Redes e Sistemas de Telecomunicações

RT001

Teoria da Informação e Codificação de Canal (Parte 3)

Prof. Dr. Estevan Lopes

Inatel

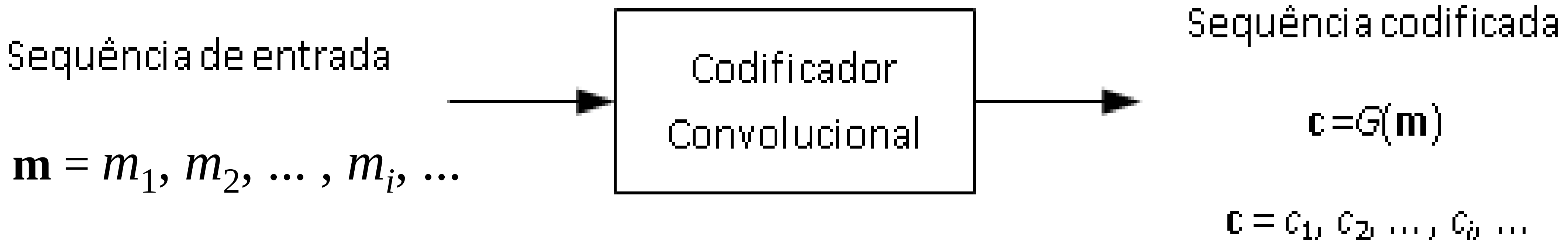
CAMINHOS QUE CONECTAM COM O FUTURO

CAPÍTULO 4

4.1. INTRODUÇÃO

- Além dos Códigos de Blocos Lineares, os Códigos Convolucionais compõem outra grande família de códigos corretores de erros.

- Considere o bloco de codificação apresentado na Figura 1, que representa um codificador convolucional. A mensagem **m** a ser codificada é representada pela sequência $\mathbf{m} = m_1, m_2, \ldots, m_i, \ldots$, onde cada m_i representa um bit e i é o índice correspondente ao tempo.



onde $c_i = c_{1i}, \ldots, c_{ji}, \ldots, c_{ni}$

Figura 1 – Bloco de codificação convolucional.

- Suponha que m_i assuma os valores zero ou um de forma equiprovável e que a presença de um ou outro seja estatisticamente independente, ou seja, o bit presente não depende de seu predecessor nem influencia seu sucessor.

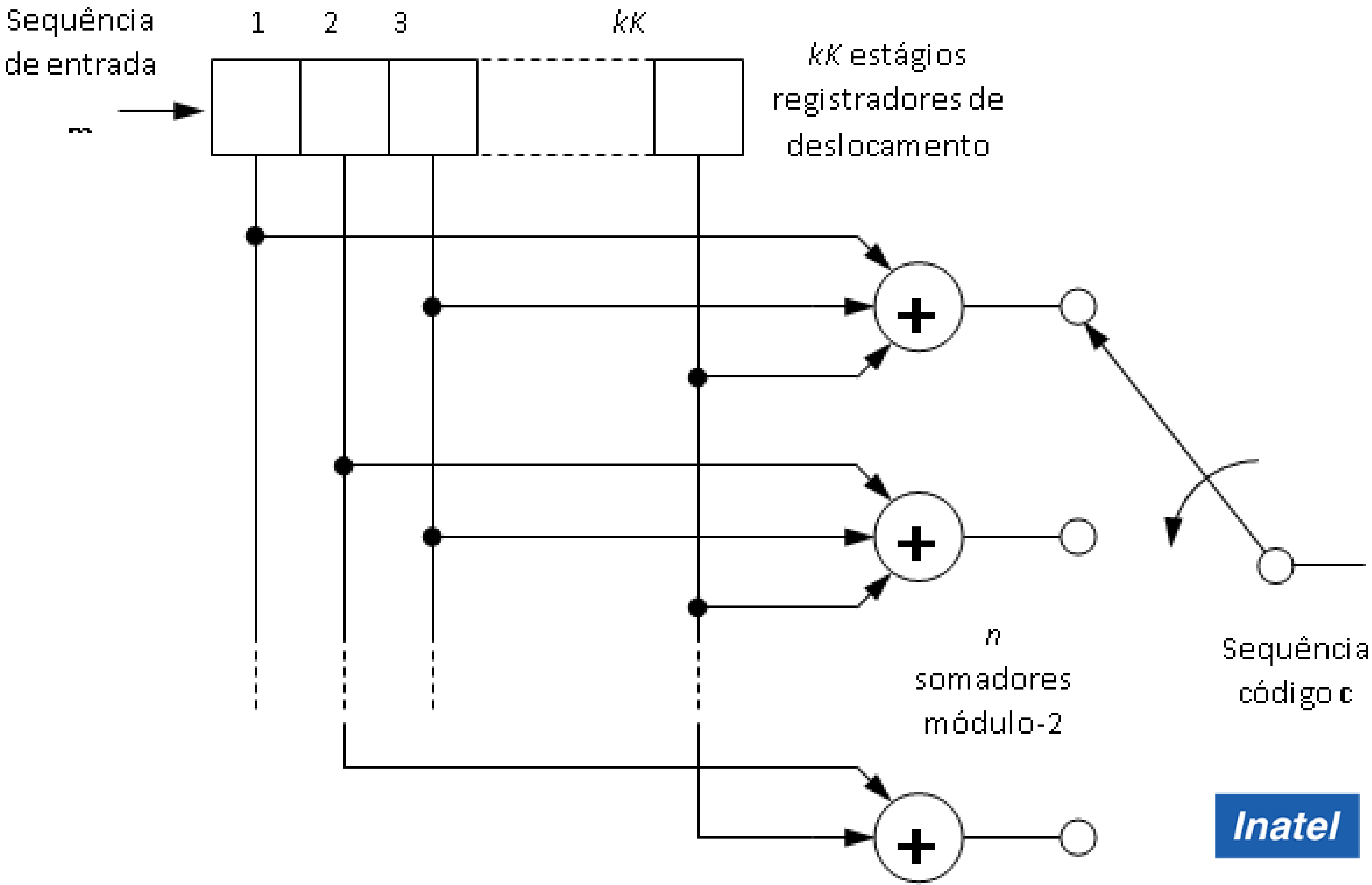
4.1. INTRODUÇÃO

- O codificador transforma cada sequência **m** em uma única sequência código **c** = *G*(**m**). A sequência **U** pode ser segmentada em uma sequência de *palavras ramos*: **c** = c_1, c_2, ... , c_i , ... Cada palavra ramo c_i é um *símbolo código* binário, frequentemente chamado de *símbolo do canal*, *bits do canal* ou *bits códigos*. Ao contrário dos bits da mensagem de entrada, os bits dos símbolos códigos não são independentes.

- Isto significa que embora uma sequência **m** defina uma única sequência **c**, uma característica chave dos códigos convolucionais é que um elemento m_i de uma sequência de entrada **m** não é suficiente para definir seu símbolo código associado c_i em **U,** uma vez que a codificação de cada elemento m_i não é apenas função do próprio m_i, mas é também do elemento m_{i-1} que o precedeu.
- Um codificador convolucional genérico, mostrado na Figura a seguir, é constituído de um registrador de deslocamento de *kK* estágios e somadores módulo-2, onde *K* determina a *extensão de influência*, ou *profundidade* do código. A profundidade influência de um código convolucional é definida como sendo o máximo número de bits codificados que podem ser afetados por um único bit de entrada.

- A cada unidade de tempo, *k* bits são deslocados para os primeiros *k* estágios do registrador; todos os bits no registrador são deslocados *k* bits para a direita e, as saídas dos *n* somadores são sequencialmente amostradas para a obtenção do símbolo código ou bits códigos. Uma vez que *n* bits códigos são obtidos na saída para cada grupo de *k* bits de mensagem, a taxa do código é *k*/*n* bits de mensagem por bits códigos, onde $k < n$.

4.1. INTRODUÇÃO



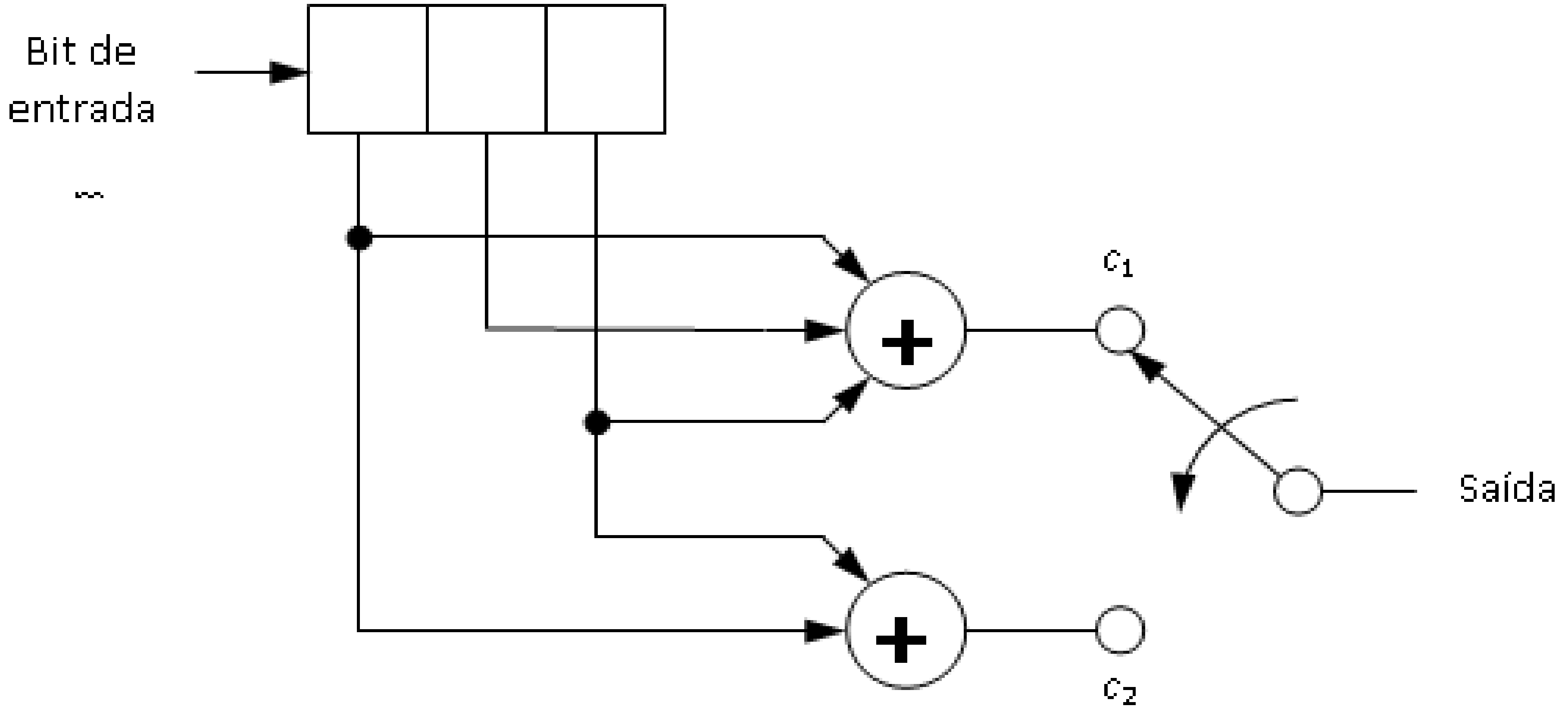
4.2. REPRESENTAÇÕES E CARACTERÍSTICAS BÁSICAS DO CODIFICADOR

- Para descrever um código convolucional é necessário caracterizar a função de codificação $G(\mathbf{m})$ de tal forma que, dada uma sequência de entrada **m**, seja possível determinar a saída codificada **c**.

4.2.1. REPRESENTAÇÃO PICTORIAL

- Considere o codificador convolucional onde a cada instante de tempo, um bit de entrada ocupa a posição mais a esquerda do registrador de deslocamento (k = 1). Como o codificador possui dois somadores módulo-2, para cada bit que entra no codificador, dois bits códigos são gerados na saída pela amostragem dos resultados dos dois somadores. Logo este codificador gera um código de taxa k/n = 1/2, com profundidade K = 3.

Codificador convolucional (taxa 1/2, K = 3)



4.2. REPRESENTAÇÕES E CARACTERÍSTICAS BÁSICAS DO CODIFICADOR

4.2.2. PRINCÍPIO DE OPERAÇÃO

- Suponha que a mensagem $\mathbf{m}(X) = 1 + X + X^3$ deve ser codificada pelo codificador. O processo de codificação é mostrado passo a passo nas figuras e as entradas, registro de todos os estados dos registradores e saída à cada deslocamento são apresentados na Tabela1.

Tabela 1 - Codificação da mensagem $\mathbf{m}(X) = 1 + X + X^3$

DESLOCAMENTO	ENTRADA	CONTEÚDO DOS REGISTRADORES			SAÍDA c_1 c_2
0	1	0	0	0	-
1	0	1	0	0	11
2	1	0	1	0	10
3	1	1	0	1	00
4	0	1	1	0	01
5	0	0	1	1	01
6	0	0	0	1	11
7	-	0	0	0	00

Saída: 11 10 00 01 01 11.

4.2.2. PRINCÍPIO DE OPERAÇÃO

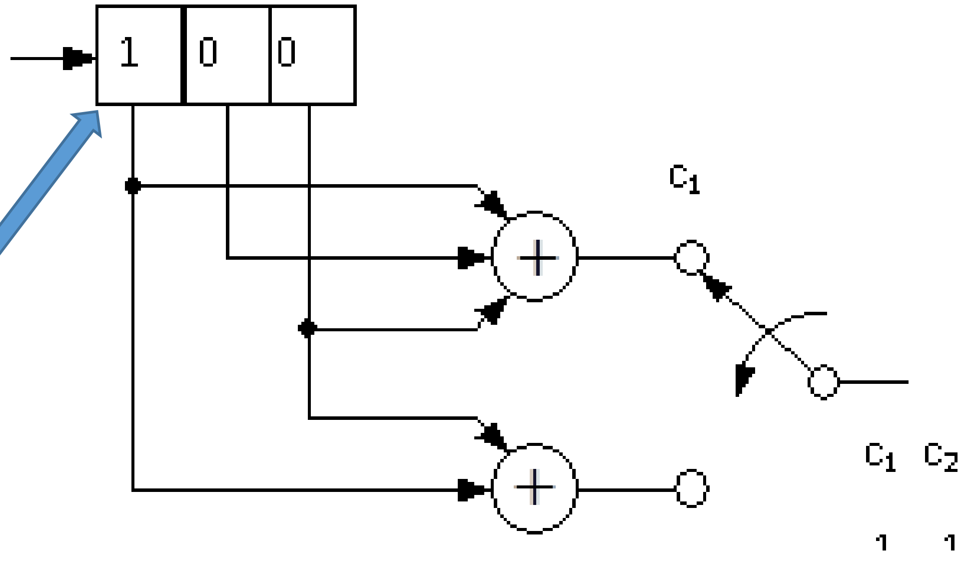
$$\mathbf{m}(X) = 1 + X + X^3$$

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

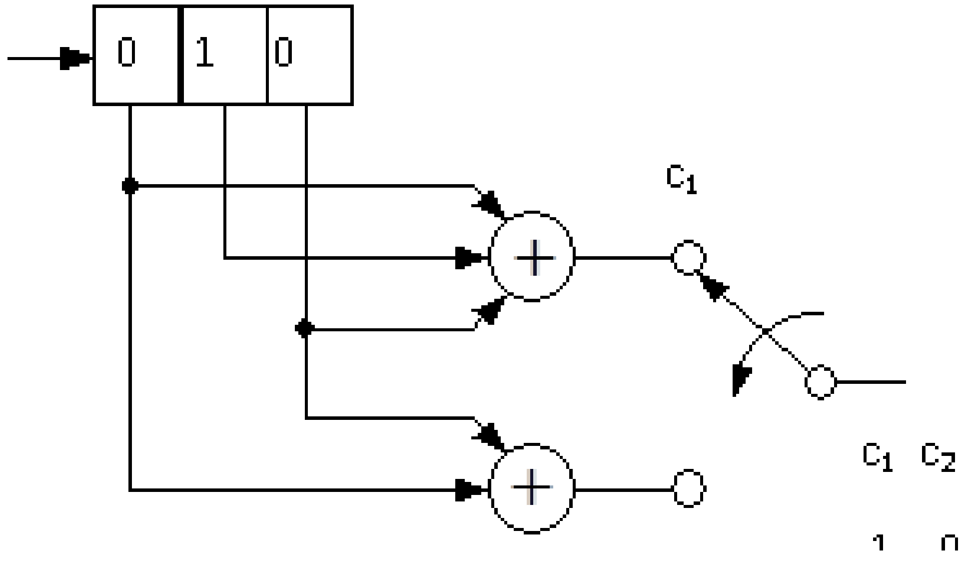
$$\text{m} \quad = 1 \quad 1 \quad 0 \quad 1$$

- Para o codificador pode-se escrever os vetores conexão $\mathbf{g}_1$ e $\mathbf{g}_2$ para os dois somadores como

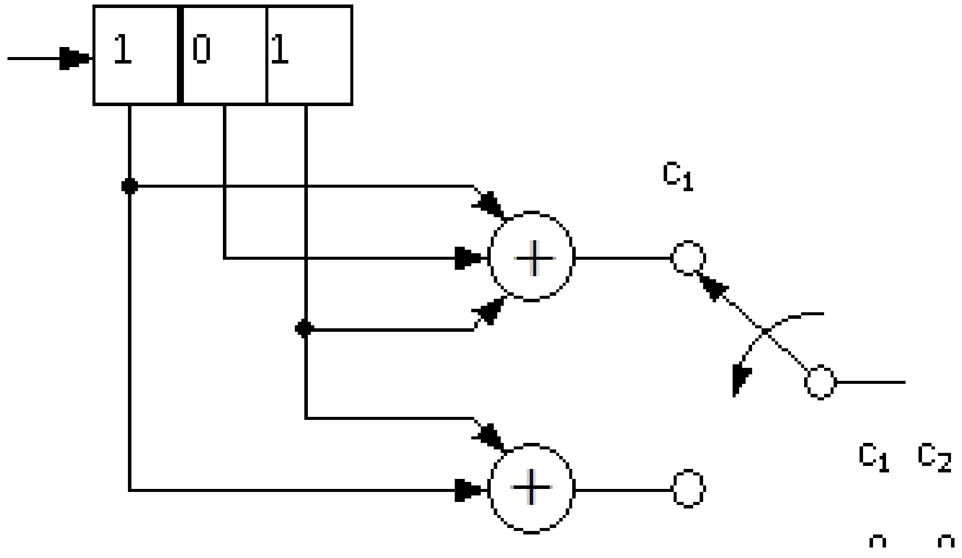
$$\mathbf{g}_1 = 111$$
$$\mathbf{g}_2 = 101$$



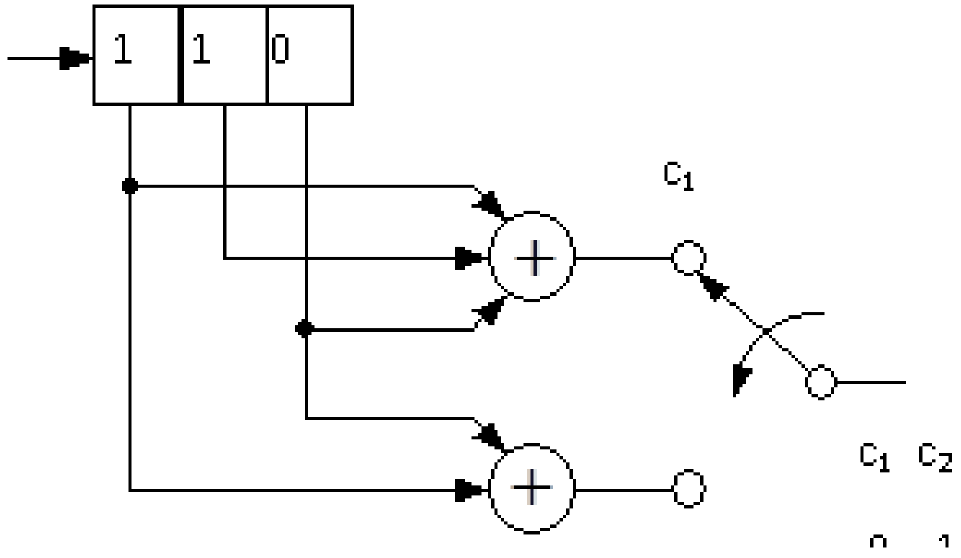
Deslocamento 1



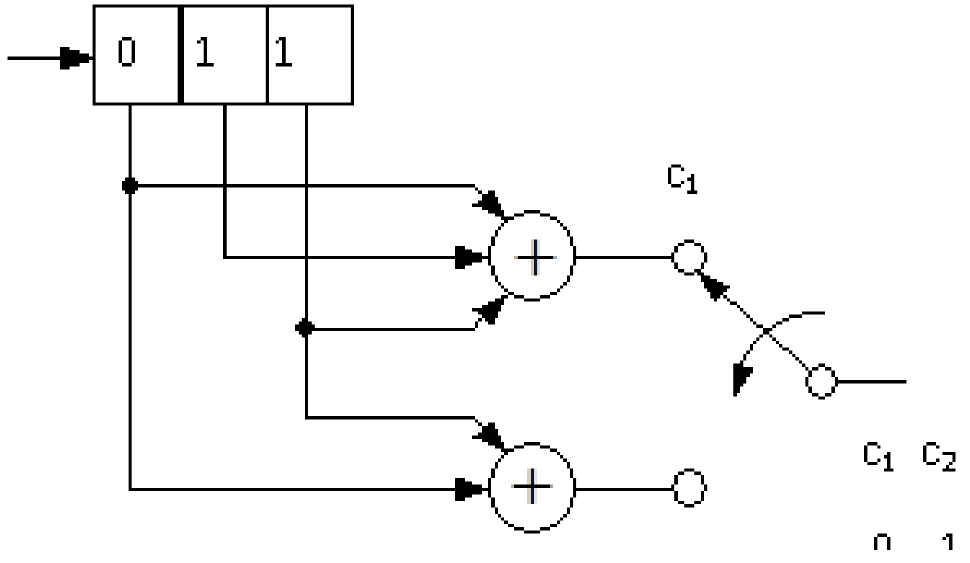
Deslocamento 2



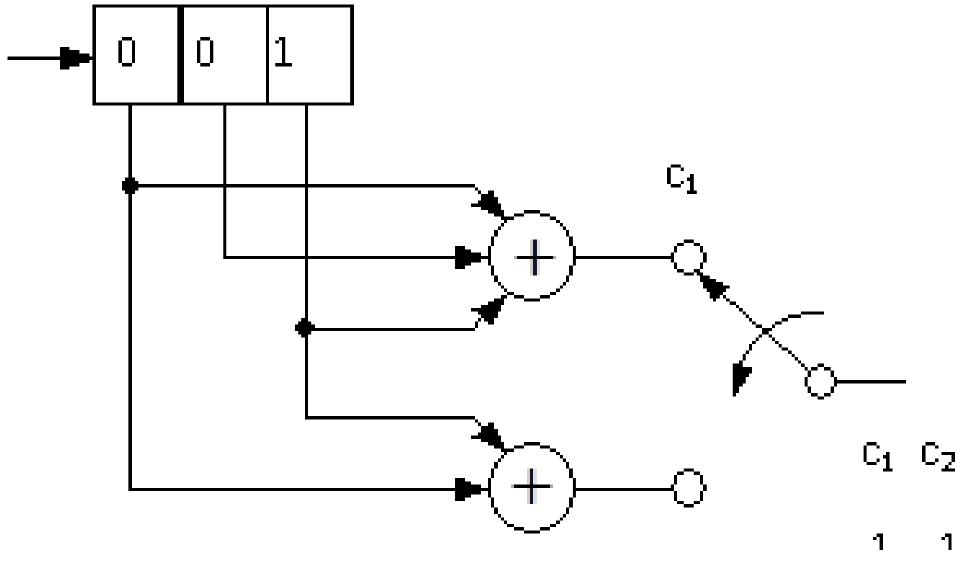
Deslocamento 3



Deslocamento 4



Deslocamento 5



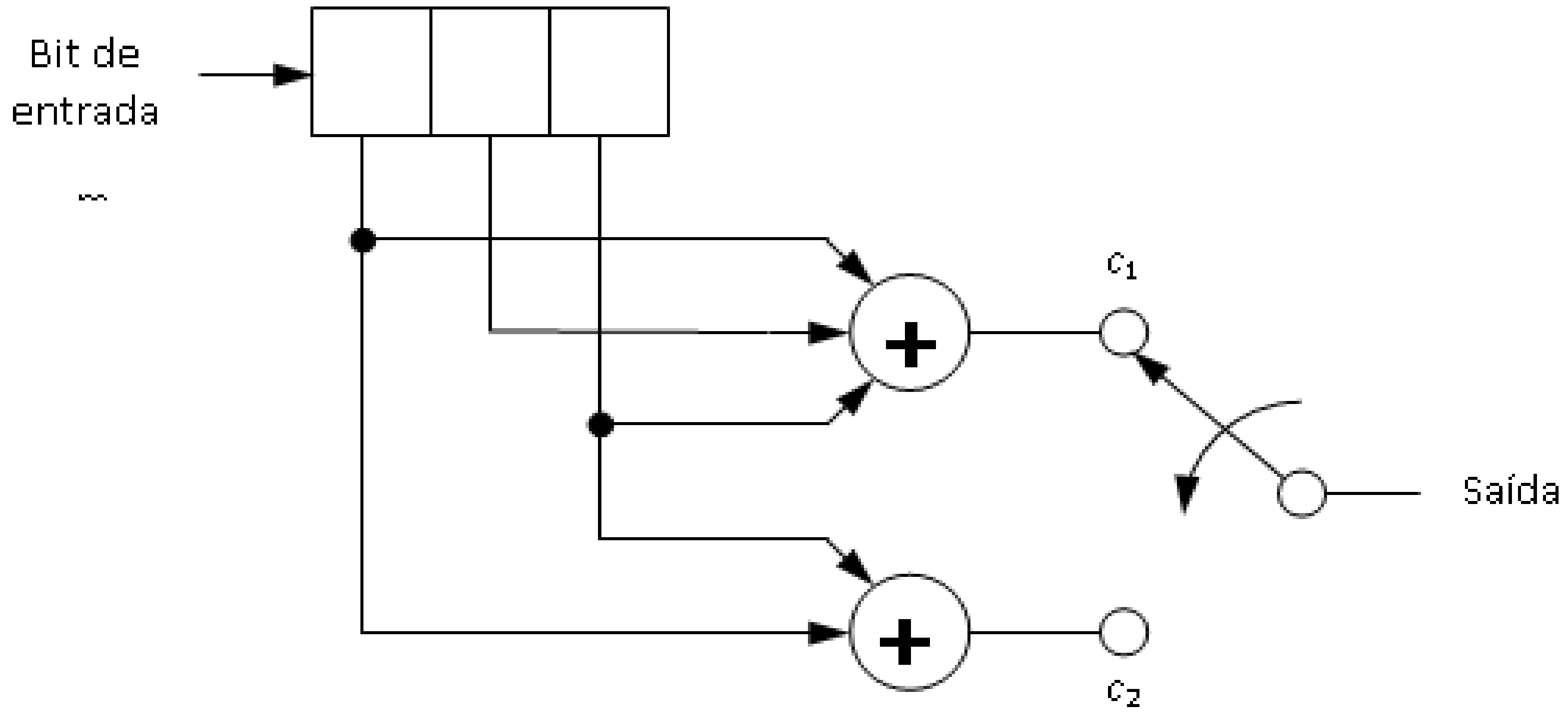
Deslocamento 6

4.2. REPRESENTAÇÕES E CARACTERÍSTICAS BÁSICAS DO CODIFICADOR

4.2.3. RESPOSTA AO IMPULSO DO CODIFICADOR

- Pode-se determinar a *resposta ao impulso* de um codificador convolucional, fazendo um único "1" atravessá-lo desde o primeiro estágio do registrador de deslocamento até o último.

- Para o codificador da Figura, a resposta ao impulso pode ser obtida de acordo com a Tabela abaixo.

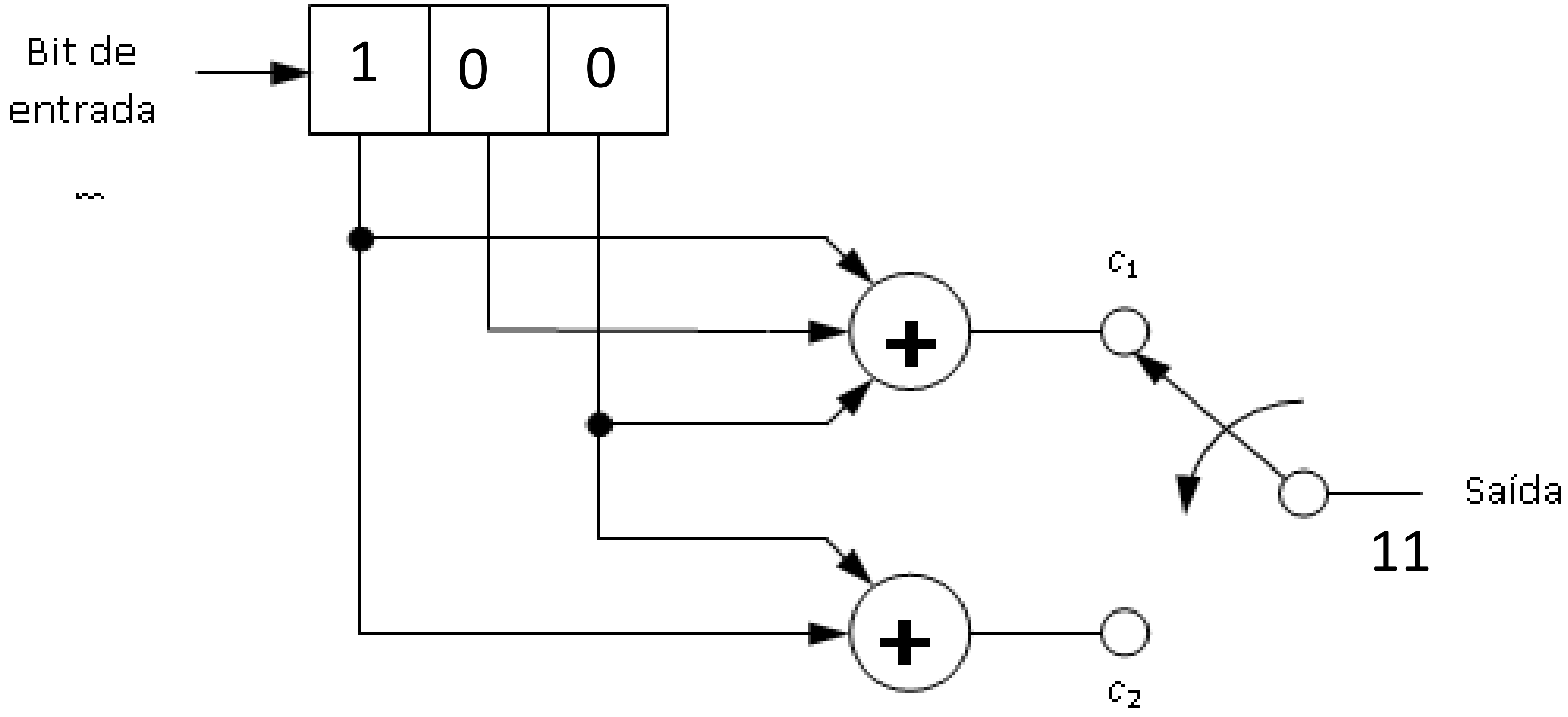


DESLOCAMENTO	CONTEÚDO DOS REGISTRADORES			SAÍDA
1	1	0	0	11
2	0	1	0	10
3	0	0	1	11

- Resposta ao impulso: 11 10 11.

4.2. REPRESENTAÇÕES E CARACTERÍSTICAS BÁSICAS DO CODIFICADOR

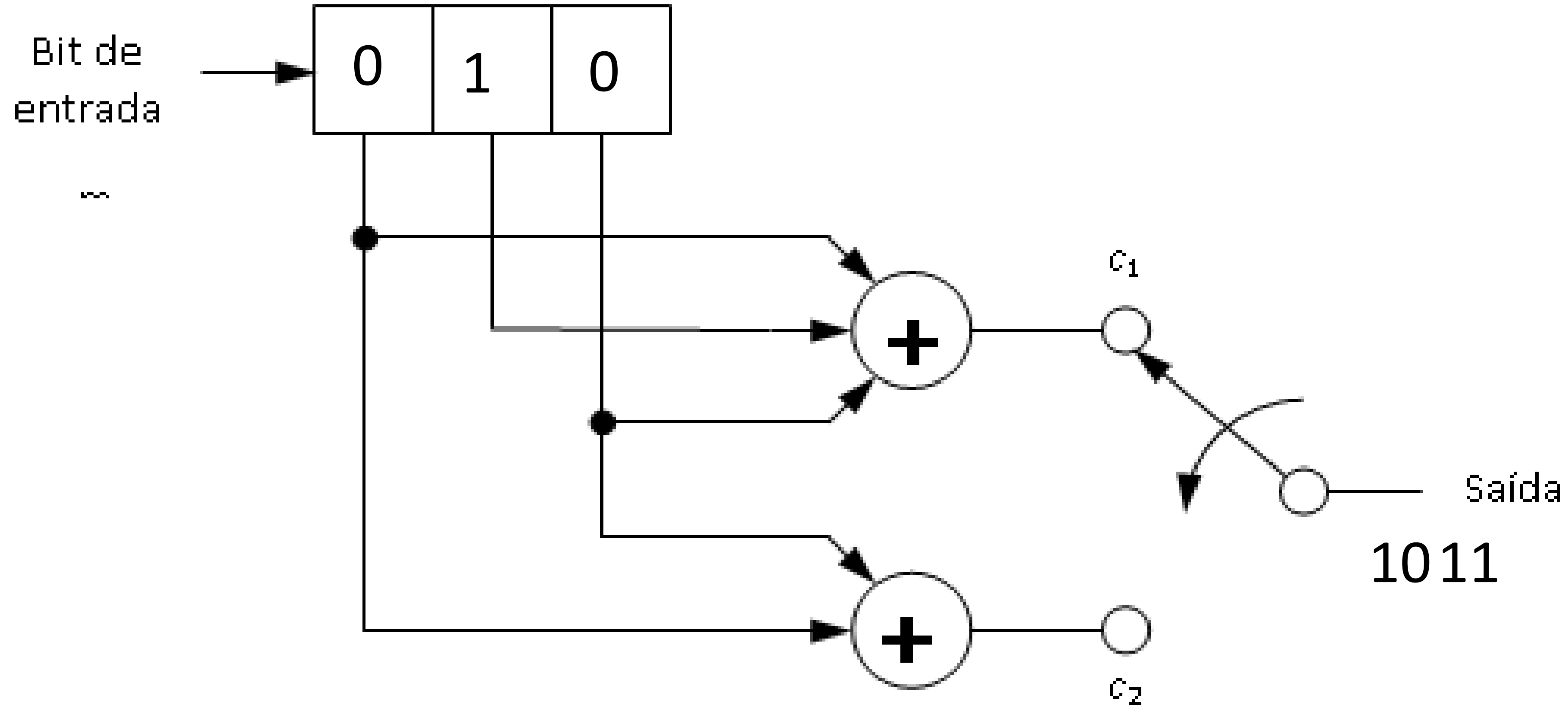
4.2.3. RESPOSTA AO IMPULSO DO CODIFICADOR



DESLOCAMENTO	CONTEÚDO DOS REGISTRADORES			SAÍDA
1	1	0	0	11

4.2. REPRESENTAÇÕES E CARACTERÍSTICAS BÁSICAS DO CODIFICADOR

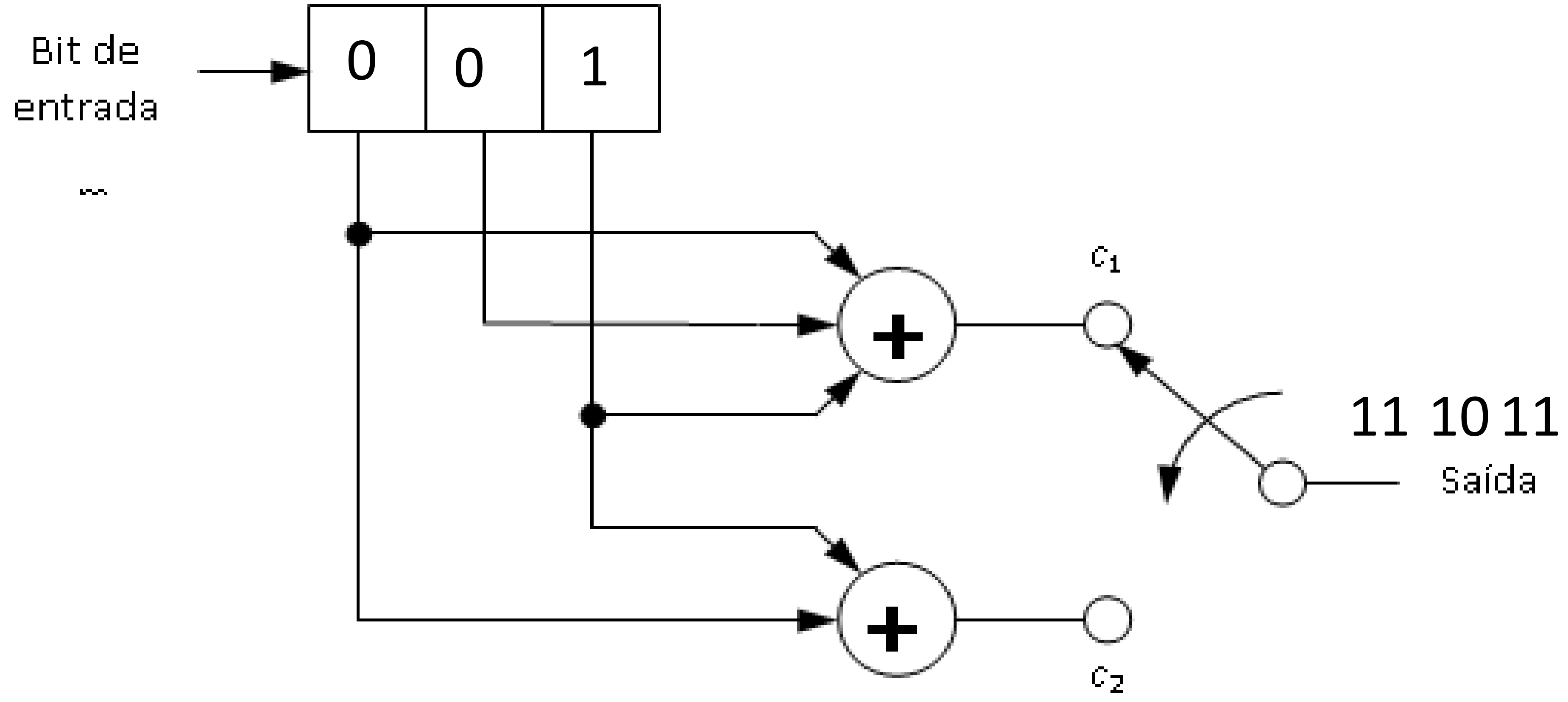
4.2.3. RESPOSTA AO IMPULSO DO CODIFICADOR



DESLOCAMENTO	CONTEÚDO DOS REGISTRADORES			SAÍDA
1	1	0	0	11
2	0	1	0	10

4.2. REPRESENTAÇÕES E CARACTERÍSTICAS BÁSICAS DO CODIFICADOR

4.2.3. RESPOSTA AO IMPULSO DO CODIFICADOR



DESLOCAMENTO	CONTEÚDO DOS REGISTRADORES			SAÍDA
1	1	0	0	11
2	0	1	0	10
3	0	0	1	11

4.2. REPRESENTAÇÕES E CARACTERÍSTICAS BÁSICAS DO CODIFICADOR

4.2.3. RESPOSTA AO IMPULSO DO CODIFICADOR

- Note que a resposta ao impulso pode ser obtida diretamente dos vetores conexão, conforme mostrado a seguir.

g_1 =	1	1	1
g_2 =	1	0	1
	11	10	11

- A resposta ao impulso permite a determinação da saída do codificador para uma dada entrada **m**, pela superposição ou adição linear das respostas ao impulso deslocadas no tempo, ou ainda, pela convolução da sequência de entrada com a resposta ao impulso do codificador.
- Para a mensagem **m** = 1 1 0 1, a saída do codificador obtida a partir da resposta ao impulso é obtida da forma como se segue.

Entrada m	Saída					
1	1 1	1 0	1 1			
0		0 0	0 0	0 0		
1			1 1	1 0	1 1	
1				1 1	1 0	1 1
Soma módulo-2	1 1	1 0	0 0	0 1	0 1	1 1

4.2. REPRESENTAÇÕES E CARACTERÍSTICAS BÁSICAS DO CODIFICADOR

4.2.4. DIAGRAMA DE ESTADOS

- Um codificador convolucional pertence a uma classe de dispositivos conhecidos como *máquinas de estado finito*, que é o nome genérico para máquinas que tem memória dos sinais passados.

- O termo *finito* refere-se ao fato de que existe apenas um número de estados único e finito que a máquina pode gerar.

- Em um sentido amplo, um *estado* consiste de uma pequena quantidade de informação que, junto com a informação presente na entrada da máquina, é capaz de determinar a saída.

- Os estados fornecem algum conhecimento de eventos passados e o restrito conjunto de possíveis saídas no futuro.

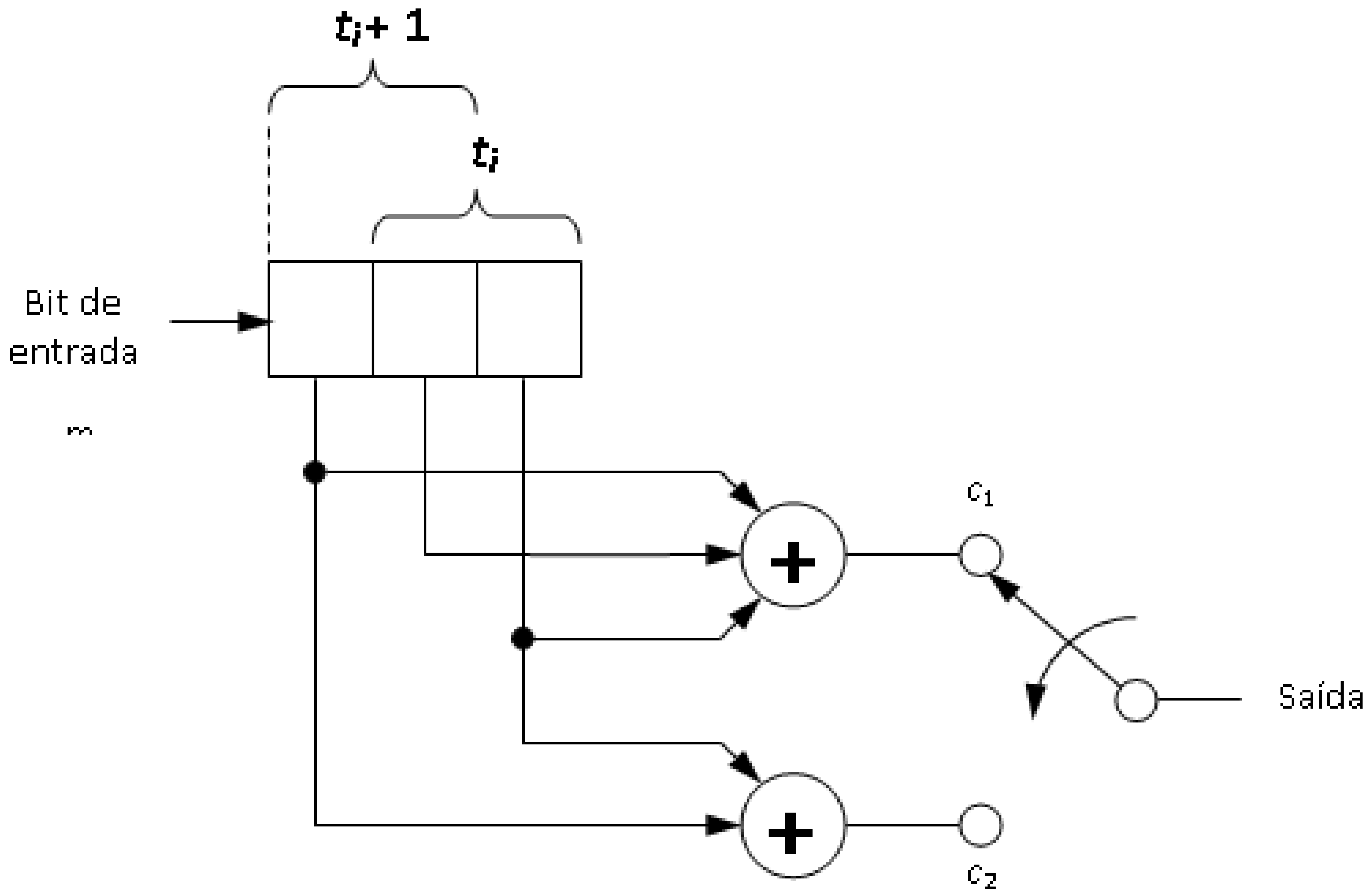
- Um estado futuro fica restringido por um estado passado. Para um codificador convolucional com taxa $1/n$, o estado atual dado pelo tempo t_i é representado pelo conteúdo dos $(K – 1)$ estágios mais a direita, conforme mostrado na Figura a seguir. O estado representado pelo tempo $t_i + 1$ é o estado futuro.

- O conhecimento do estado e da próxima entrada é necessário e suficiente para determinar a próxima saída.

4.2. REPRESENTAÇÕES E CARACTERÍSTICAS BÁSICAS DO CODIFICADOR

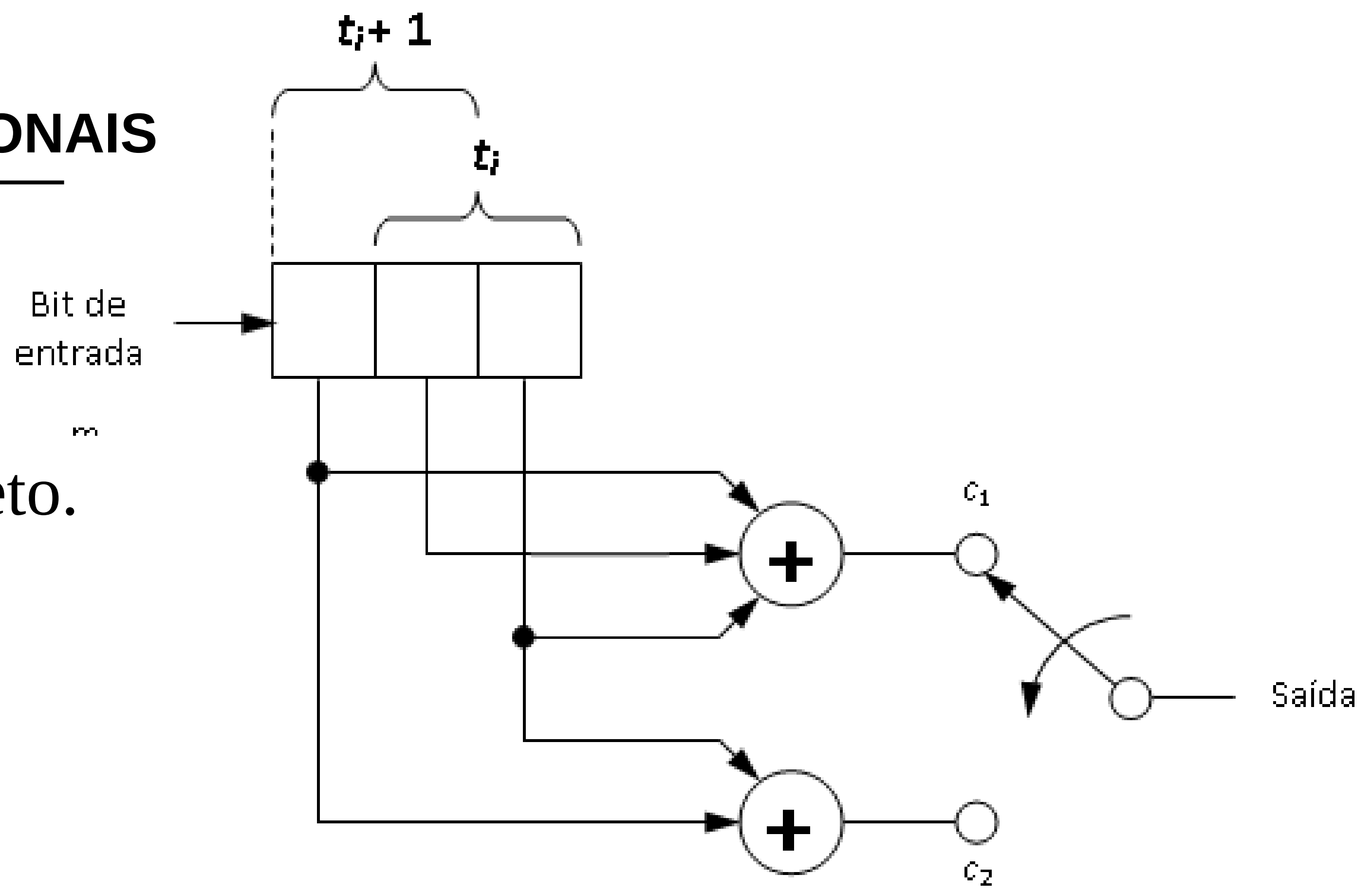
4.2.4. DIAGRAMA DE ESTADOS

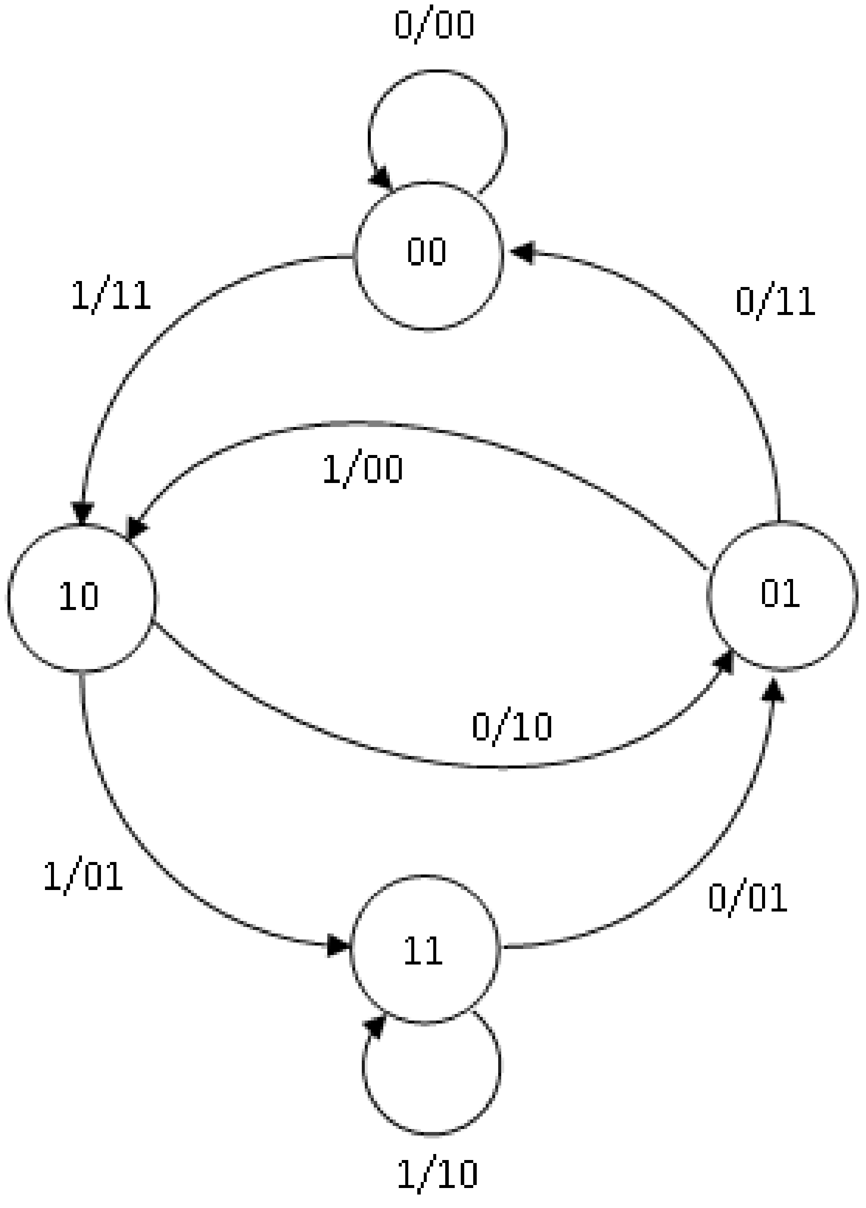
- Definição do estado atual (t_i) e estado futuro (t_i + 1) para um codificador com K = 3.



4.2.4. DIAGRAMA DE ESTADOS

- A Figura apresenta o diagrama de estados completo.





Tabela

Entrada m_i	Conteúdos dos registradores	Estado em t_i	Estado em $t_i + 1$	Saída em t_i c_1 c_2
0	000	00	00	00
1	100	00	10	11
0	010	10	01	10
1	110	10	11	01
0	011	11	01	01
1	111	11	11	10
0	001	01	00	11
1	101	01	10	00

4.2.5. DIAGRAMA DE TRELIÇA

- No diagrama de treliça tem-se um histórico de entradas, transições e saídas. A Figura a seguir apresenta o diagrama de treliça para o codificador apresentado anteriormente. Neste diagrama, os estados são representados pelos níveis horizontais e as entradas e saídas são representadas pela mesma convenção utilizada no diagrama de estados, ou seja, m_i/u_1u_2, que são colocados sobre cada braço da treliça, que por sua vez, representa uma transição.

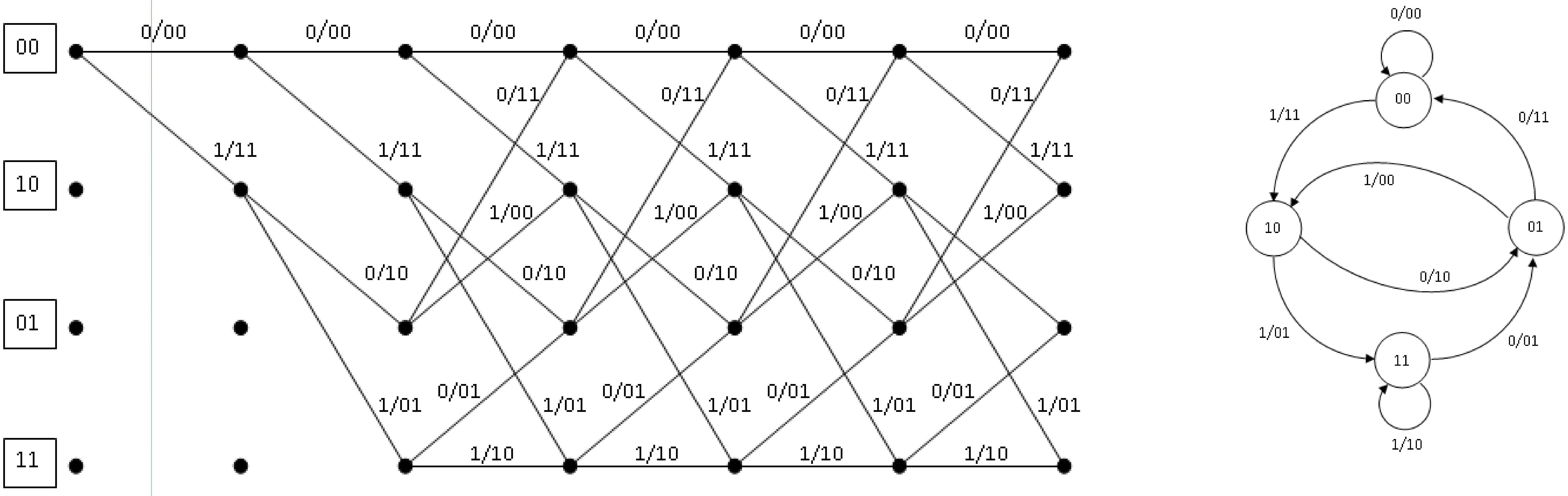
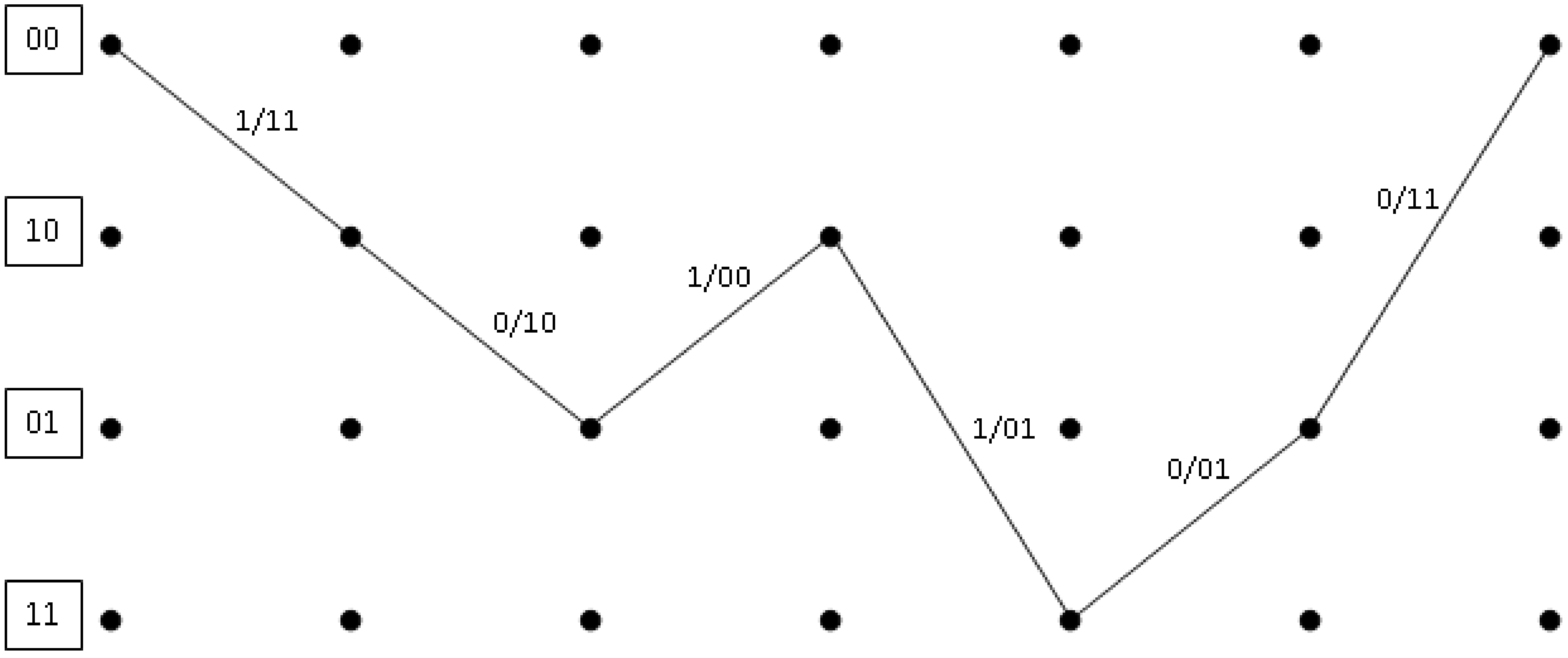


Diagrama de treliça para o codificador convolucional.

4.2.5. DIAGRAMA DE TRELIÇA

- A mensagem m = 1101 estabelece, no diagrama de treliça, a trajetória mostrada na Figura a seguir, resultando, como era de se esperar, nas saídas 11 10 00 01. Note que para esvaziar os registradores do codificador, mais dois zeros na entrada são necessários, resultando em uma saída complementar igual a 01 11. Assim, a sequência de saída completa para a mensagem m = 1101 torna-se 11 10 00 01 01 11. Este resultado é rigorosamente igual aos resultados apresentados anteriormente.



Trajetória para a mensagem m = 1101.

4.3. DECODIFICAÇÃO DE CÓDIGOS CONVOLUCIONAIS PELO ALGORITMO DE VITERBI

- O algoritmo de Viterbi é um algoritmo de máxima verossimilhança com baixa carga computacional em função da utilização da estrutura dos diagramas de treliça dos códigos convolucionais.

- A vantagem da decodificação de Viterbi, comparado com a decodificação por força bruta, é que a complexidade de um decodificador não é função do número de símbolos da sequência código, mas função de uma medida de *similaridade* ou *distância* entre o sinal recebido em um tempo t_i e todos os braços da treliça que entram em cada estado no tempo t_i.

- Quando dois braços entram no mesmo estado em um tempo t_i, o que possuir melhor métrica ou maior semelhança com o sinal recebido é escolhido e chamado de *caminho sobrevivente*.

- Existem, basicamente, duas distâncias que podem ser utilizadas no algoritmo de Viterbi para a medida de similaridade entre a sequência recebida pelo decodificador e as sequências possíveis sobre a treliça:

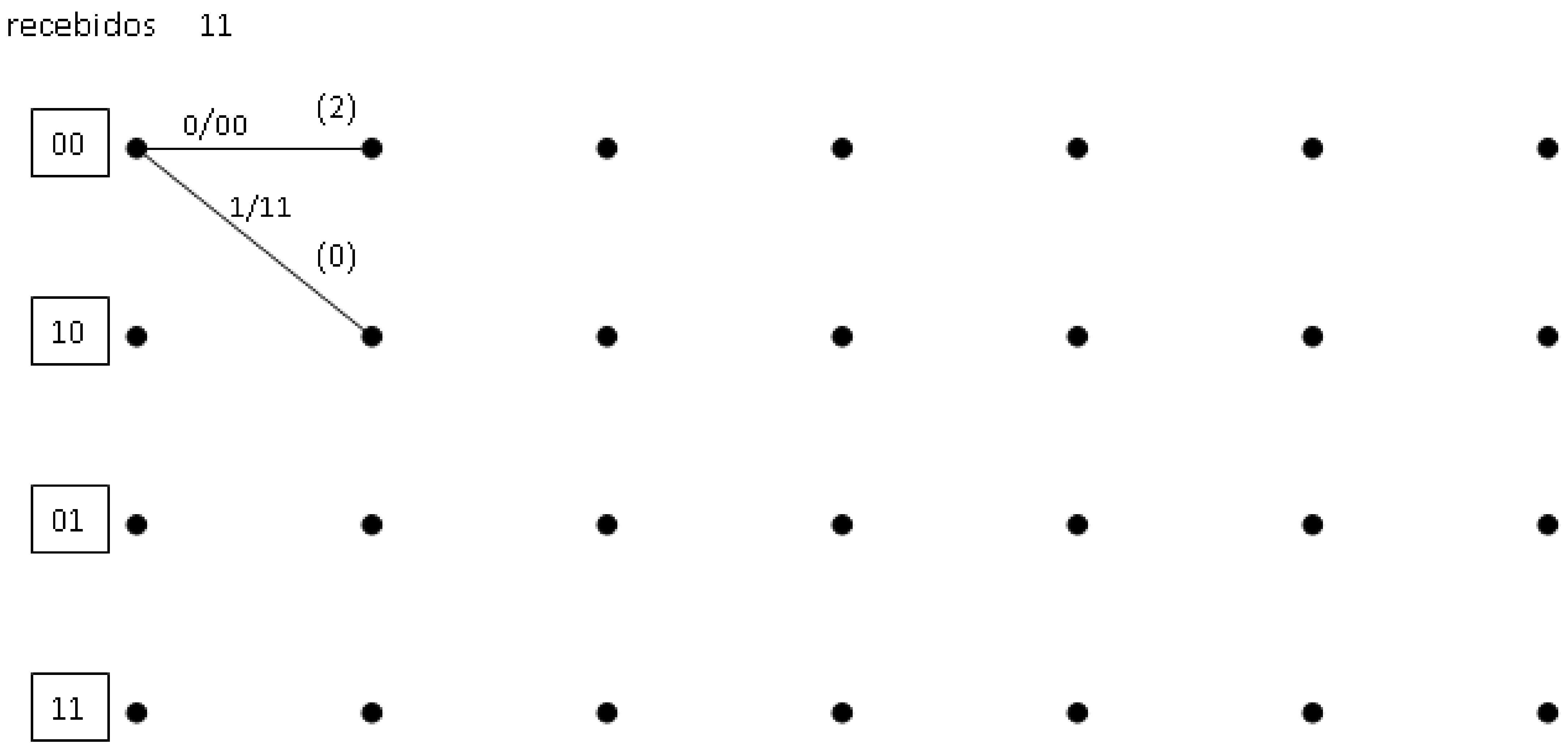
A distância de Hamming e a distância Euclidiana.

  - A distância de Hamming é utilizada em um processo de decisão chamado de *decisão abrupta* (*hard decision*)
  - A distância Euclidiana é utilizada em um processo de decisão chamado de *decisão suave* (*soft decision*)
  - Neste curso é abordado a decisão abrupta pela simplicidade e baixa carga computacional.

4.3. DECODIFICAÇÃO DE CÓDIGOS CONVOLUCIONAIS PELO ALGORITMO DE VITERBI

4.3.1. DECODIFICAÇÃO ABRUPTA PELO ALGORITMO DE VITERBI

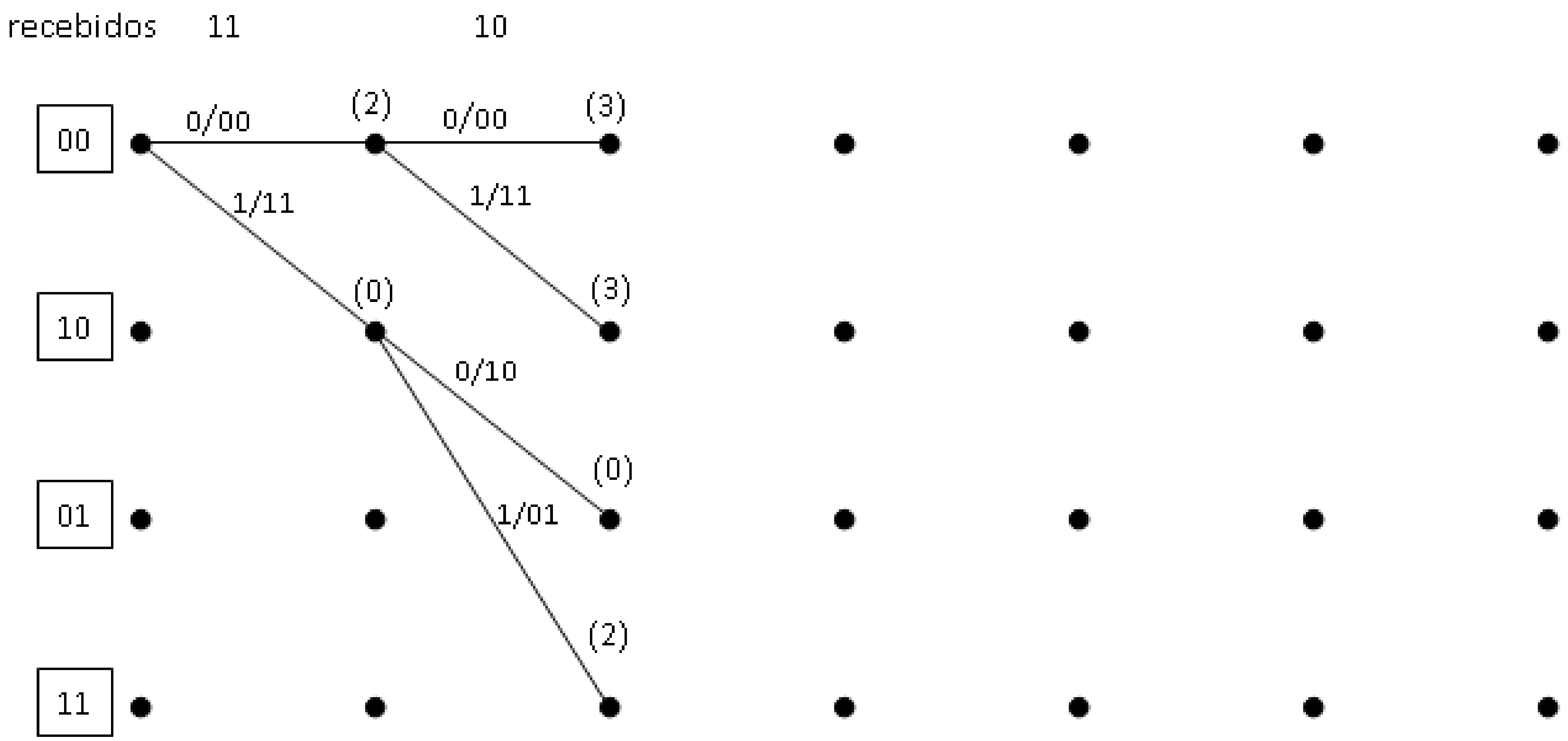
- Considere a sequência codificada e recebida 11 10 10 01 01 11. Note que foi introduzido um erro no terceiro par de bits.
- Passo 1: Partindo do estado 00, compara-se a distância de Hamming do primeiro par de bits da sequência código (11) com as distâncias de Hamming das duas transições possíveis a partir do estado 00. As distâncias de Hamming obtidas são armazenadas (valores entre parênteses), conforme apresentado na Figura.



4.3. DECODIFICAÇÃO DE CÓDIGOS CONVOLUCIONAIS PELO ALGORITMO DE VITERBI

4.3.1. DECODIFICAÇÃO ABRUPTA PELO ALGORITMO DE VITERBI

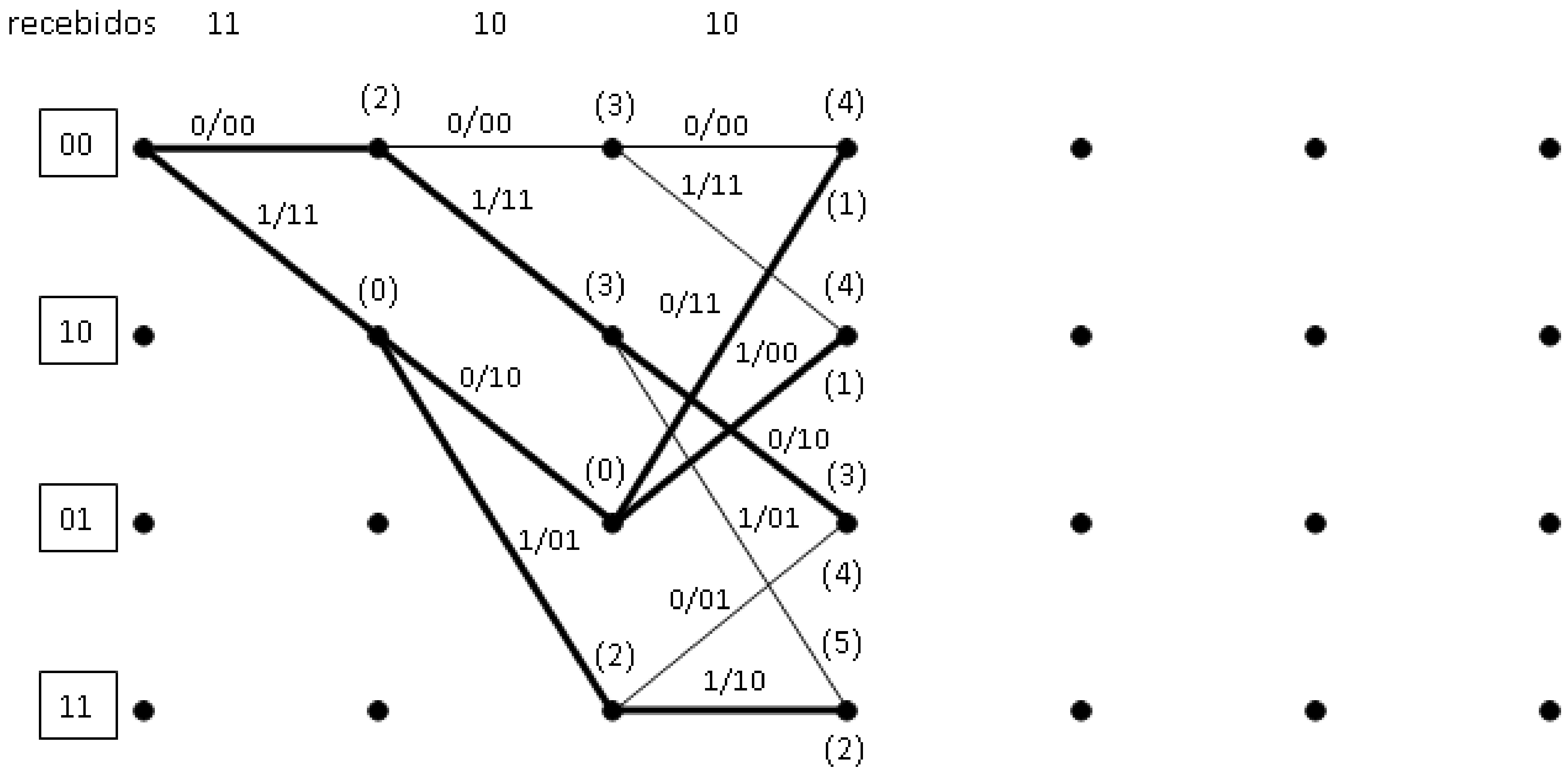
- Passo 2: A distância de Hamming do próximo par de bits da sequência código (10) é comparada com as distâncias de Hamming das transições possíveis a partir dos estados alcançados após o passo anterior. As distâncias de Hamming encontradas são somadas àquelas obtidas nas transições anteriores. Veja Figura.



4.3. DECODIFICAÇÃO DE CÓDIGOS CONVOLUCIONAIS PELO ALGORITMO DE VITERBI

4.3.1. DECODIFICAÇÃO ABRUPTA PELO ALGORITMO DE VITERBI

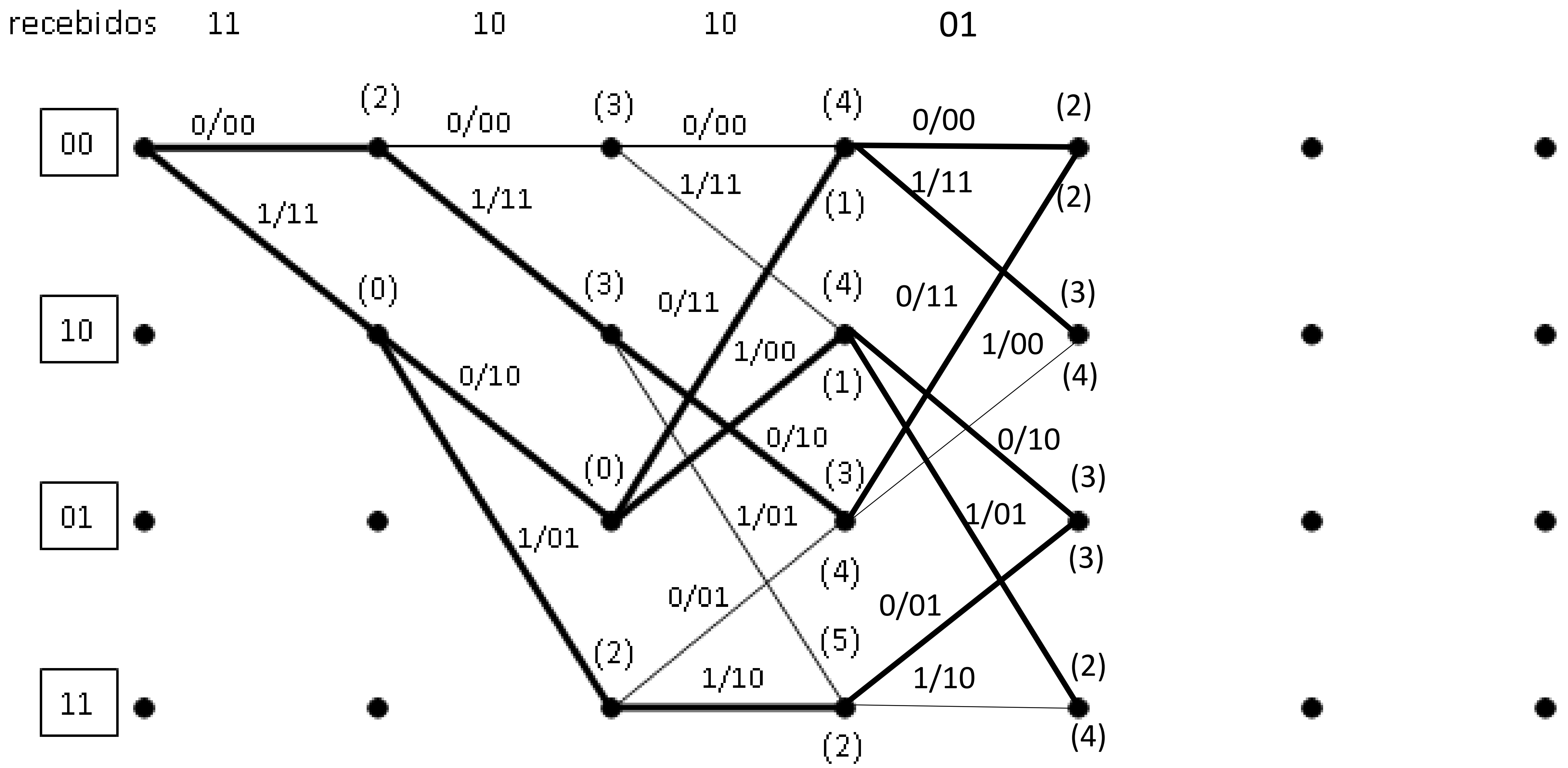
- Passo 3: O mesmo procedimento apresentado no Passo 2 é repetido para o próximo par da sequência código (10). Note que neste terceiro passo alguns dos estados são alcançados por dois caminhos diferentes. O caminho sobrevivente deve ser aquele que apresenta a menor distância de Hamming acumulada. Na Figura, os caminhos sobreviventes estão representados por uma linha mais espessa.



4.3. DECODIFICAÇÃO DE CÓDIGOS CONVOLUCIONAIS PELO ALGORITMO DE VITERBI

4.3.1. DECODIFICAÇÃO ABRUPTA PELO ALGORITMO DE VITERBI

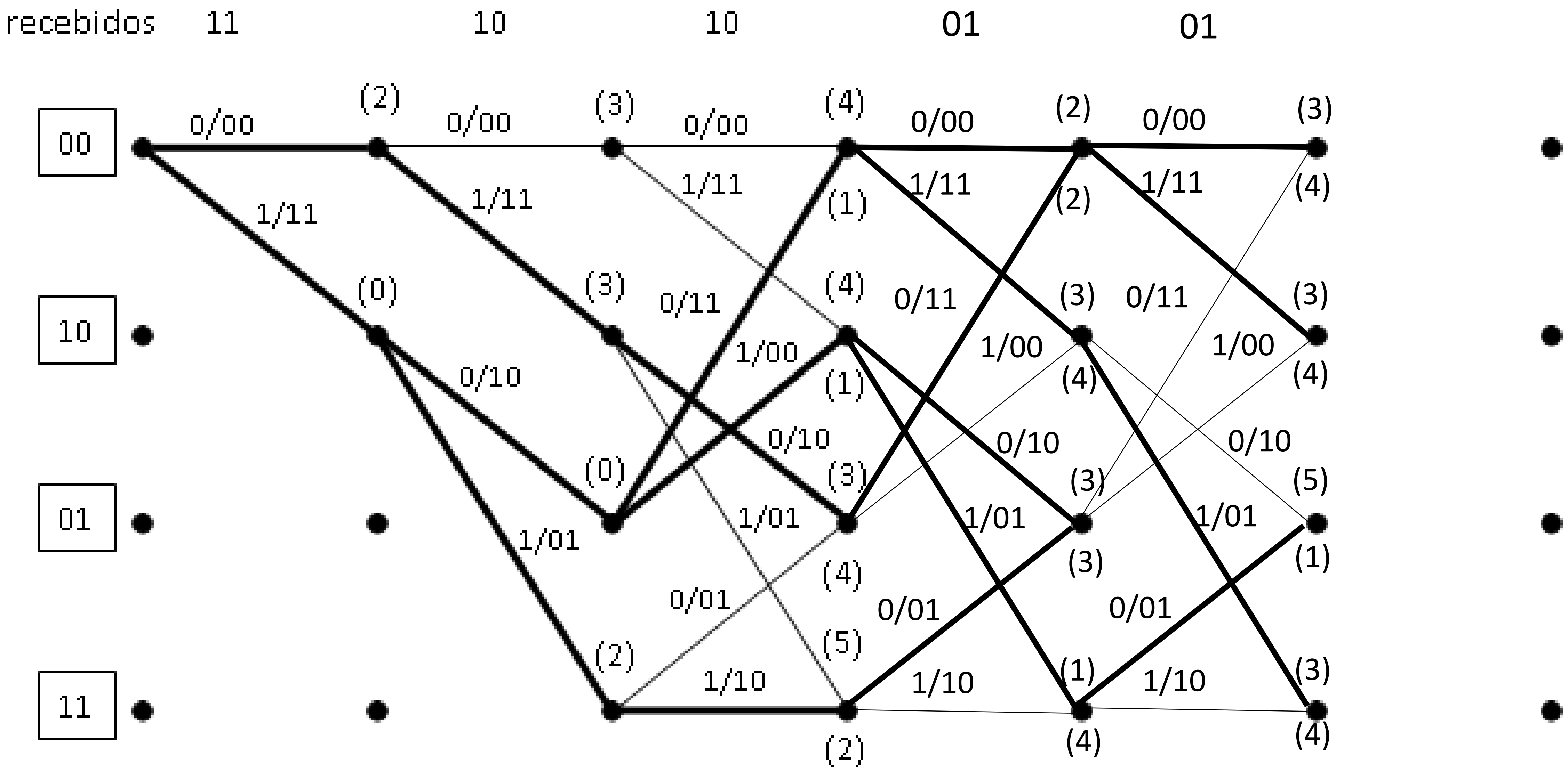
- Passo 4: O mesmo procedimento apresentado no Passo 2 é repetido para o próximo par da sequência código (10). Note que neste terceiro passo alguns dos estados são alcançados por dois caminhos diferentes. O caminho sobrevivente deve ser aquele que apresenta a menor distância de Hamming acumulada. Na Figura, os caminhos sobreviventes estão representados por uma linha mais espessa.



4.3. DECODIFICAÇÃO DE CÓDIGOS CONVOLUCIONAIS PELO ALGORITMO DE VITERBI

4.3.1. DECODIFICAÇÃO ABRUPTA PELO ALGORITMO DE VITERBI

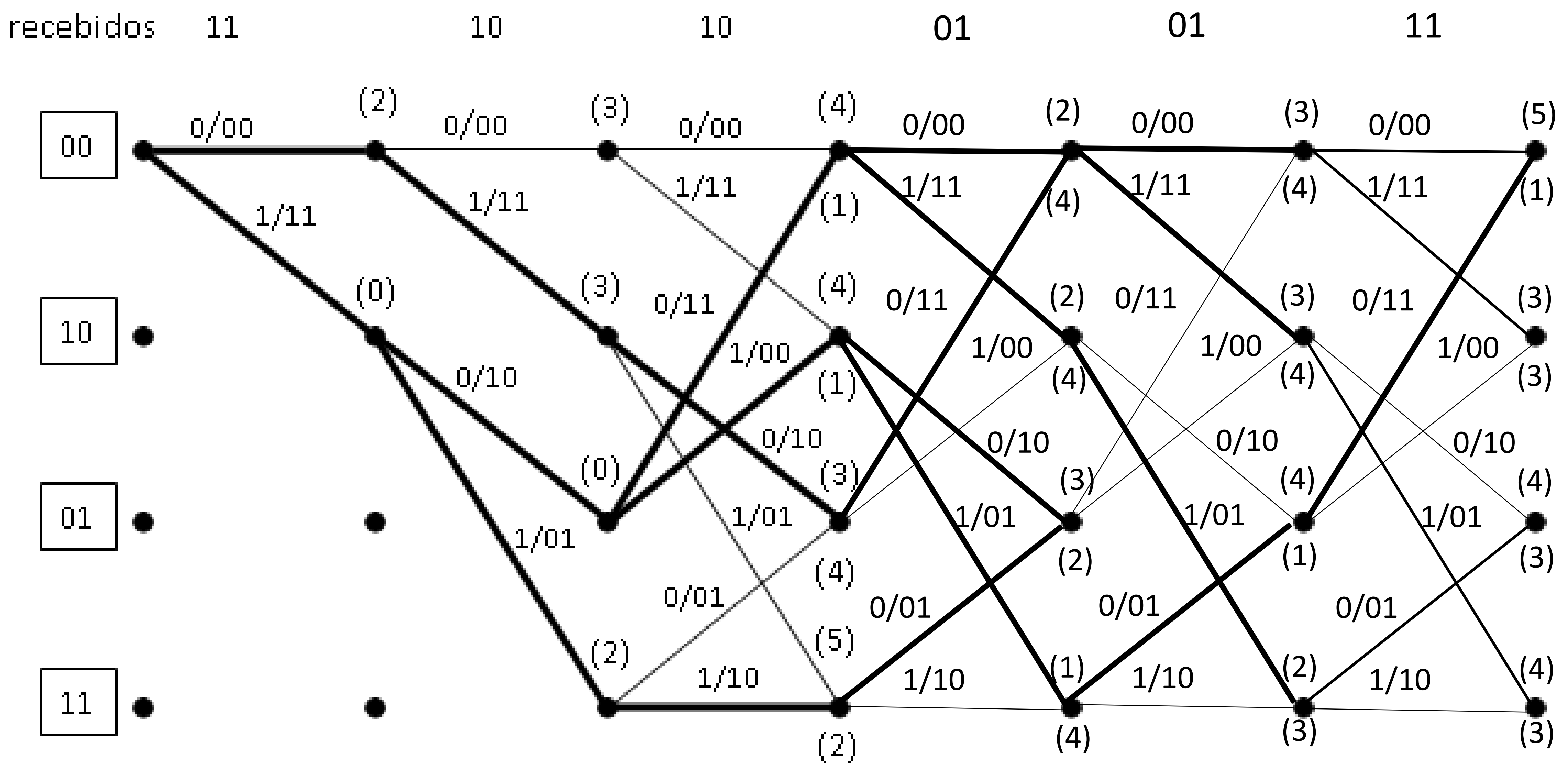
- Passo 5: O mesmo procedimento apresentado no Passo 2 é repetido para o próximo par da sequência código (10). Note que neste terceiro passo alguns dos estados são alcançados por dois caminhos diferentes. O caminho sobrevivente deve ser aquele que apresenta a menor distância de Hamming acumulada. Na Figura, os caminhos sobreviventes estão representados por uma linha mais espessa.



4.3. DECODIFICAÇÃO DE CÓDIGOS CONVOLUCIONAIS PELO ALGORITMO DE VITERBI

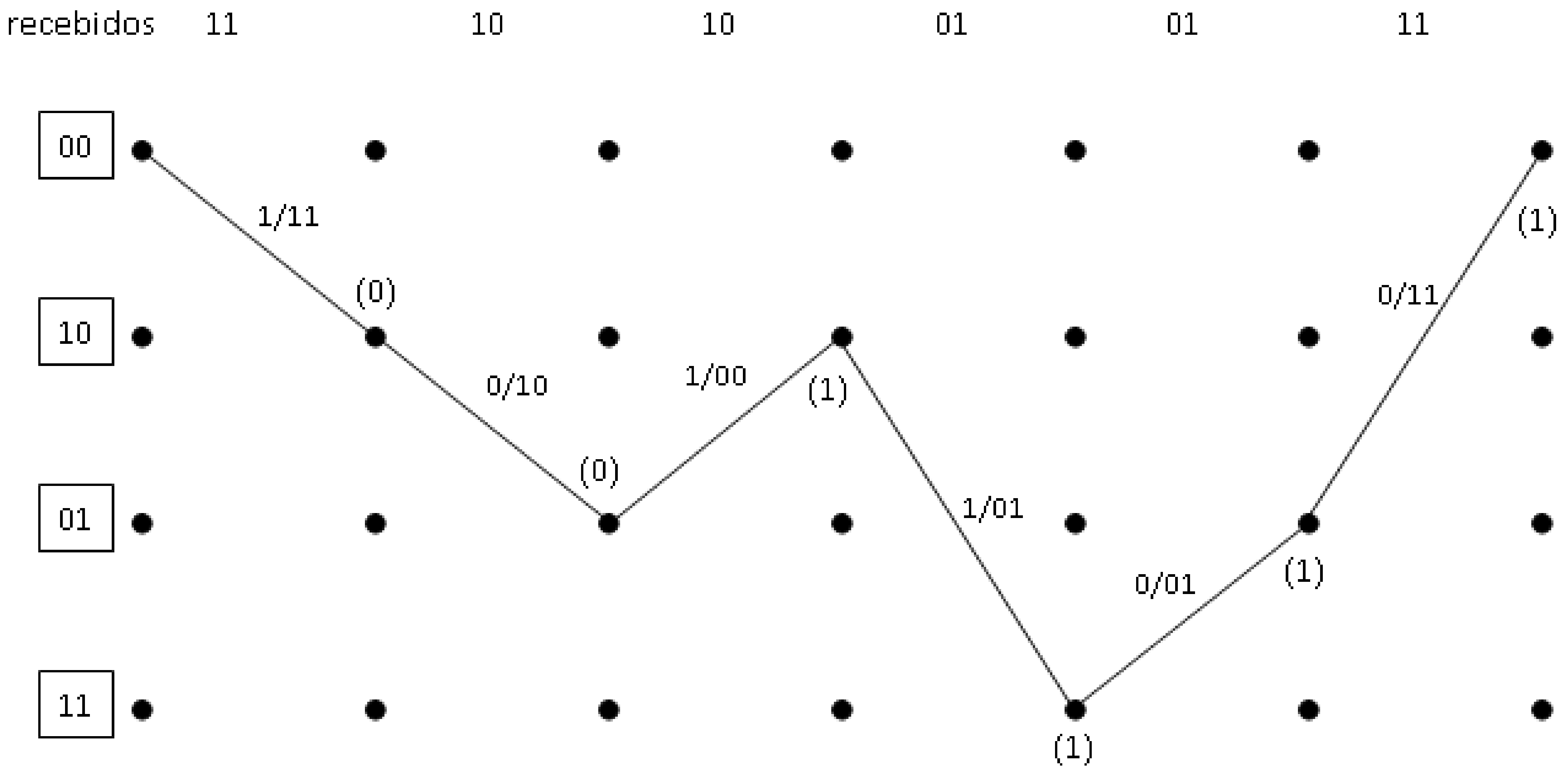
4.3.1. DECODIFICAÇÃO ABRUPTA PELO ALGORITMO DE VITERBI

- Passo 6: O mesmo procedimento apresentado no Passo 2 é repetido para o próximo par da sequência código (10). Note que neste terceiro passo alguns dos estados são alcançados por dois caminhos diferentes. O caminho sobrevivente deve ser aquele que apresenta a menor distância de Hamming acumulada. Na Figura, os caminhos sobreviventes estão representados por uma linha mais espessa.



4.3. DECODIFICAÇÃO DE CÓDIGOS CONVOLUCIONAIS PELO ALGORITMO DE VITERBI

4.3.1. DECODIFICAÇÃO ABRUPTA PELO ALGORITMO DE VITERBI



- Após a análise do último par de bits recebidos, o caminho sobrevivente aponta uma distância de Hamming acumulada igual a 1. Este resultado mostra que a sequência decodificada com maior probabilidade de ter sido a sequência transmitida é a sequência 11 10 00 01 01 11, correspondente a mensagem m = 101100. Neste caso a sequência recebida apresenta um erro em relação à sequência decodificada, correspondente à distância de Hamming acumulada igual a 1.

Obrigado!