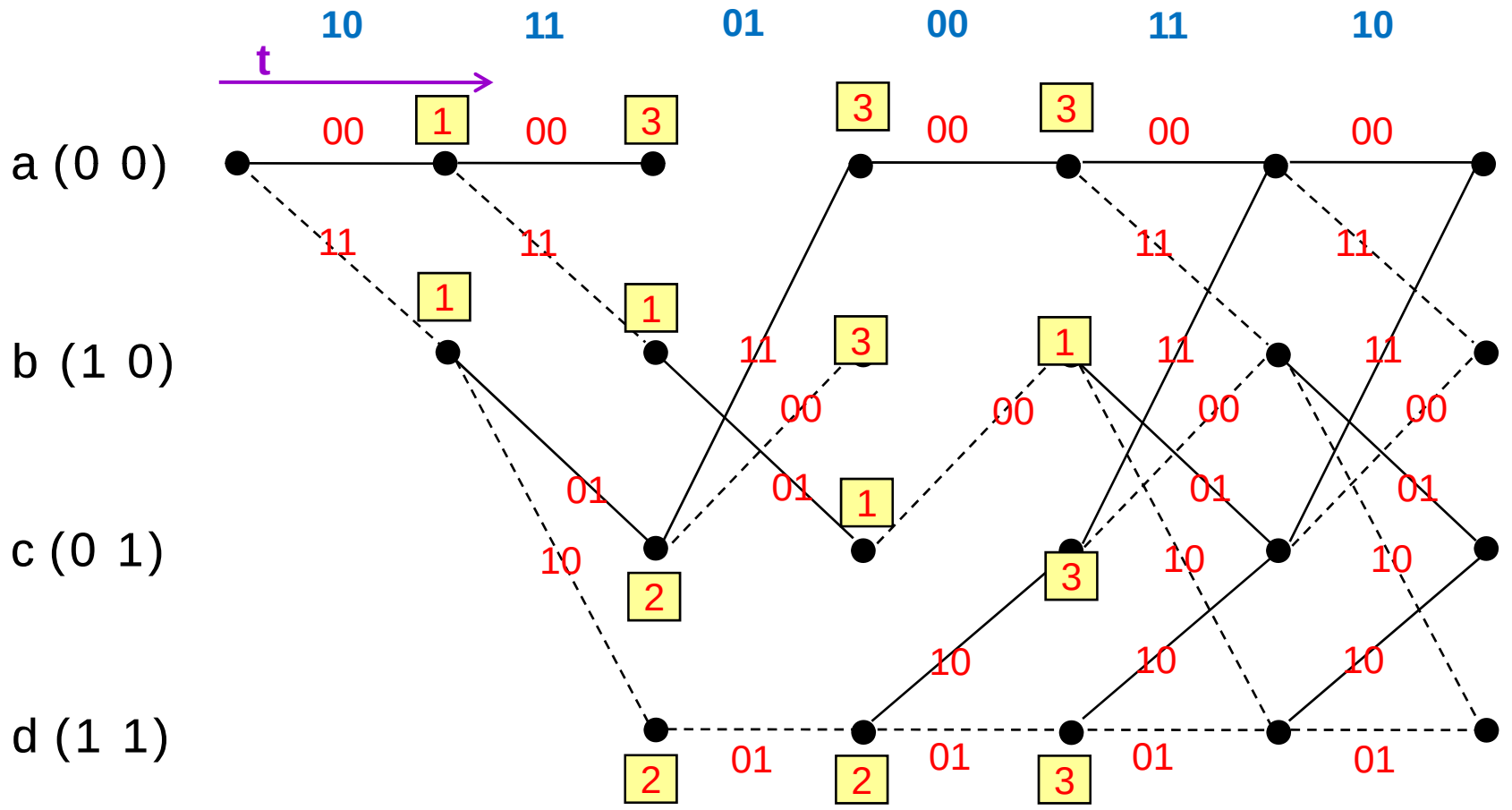
Viterbi Decoder Hard Decision

Assume **received** (Hard Decision) vector is 01 11 00 10 11 01



Viterbi Decoder Hard Decision

Assume **received** (Hard Decision) vector is t ← 01 11 00 10 11 01

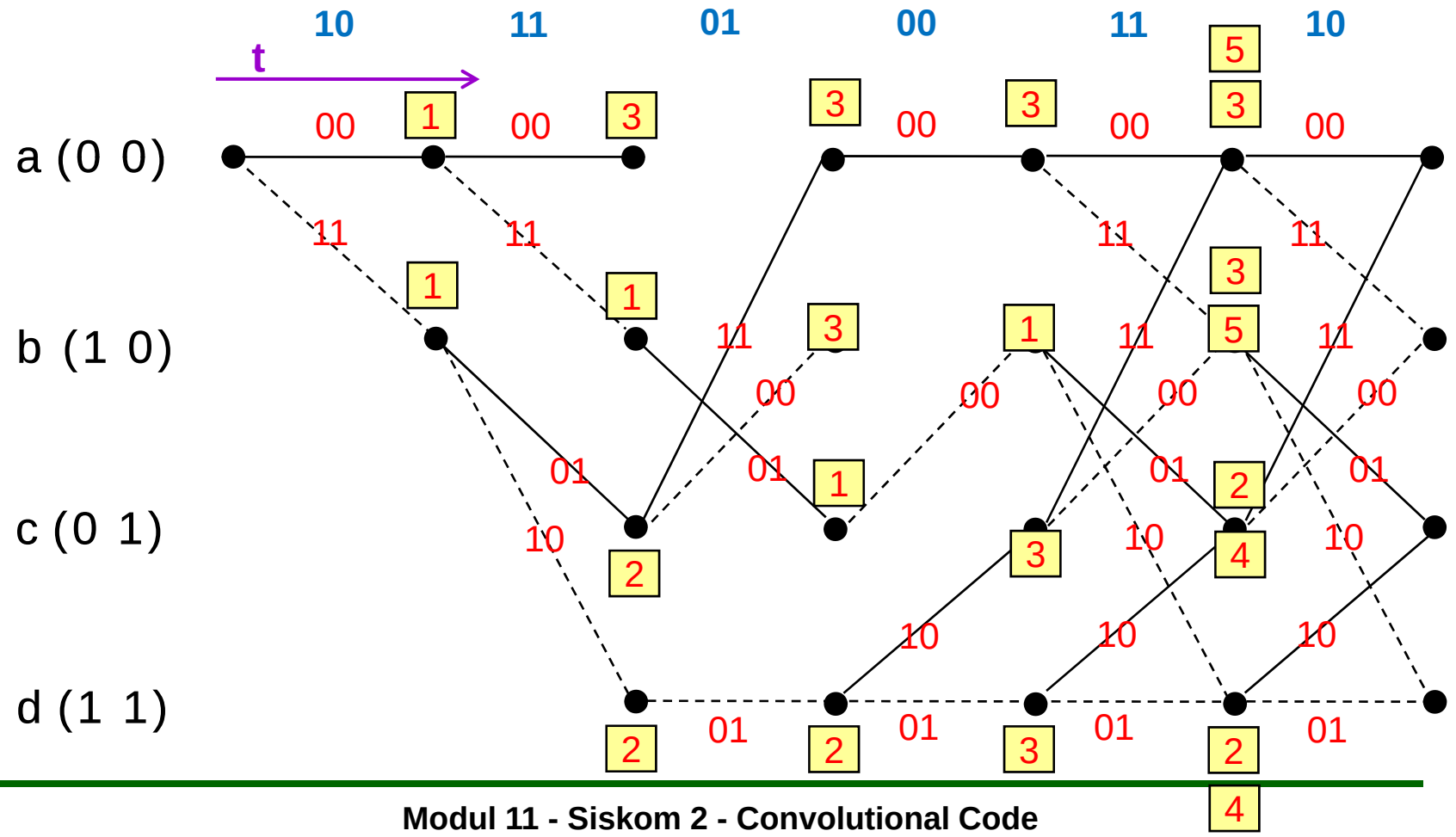


Viterbi Decoder Hard Decision

Assume **received** (Hard Decision) vector is

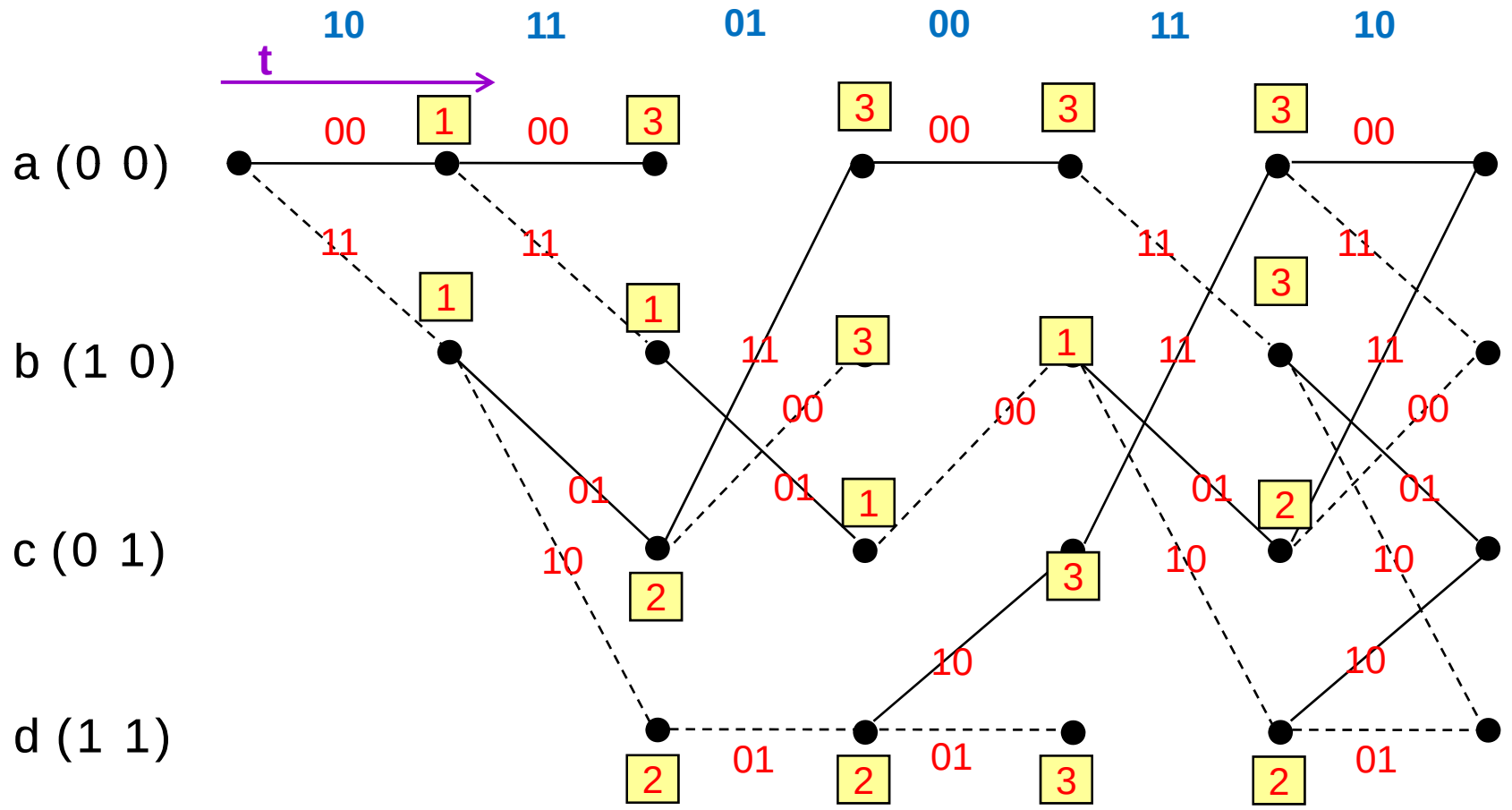
t

01 11 00 10 11 01



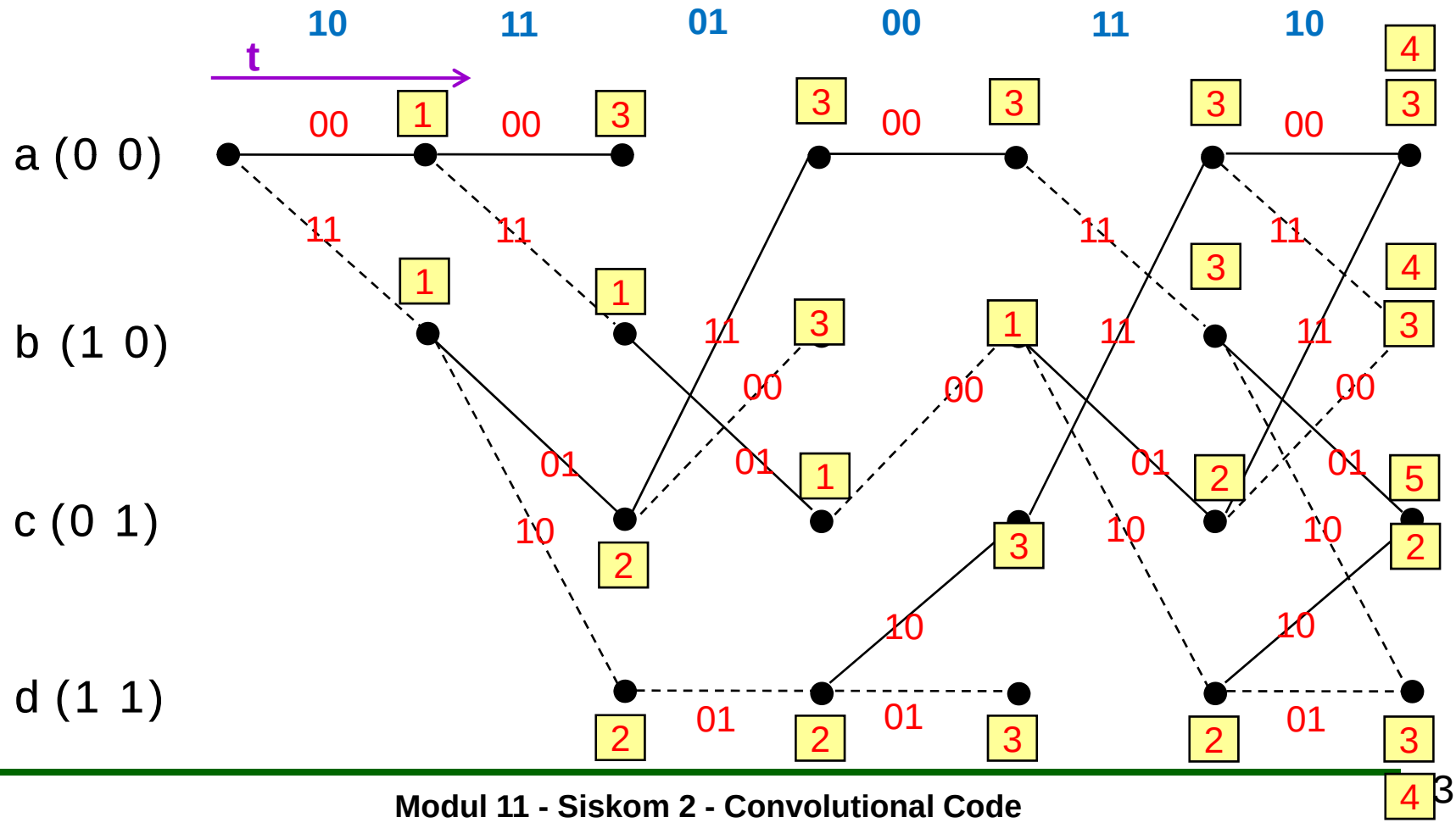
Viterbi Decoder Hard Decision

Assume **received** (Hard Decision) vector is t ← 01 11 00 10 11 01



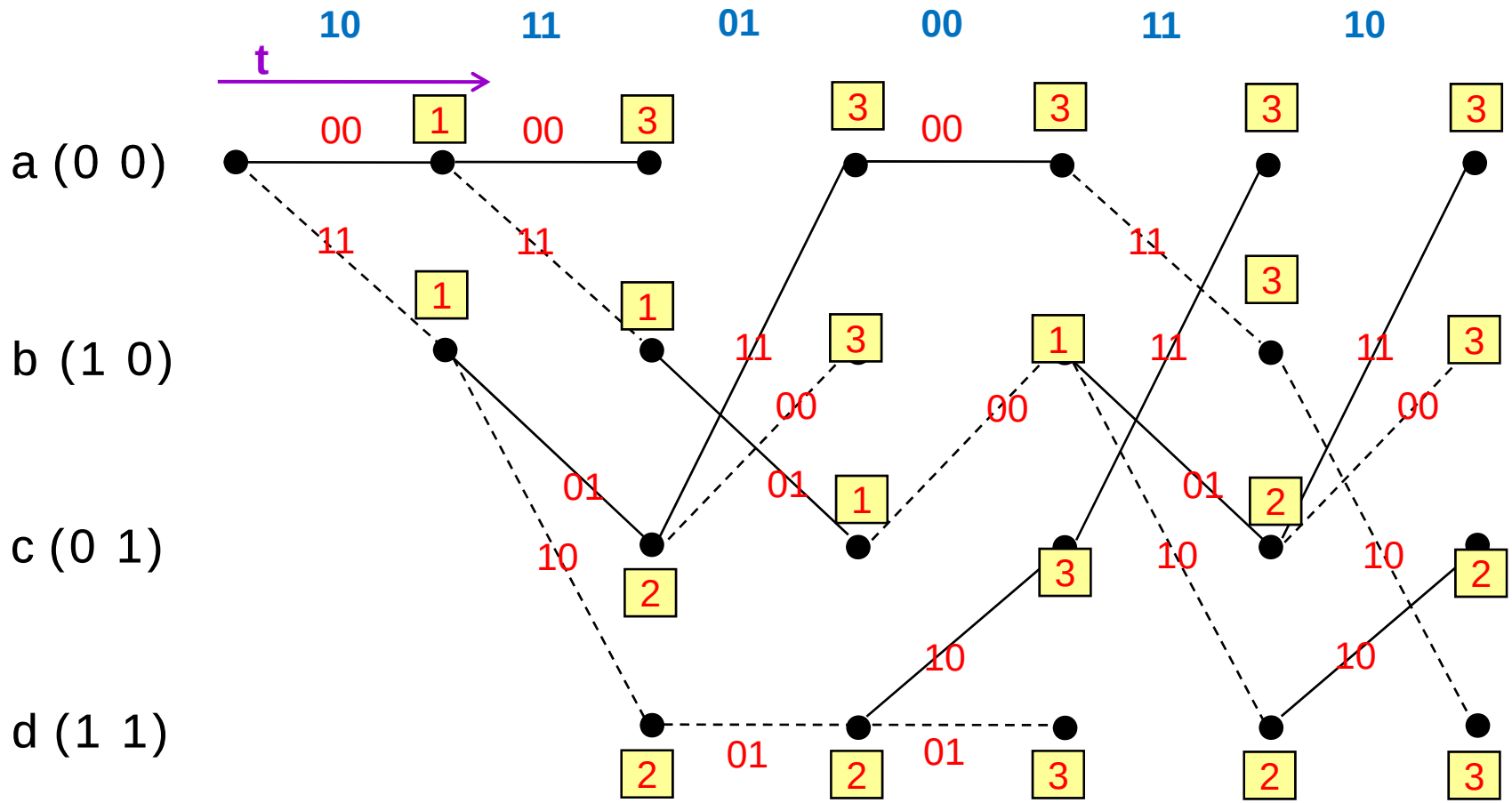
Viterbi Decoder Hard Decision

Assume **received** (Hard Decision) vector is t ← 01 11 00 10 11 01



Viterbi Decoder Hard Decision

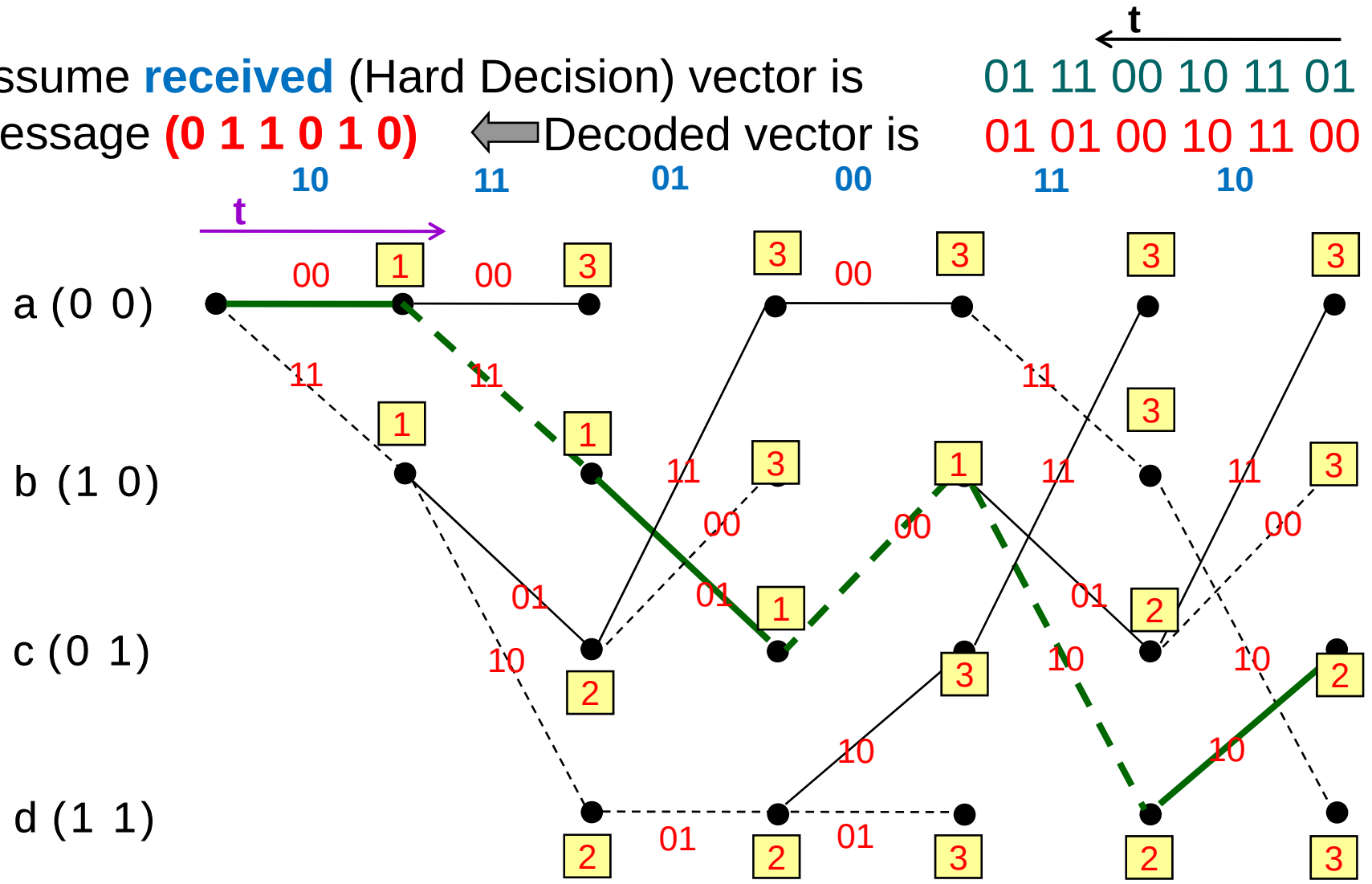
Assume **received** (Hard Decision) vector is t ← 01 11 00 10 11 01



Viterbi Decoder Hard Decision

Assume **received** (Hard Decision) vector is 01 11 00 10 11 01

Message **(0 1 1 0 1 0)** ⇐ Decoded vector is 01 01 00 10 11 00

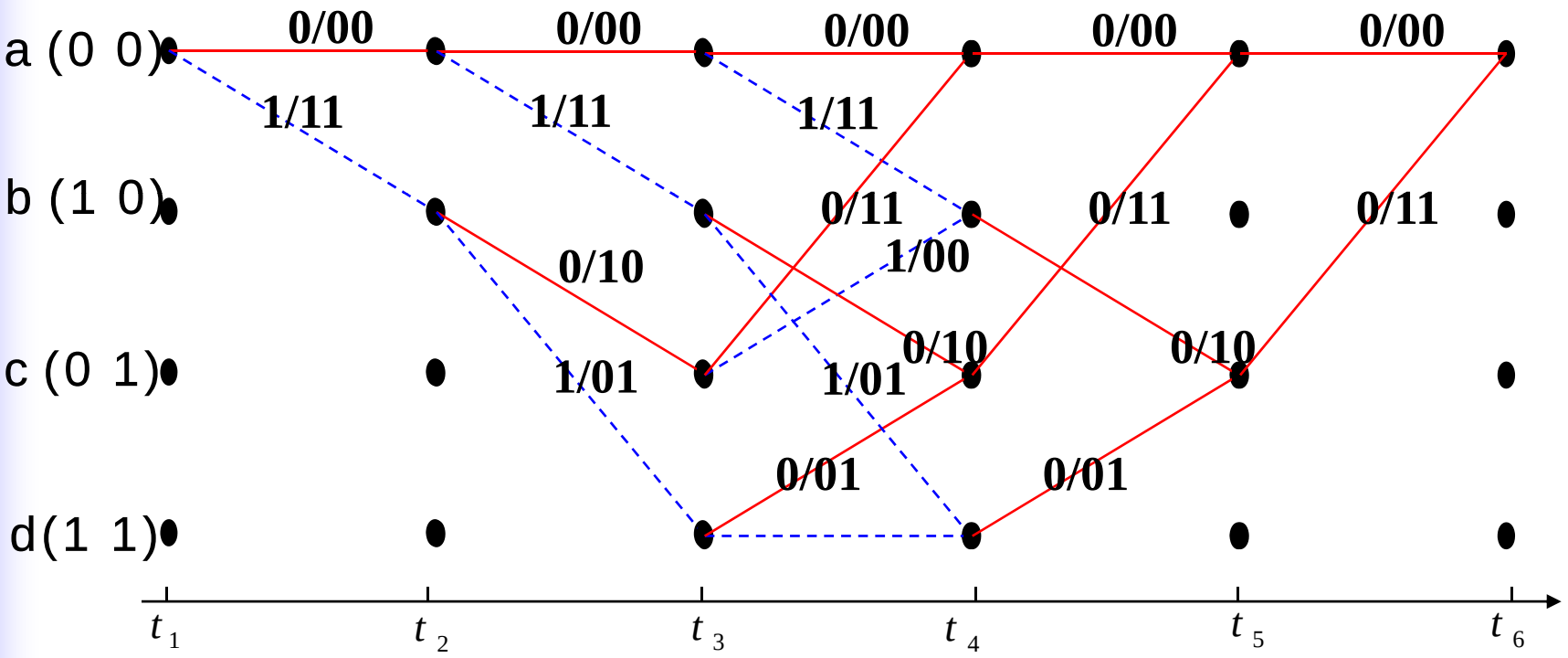


Latihan soal: Hard decision Viterbi decoding

m = (10100)

$\mathbf{U} = (11\ \ 10\ \ 00\ \ 10\ \ 11)$

$\mathbf{Z} = (11\ \ 10\ \ 11\ \ 10\ \ 01)$



Tentukan data kirim ?

Apakah terjadi error data ?

Free distance of Convolutional codes

- Distance properties:
  - Since a Convolutional encoder generates codewords with various sizes (as opposite to the block codes), the following approach is used to find the minimum distance between all pairs of codewords:
    - Since the code is linear, the minimum distance of the code is the minimum distance between each of the codewords and the all-zero codeword.
    - This is the minimum distance in the set of all arbitrary long paths along the trellis that diverge and remerge to the all-zero path.
    - It is called the minimum free distance or the free distance of the code, denoted by d_{free} or d_f

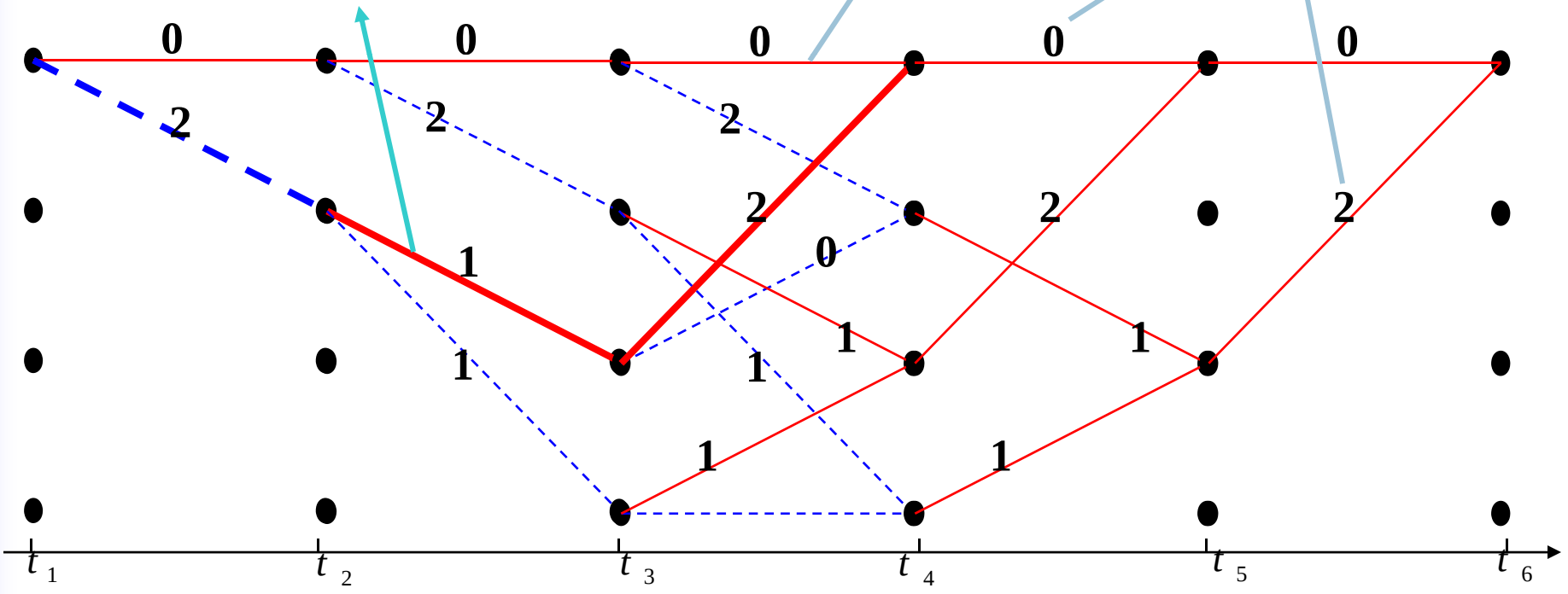
Free distance ...

The path diverging and remerging to all-zero path with **minimum weight**

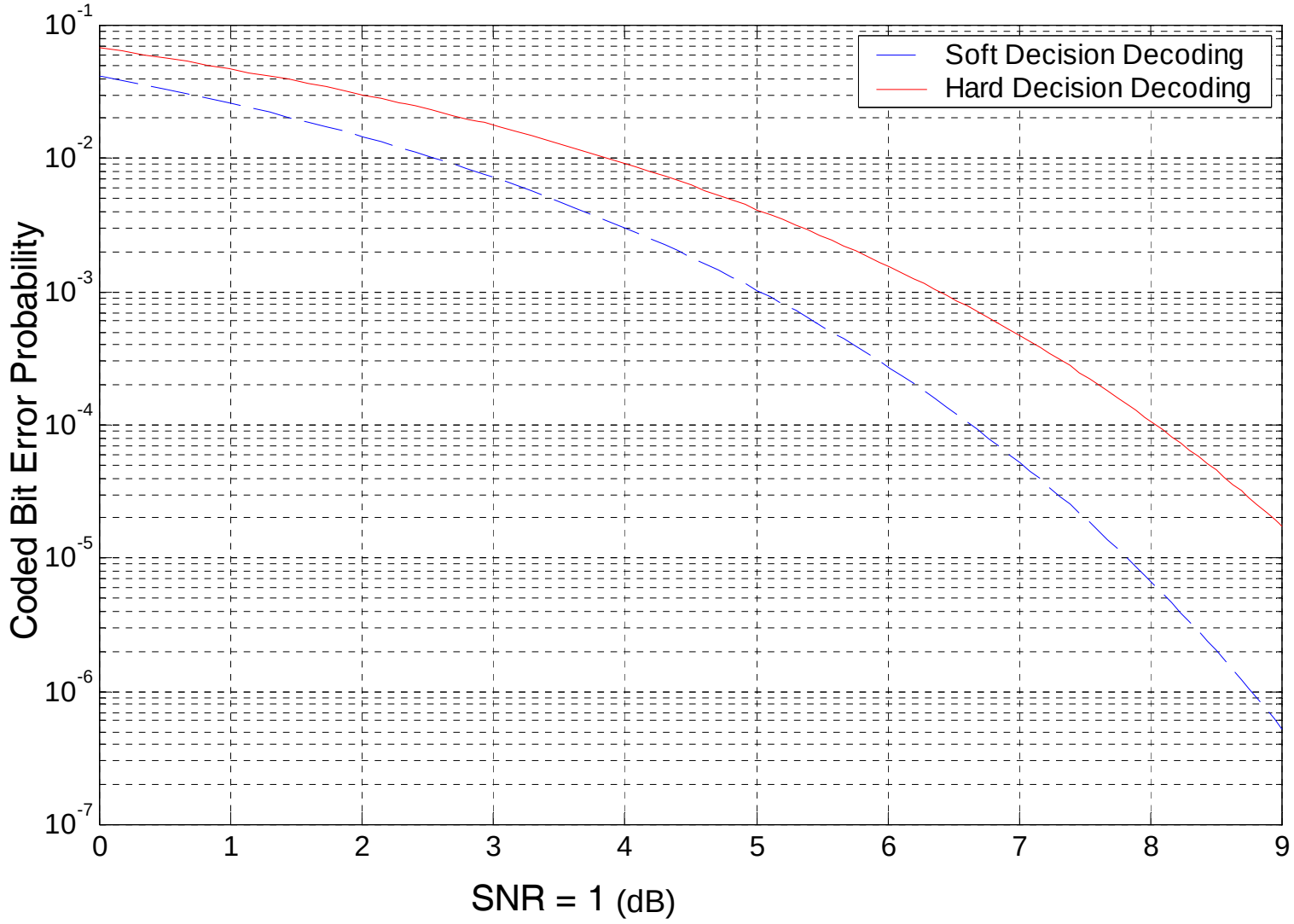
$d_f = 5$

All-zero path

Hamming weight of the branch



Hard Decision Vs Soft Decision Performance: Example 1/3 Repetition Code



Tugas, Dikumpulkan !

Exercise 1

Draw the state diagram, tree diagram, and trellis diagram for the Convolutional encoder with the generator vectors

$$\mathbf{g}_1 = (1, 1, 1)$$
$$\mathbf{g}_2 = (0, 1, 1).$$

Exercise 2

Consider the $K = 3$, rate $\frac{1}{2}$ encoder with generator vectors

$$\mathbf{g}_1 = (1, 1, 1)$$
$$\mathbf{g}_2 = (1, 0, 1).$$

This code is used over a binary symmetric channel (BSC). Assume that the initial encoder state is the 00 state. At the output of the BSC, the sequence $\mathbf{Z} = (11, 00, 00, 10, 11,$ rest all "0") is received.

a) Find the maximum likehood path through the trellis diagram !
b) Determine the first 5 decode data bits !