ECE5984

Orthogonal Frequency Division Multiplexing and Related Technologies

Fall 2007

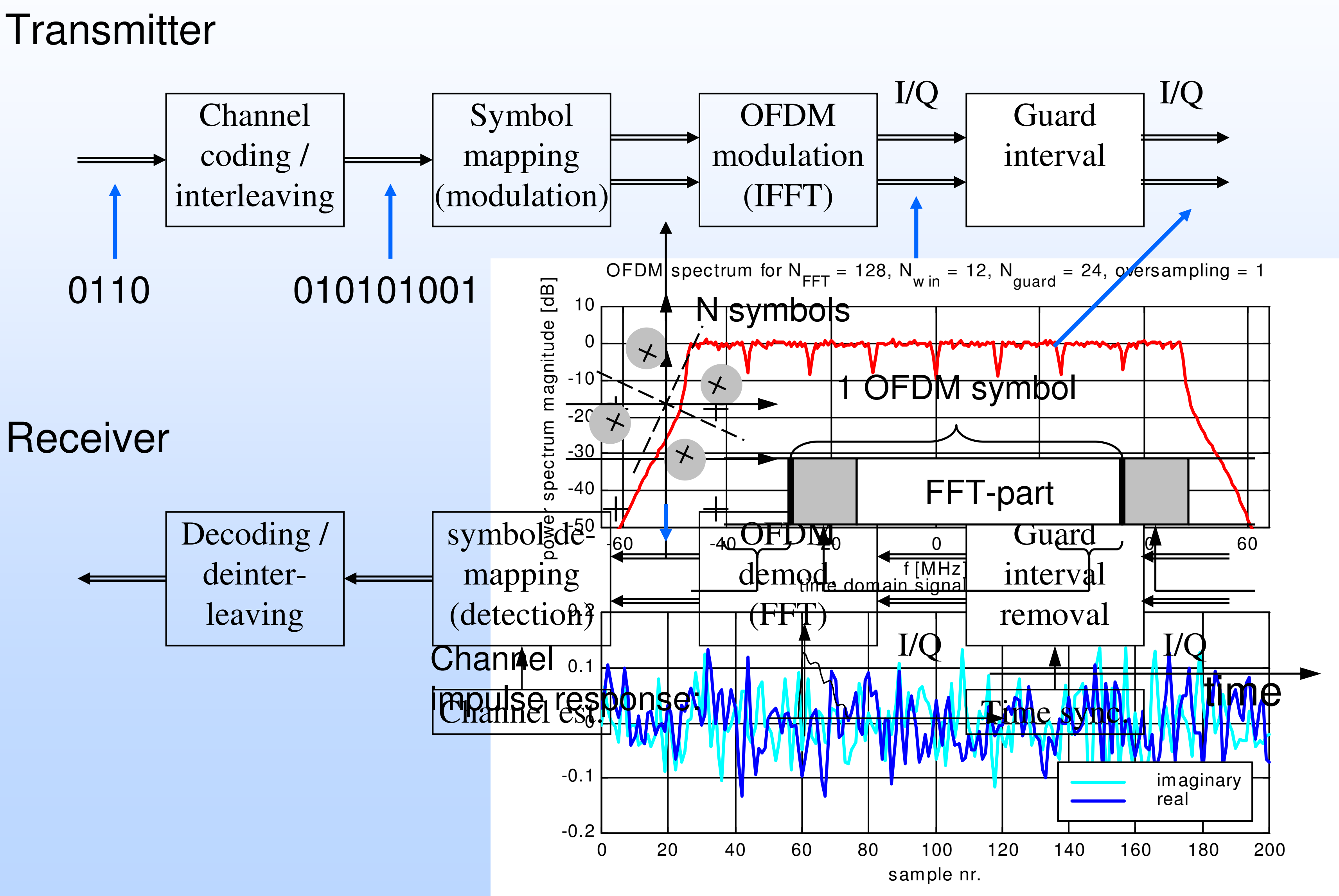
Mohamed Essam Khedr

Coding in OFDM

Syllabus

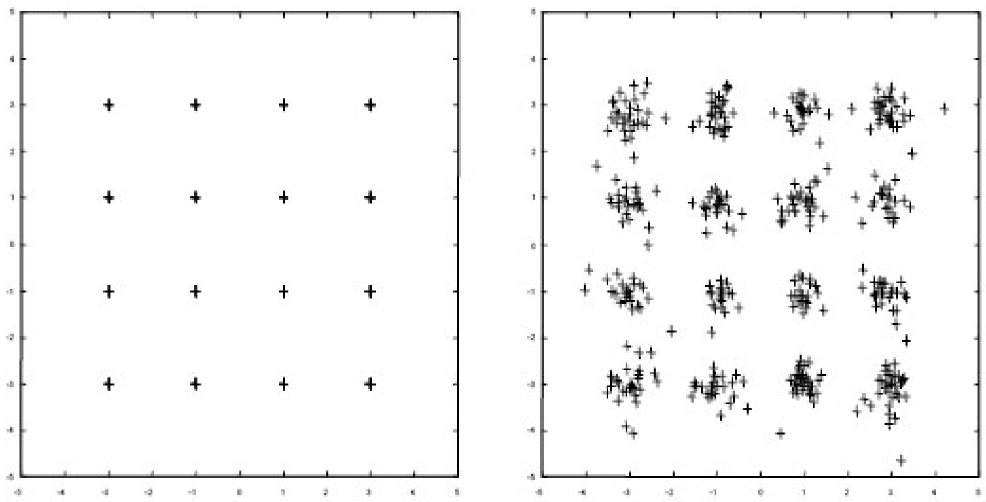
- **Wireless channels characteristics** (7.5%) **1**
- **OFDM Basics** (10%) **1**
- **Modulation and Coding** (10%) **2**
  - Linear and nonlinear modulation
  - Interleavin\g and channel coding
  - Optimal bit and power allocation
  - Adaptive modulation

OFDM Block Diagram



Example of ISI/ICI

(a)



(b)

16-QAM constellations for a 48-subcarrier OFDM signal in a 2-ray multipath channel with

(a) multipath delay < guard time

(b) multipath delay = 1.03*guard time

What is channel coding?

- **Channel coding:**

  Transforming signals to improve communications performance by increasing the robustness against channel impairments (noise, interference, fading, ..)

  - Waveform coding: Transforming waveforms to better waveforms
  - Structured sequences: Transforming data sequences into better sequences, having structured redundancy.

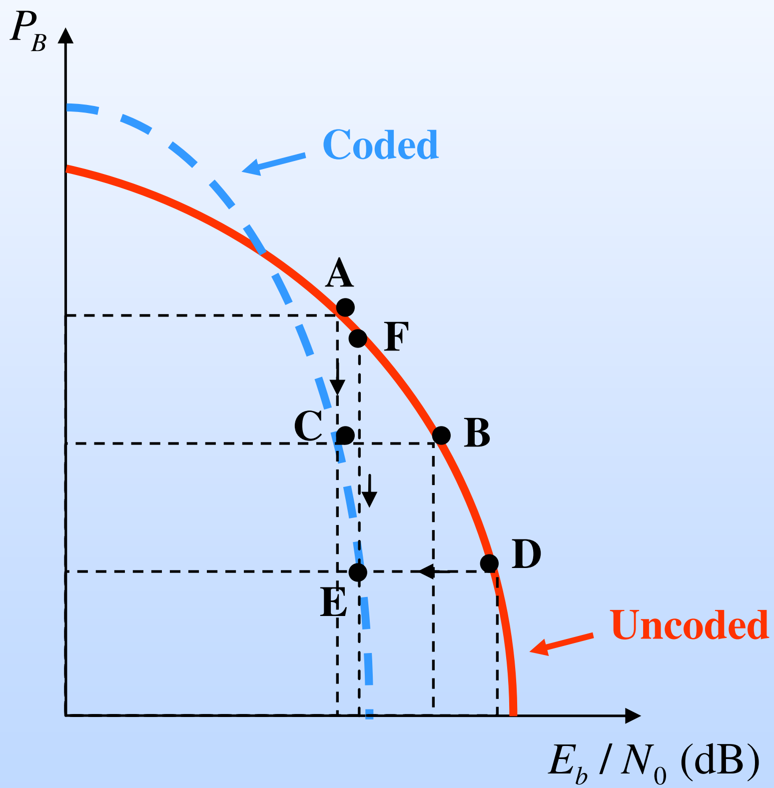
    -“Better” in the sense of making the decision process less subject to errors.

Error control techniques

- **Automatic Repeat reQuest (ARQ)**
  - Full-duplex connection, error detection codes
  - The receiver sends a feedback to the transmitter, saying that if any error is detected in the received packet or not (Not-Acknowledgement (NACK) and Acknowledgement (ACK), respectively).
  - The transmitter retransmits the previously sent packet if it receives NACK.
- **Forward Error Correction (FEC)**
  - Simplex connection, error correction codes
  - The receiver tries to correct some errors
- **Hybrid ARQ (ARQ+FEC)**
  - Full-duplex, error detection and correction codes

Why using error correction coding?

- Error performance vs. bandwidth
- Power vs. bandwidth
- Data rate vs. bandwidth



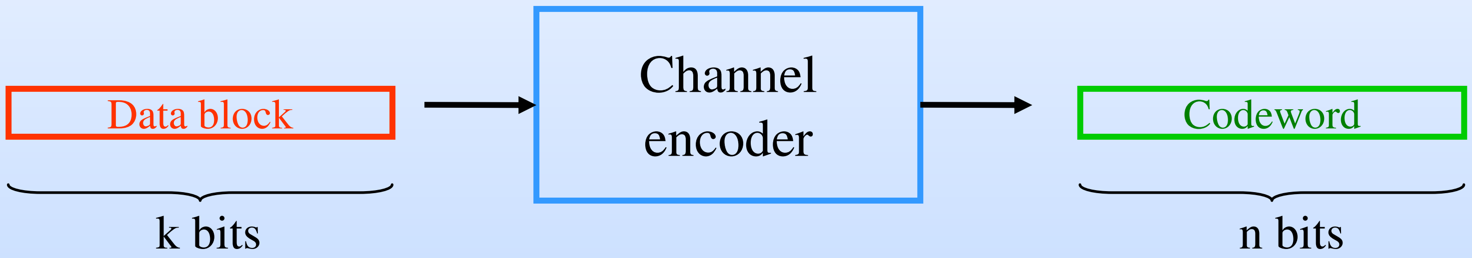
Coding gain:

For a given bit-error probability, the reduction in the Eb/N0 that can be realized through the use of code:

$$G\,[\text{dB}] = \left(\frac{E_b}{N_0}\right)_u [\text{dB}] - \left(\frac{E_b}{N_0}\right)_c [\text{dB}]$$

Linear block codes

- **The information bit stream is chopped into blocks of k bits.**
- **Each block is encoded to a larger block of n bits.**
- **The coded bits are modulated and sent over channel.**
- **The reverse procedure is done at the receiver.**



$n\text{-}k$ Redundant bits

$$R_c = \frac{k}{n}$$ Code rate

Linear block codes – cont’d

- **The Hamming weight of vector U, denoted by w(U), is the number of non-zero elements in U.**
- **The Hamming distance between two vectors U and V, is the number of elements in which they differ.**

- **The minimum distance of a block code is**

$$d(\mathbf{U},\mathbf{V}) = w(\mathbf{U} \oplus \mathbf{V})$$

$$d_{\min} = \min_{i \neq j} d(\mathbf{U}_i, \mathbf{U}_j) = \min_{i} w(\mathbf{U}_i)$$

Linear block codes – cont'd

- **Error detection capability is given by**

$$e = d_{\min} - 1$$

- **Error correcting-capability t of a code, which is defined as the maximum number of guaranteed correctable errors per codeword, is**

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$$

Linear block codes – cont'd

- **For memory less channels, the probability that the decoder commits an erroneous decoding is**

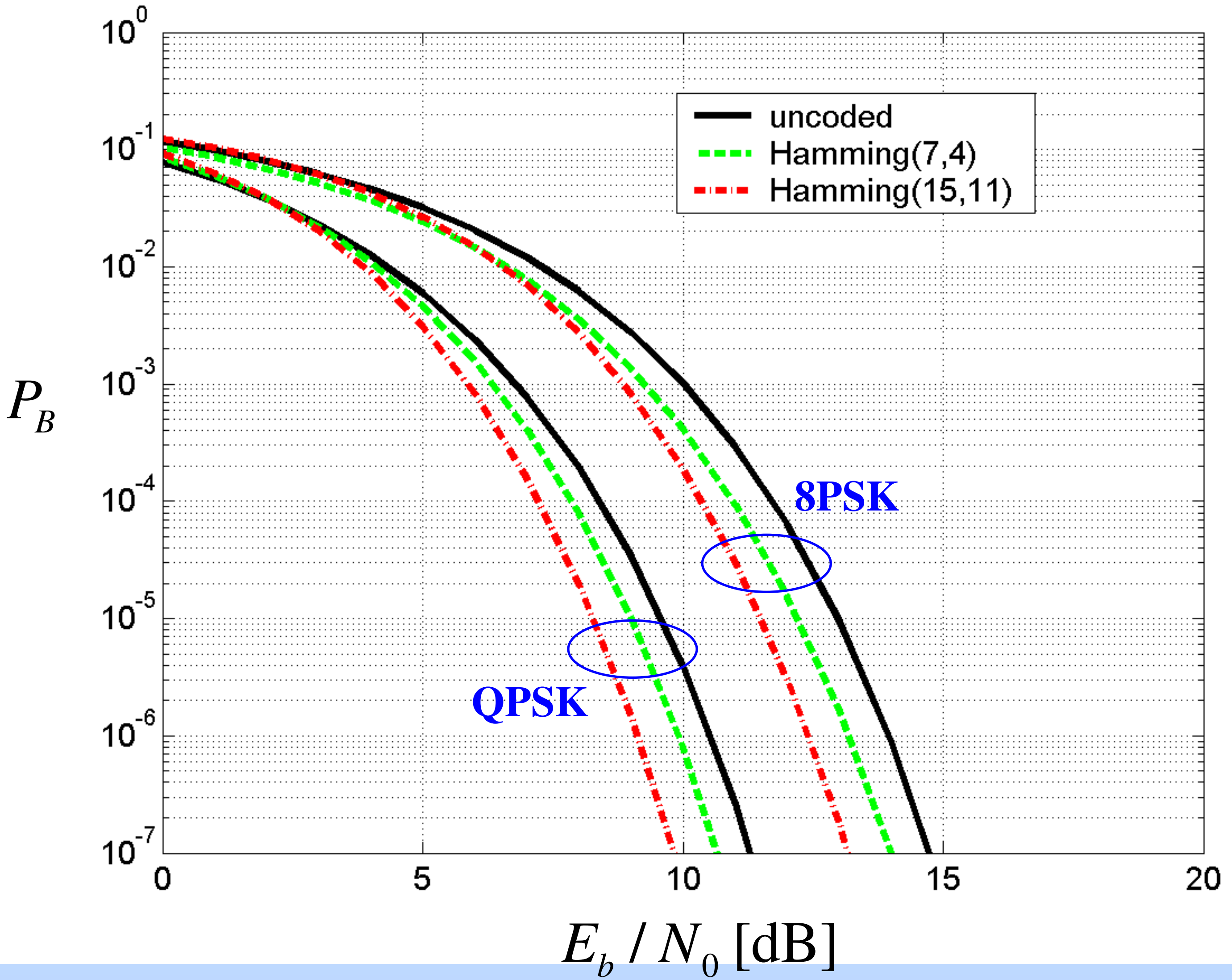
$$P_M \leq \sum_{j=t+1}^{n} \binom{n}{j} p^j (1-p)^{n-j}$$

  - p is the transition probability or bit error probability over channel.

- **The decoded bit error probability is**

$$P_B \approx \frac{1}{n} \sum_{j=t+1}^{n} j \binom{n}{j} p^j (1-p)^{n-j}$$

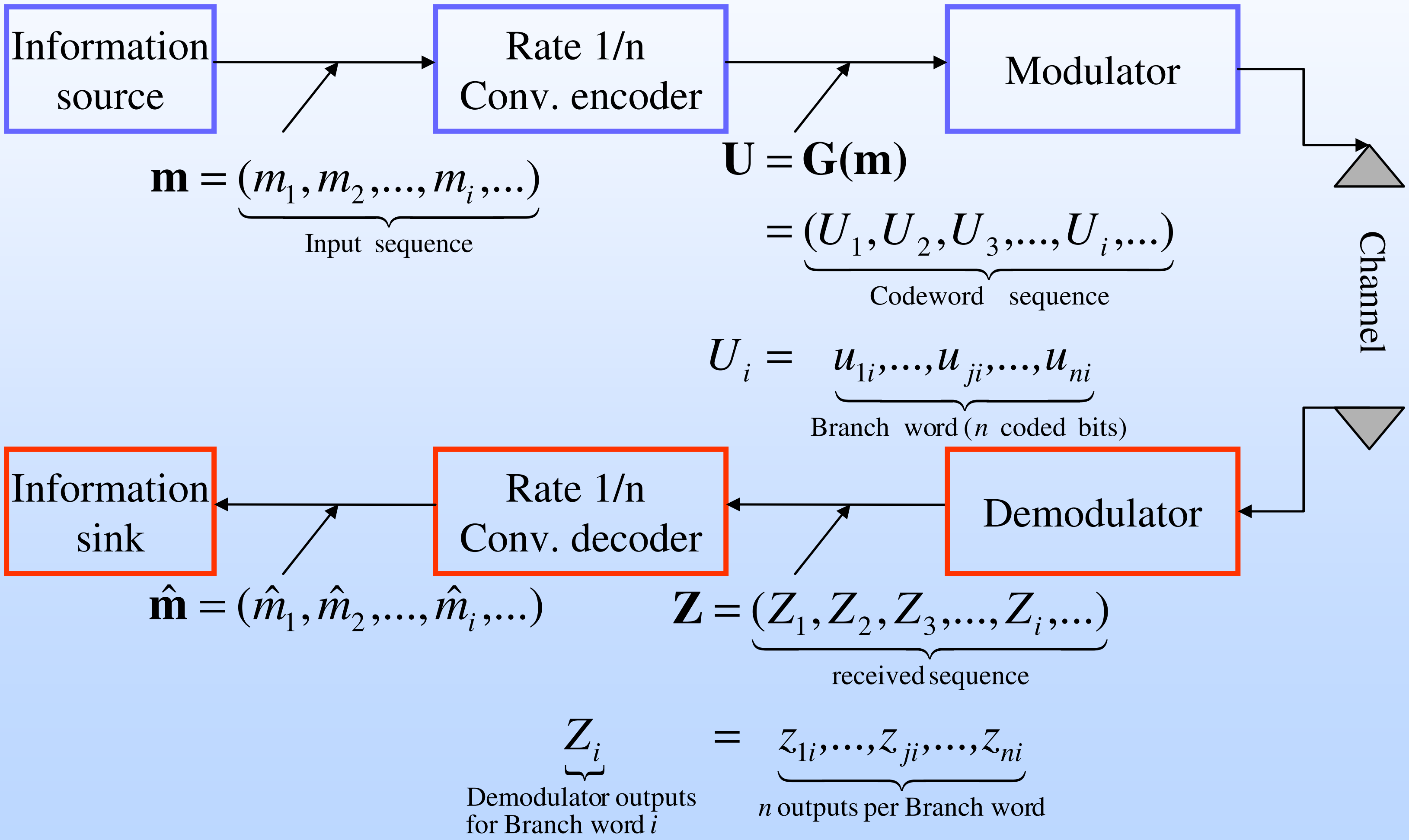
Example of the block codes



Convolutional codes

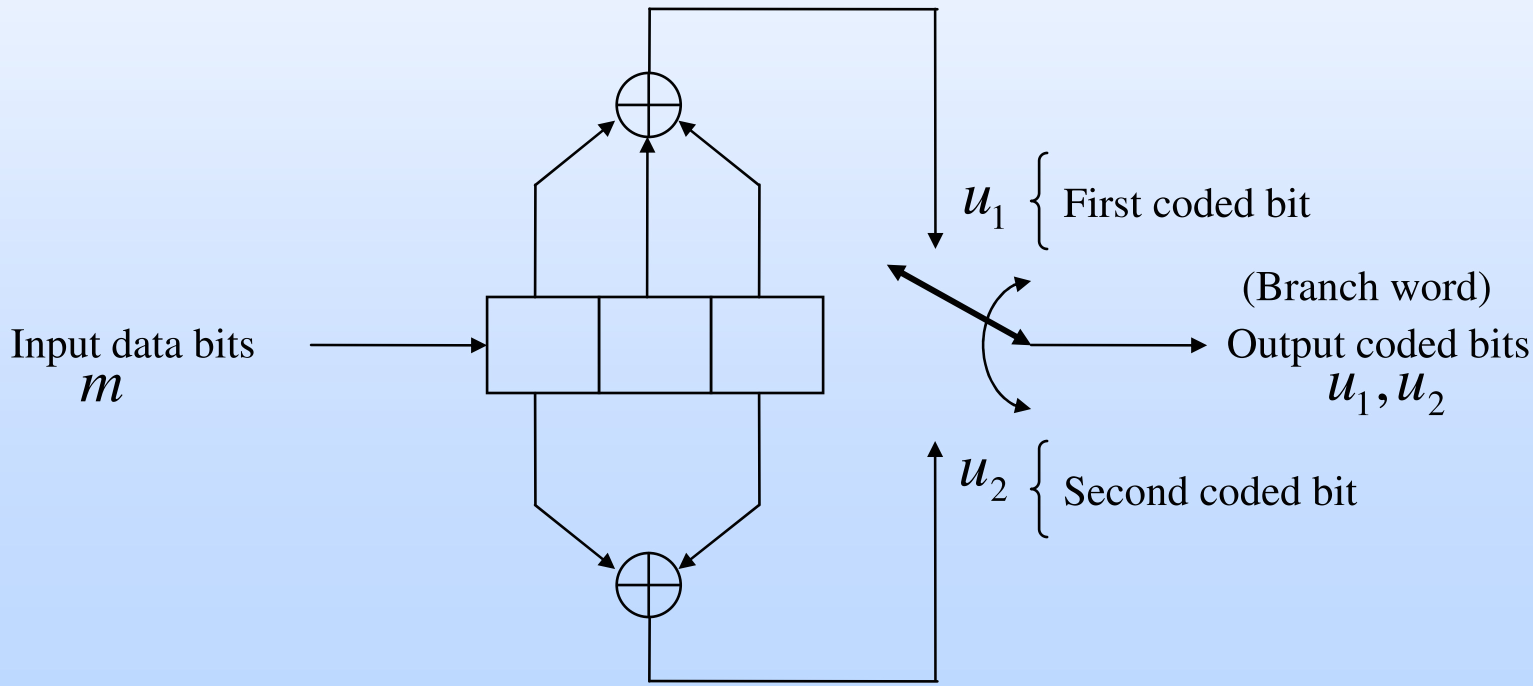
- **A Convolutional code is specified by three parameters** (n,k,K) **or** $(k/n,K)$ **where**
  - $R_c = k/n$ is the coding rate, determining the number of data bits per coded bit.
    - In practice, usually *k=1* is chosen and we assume that from now on.
  - *K* is the constraint length of the encoder and where the encoder has *K-1* memory elements.
    - There is different definitions in literatures for constraint length.

Block diagram of the DCS



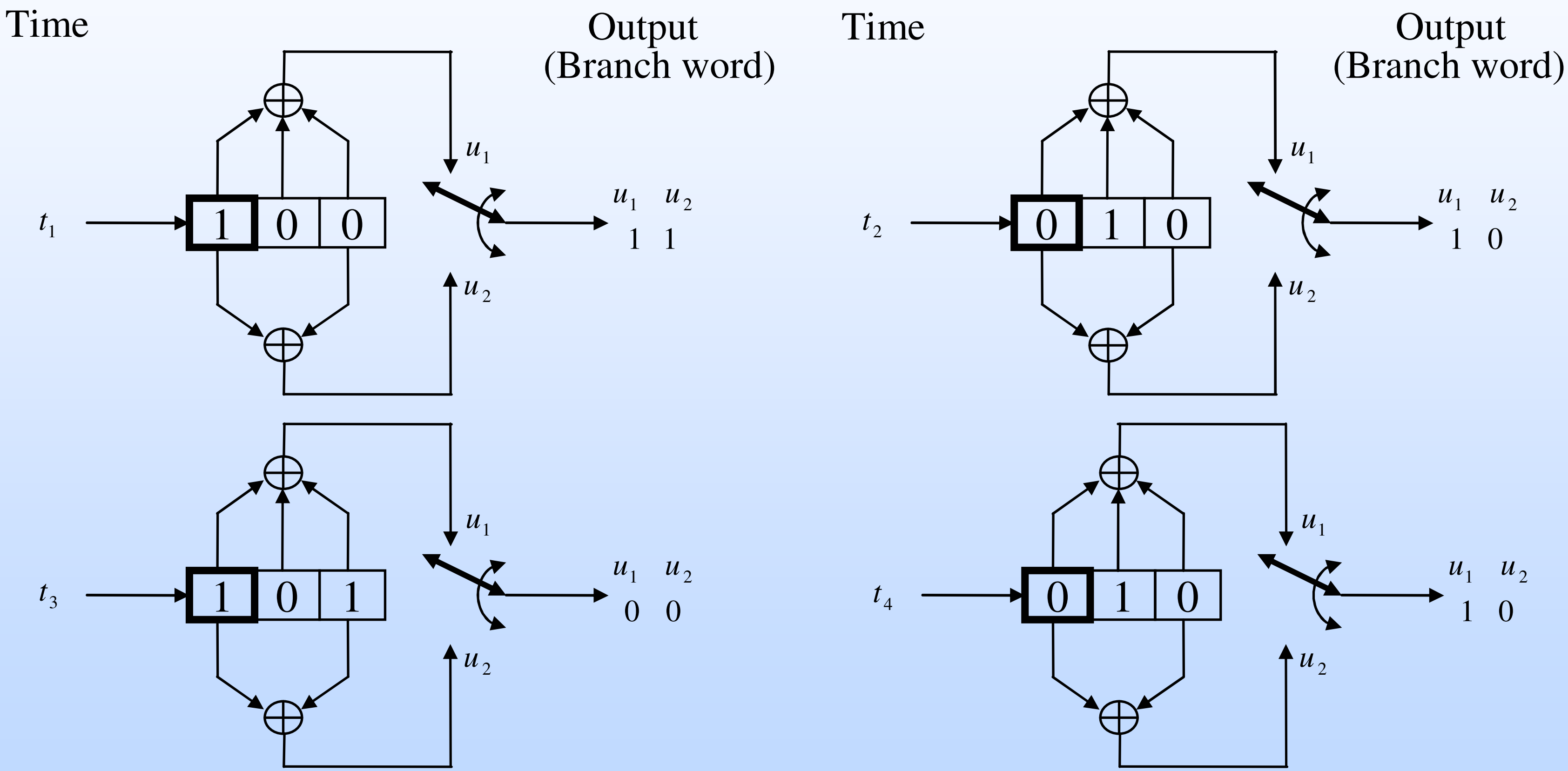
A Rate ½ Convolutional encoder

- **Convolutional encoder (rate ½, K=3)**
  - 3 shift-registers where the first one takes the incoming data bit and the rest, form the memory of the encoder.

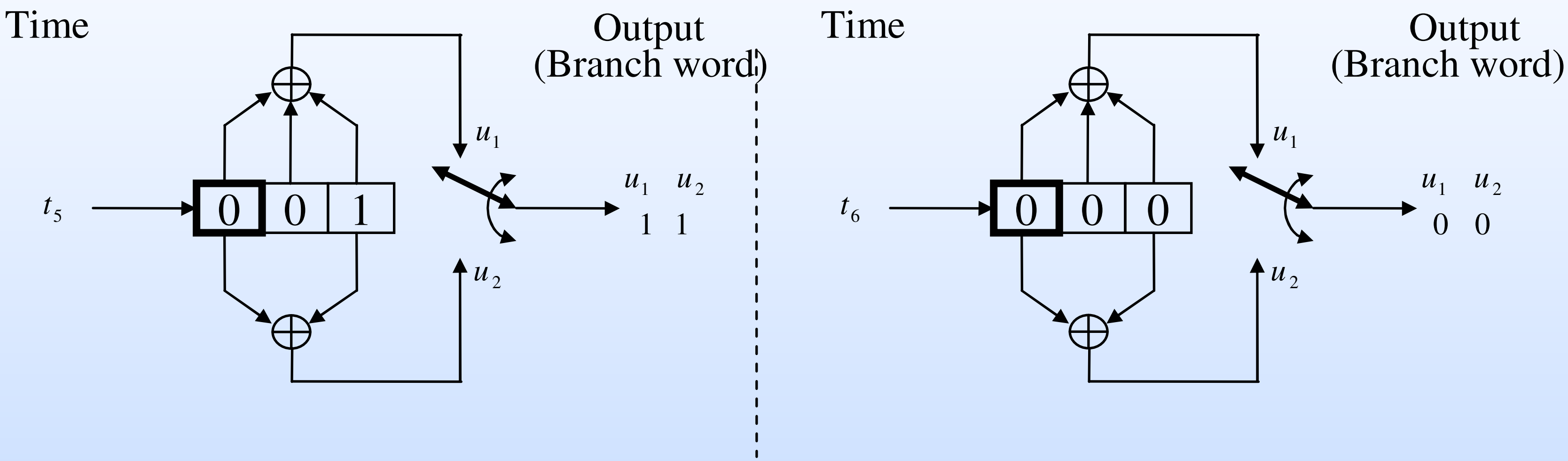


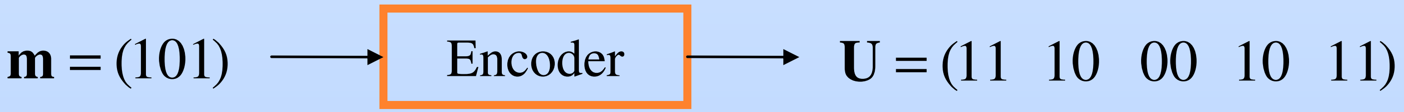
A Rate ½ Convolutional encoder

Message sequence: $\mathbf{m} = (101)$



A Rate ½ Convolutional encoder

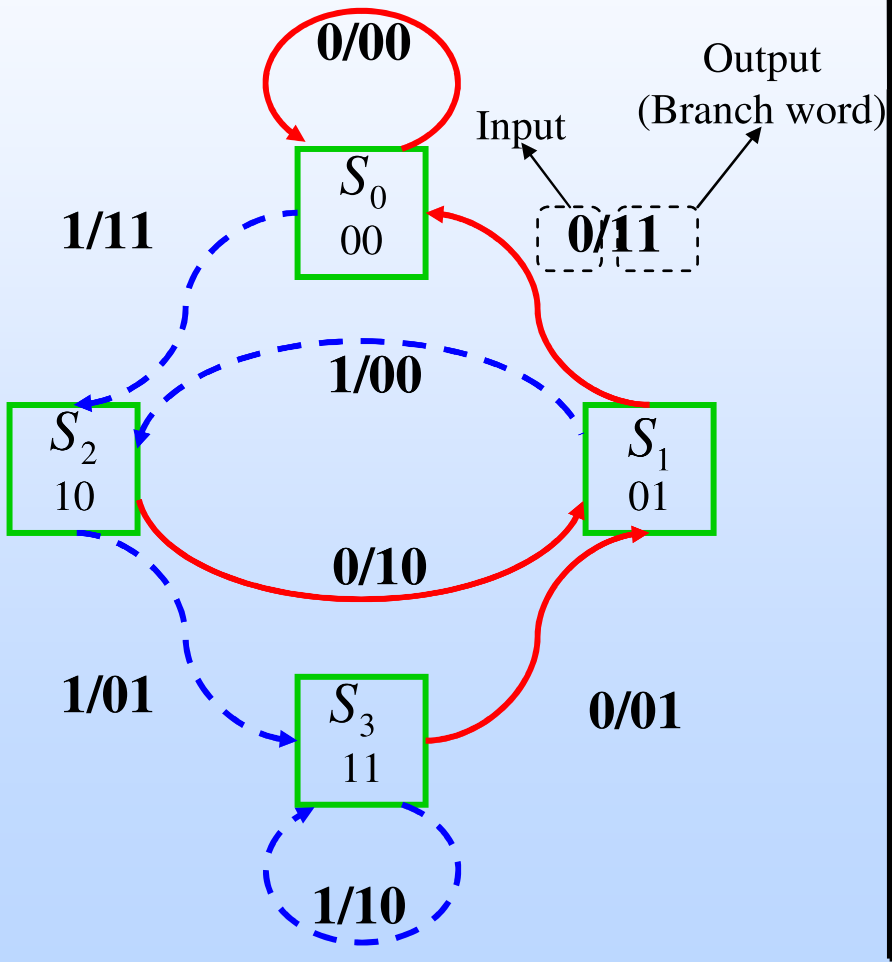




State diagram

- A finite-state machine only encounters a finite number of states.
- State of a machine: the smallest amount of information that, together with a current input to the machine, can predict the output of the machine.
- In a Convolutional encoder, the state is represented by the content of the memory.
- Hence, there are 2^{K-1} states.
- A state diagram is a way to represent the encoder.
- A state diagram contains all the states and all possible transitions between them.
- Only two transitions initiating from a state
- Only two transitions ending up in a state.

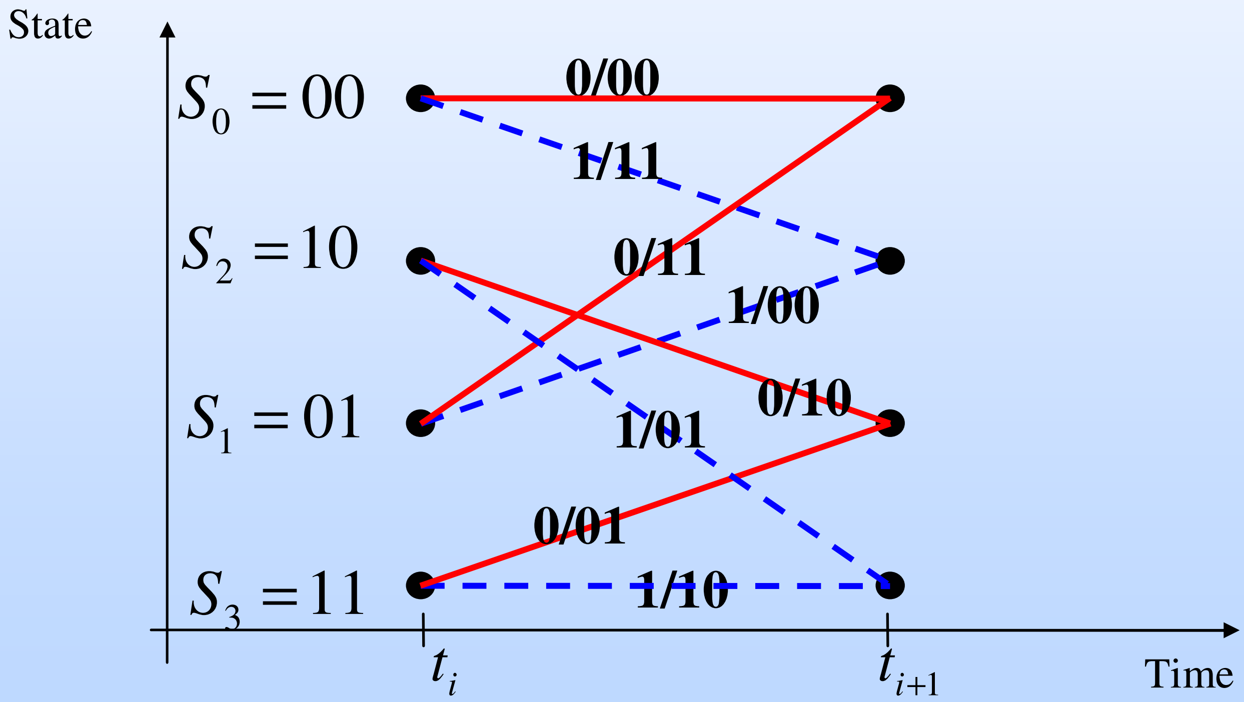
State diagram – cont'd



Current state	input	Next state	output
S_0 00	0	S_0	00
	1	S_2	11
S_1 01	0	S_0	11
	1	S_2	00
S_2 10	0	S_1	10
	1	S_3	01
S_3 11	0	S_1	01
	1	S_3	10

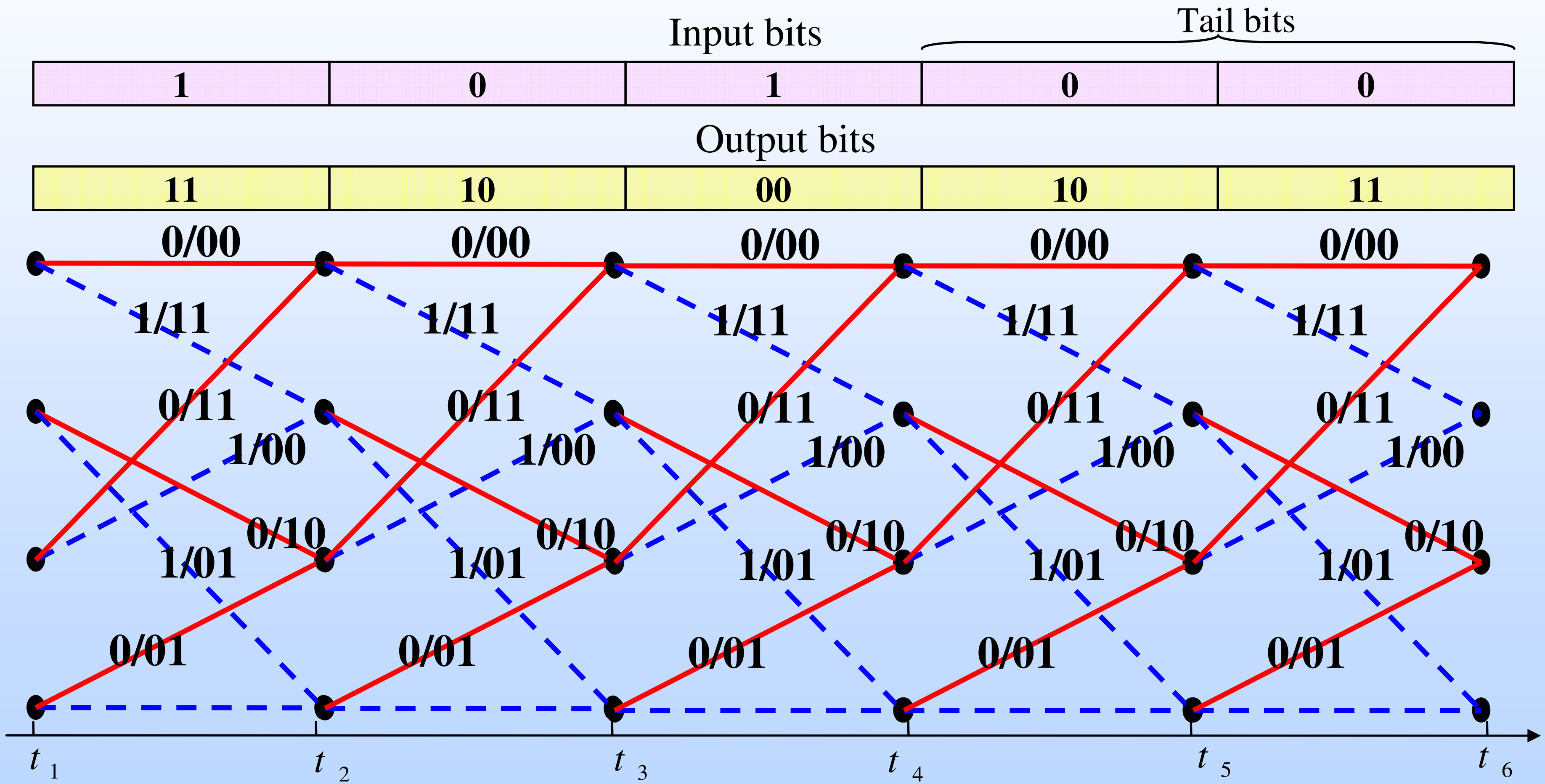
Trellis

- **Trellis diagram is an extension of the state diagram that shows the passage of time.**
  - Example of a section of trellis for the rate ½ code

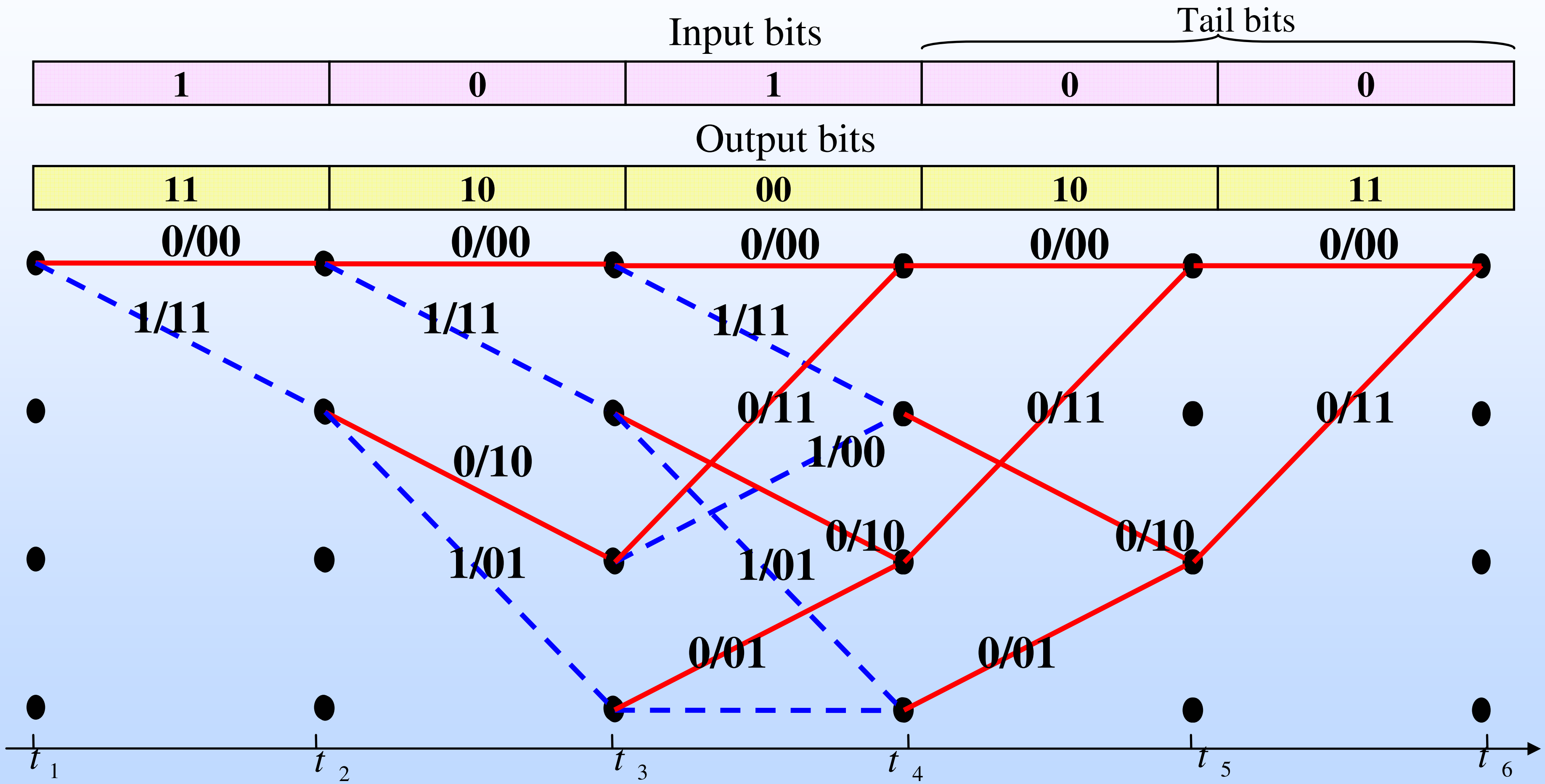


Trellis –cont'd

- **A trellis diagram for the example code**



Trellis – cont'd



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 smallest 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 $\Gamma\left(S(t_i), t_i\right)$ 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.

1. If $i < L + K$, increase i by 1 and return to step 2.

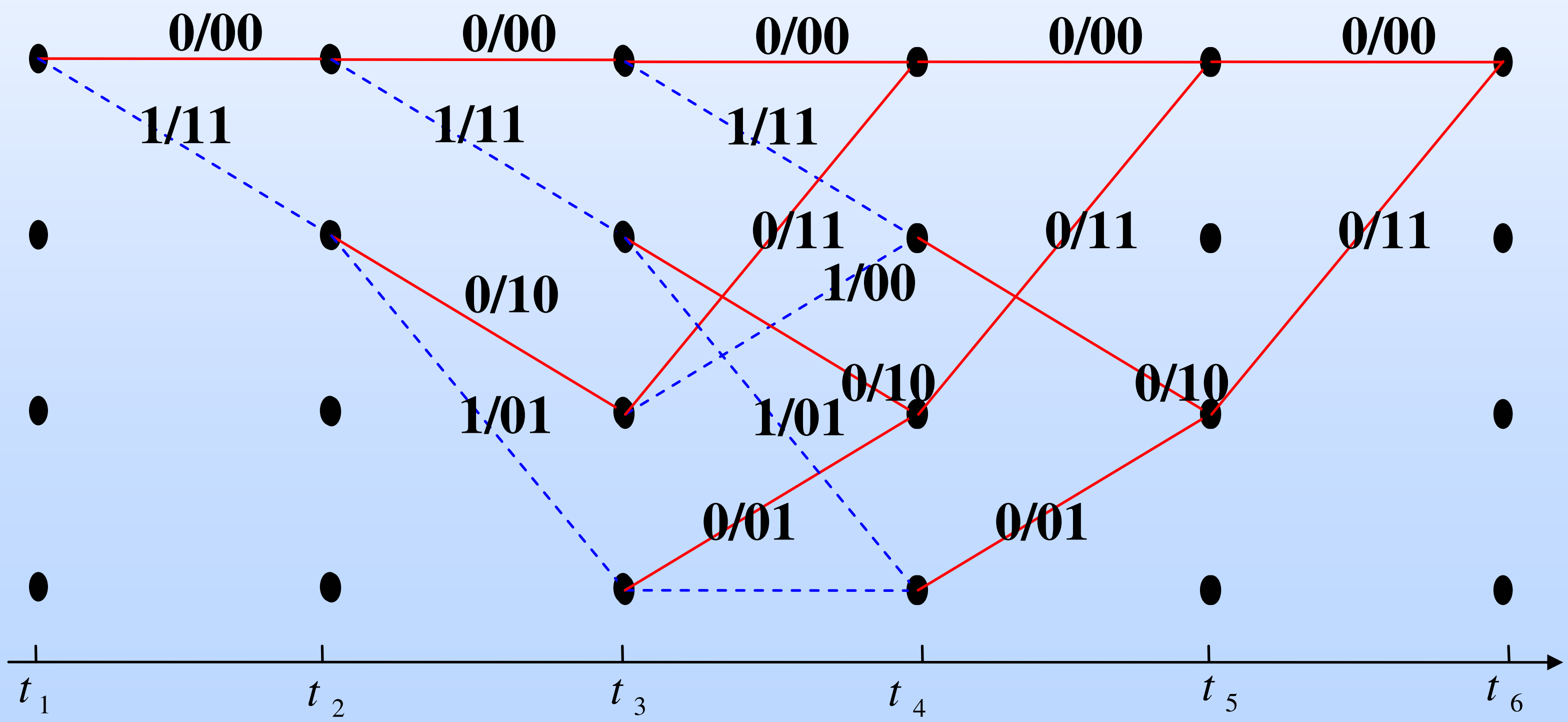
A. Start at state zero at time t_{L+K}. Follow the surviving branches backwards through the trellis. The path thus defined is unique and correspond to the ML codeword.

Example of Hard decision Viterbi decoding

$\mathbf{m} = (101)$

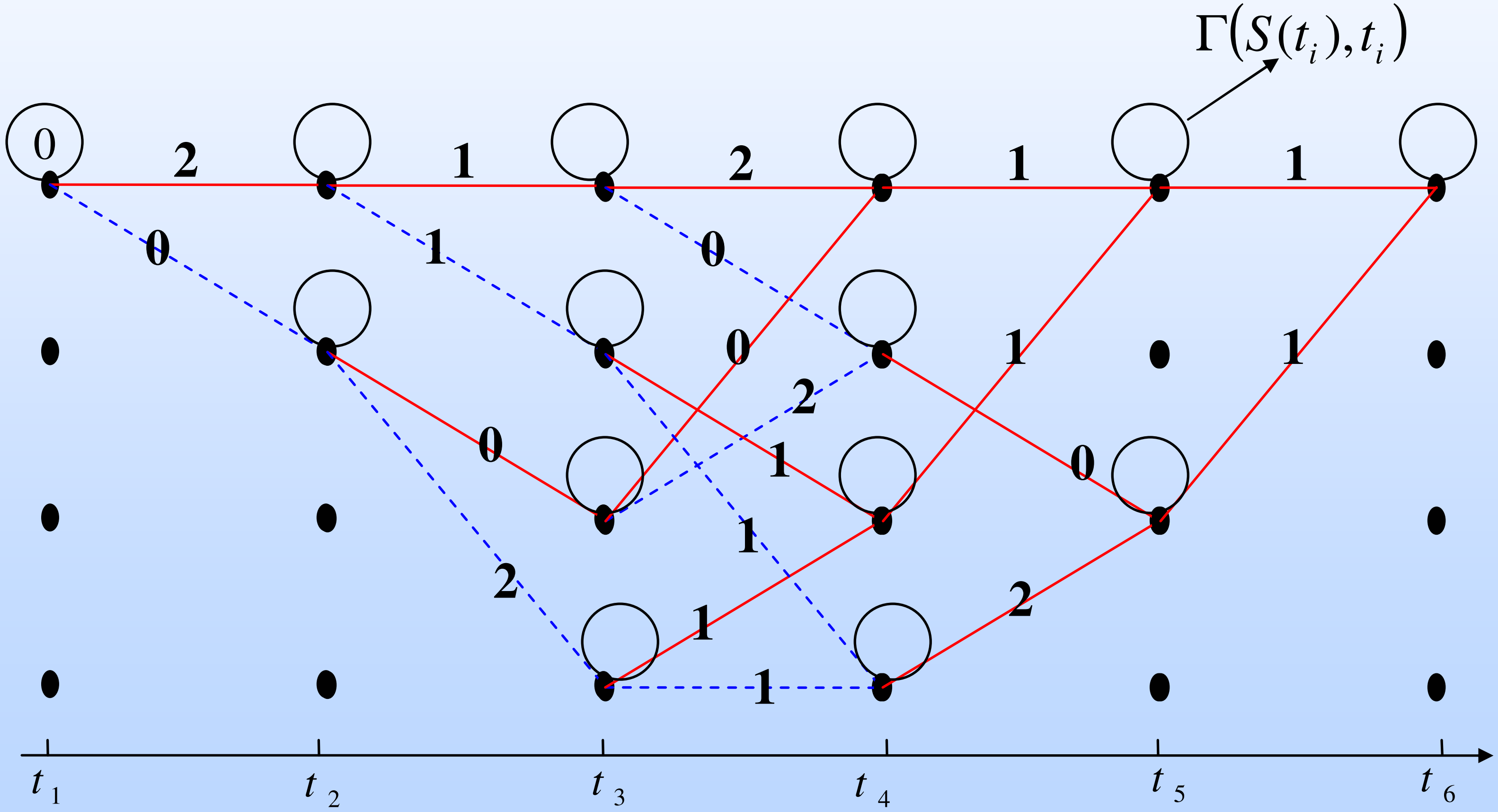
$\mathbf{U} = (11\ \ 10\ \ 00\ \ 10\ \ 11)$

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



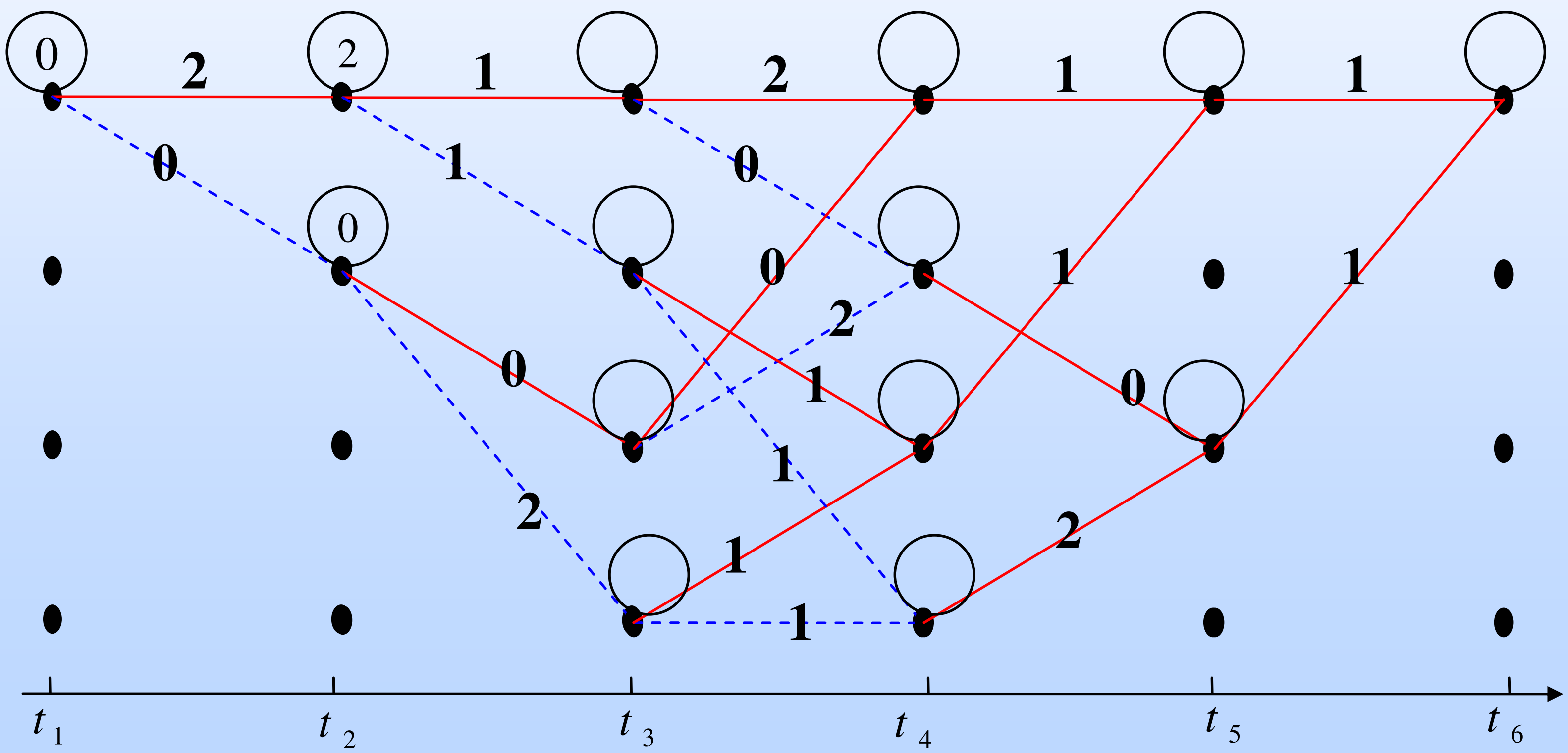
Example of Hard decision Viterbi decoding-cont'd

- **Label al the branches with the branch metric (Hamming distance)**



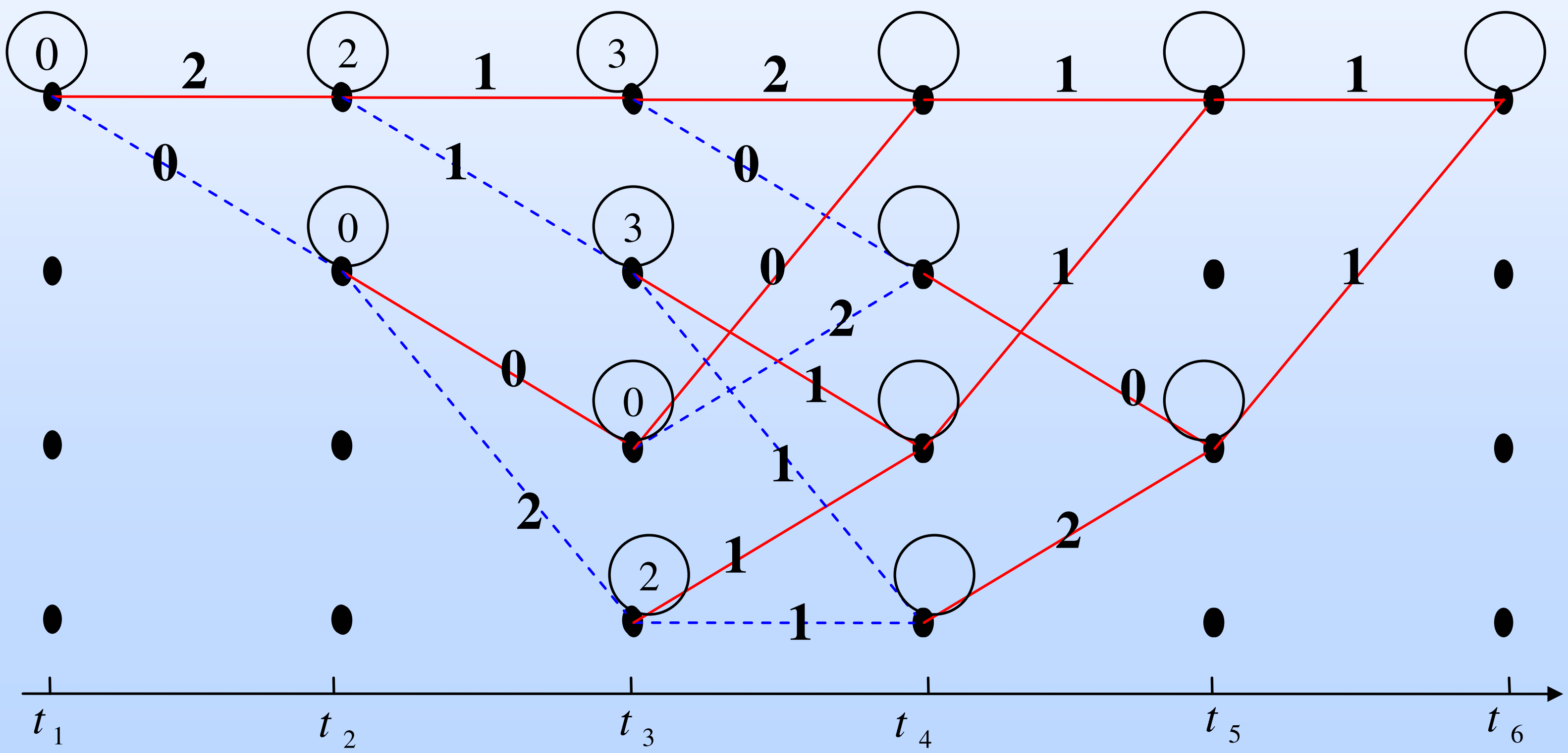
Example of Hard decision Viterbi decoding-cont'd

- **i=2**



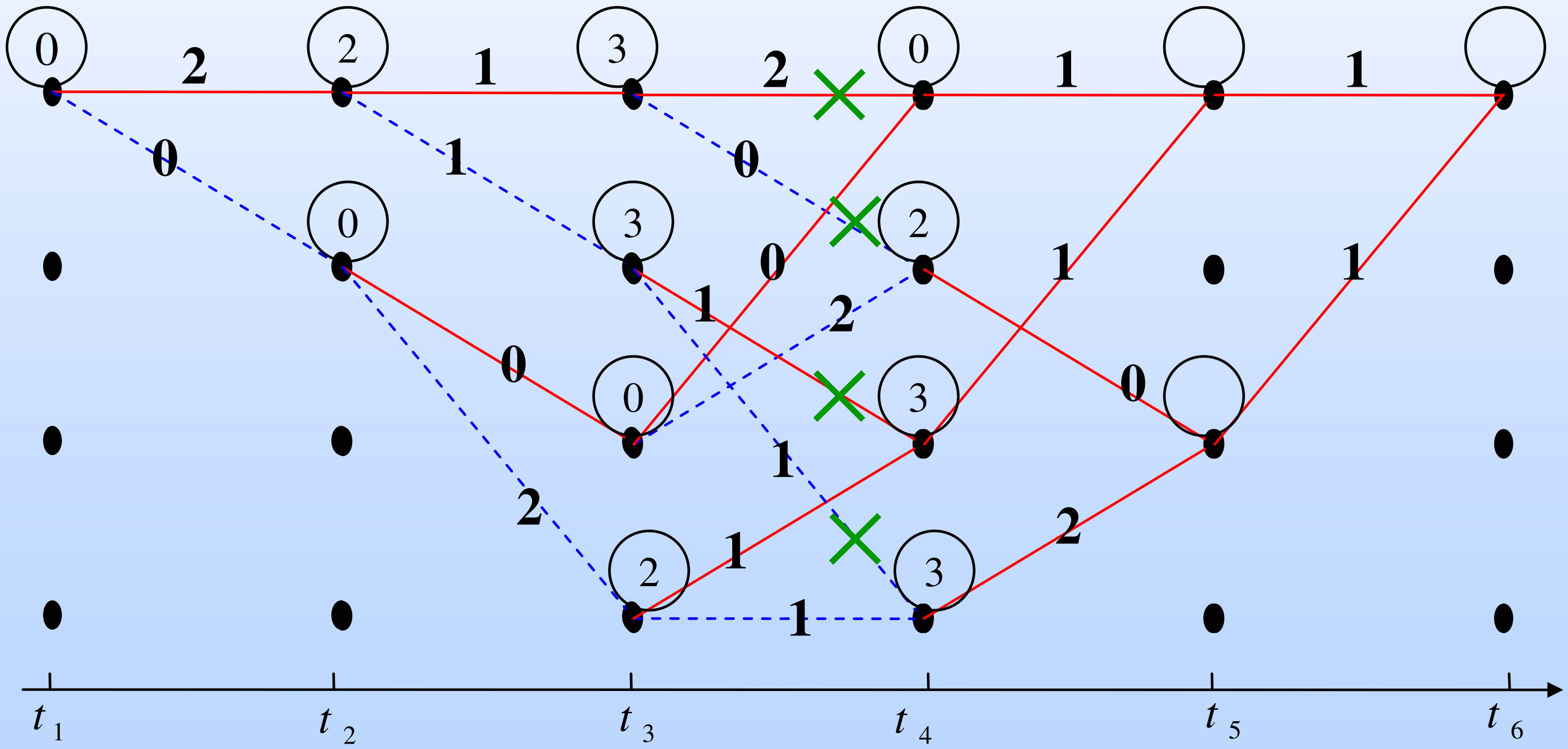
Example of Hard decision Viterbi decoding-cont'd

- **i=3**



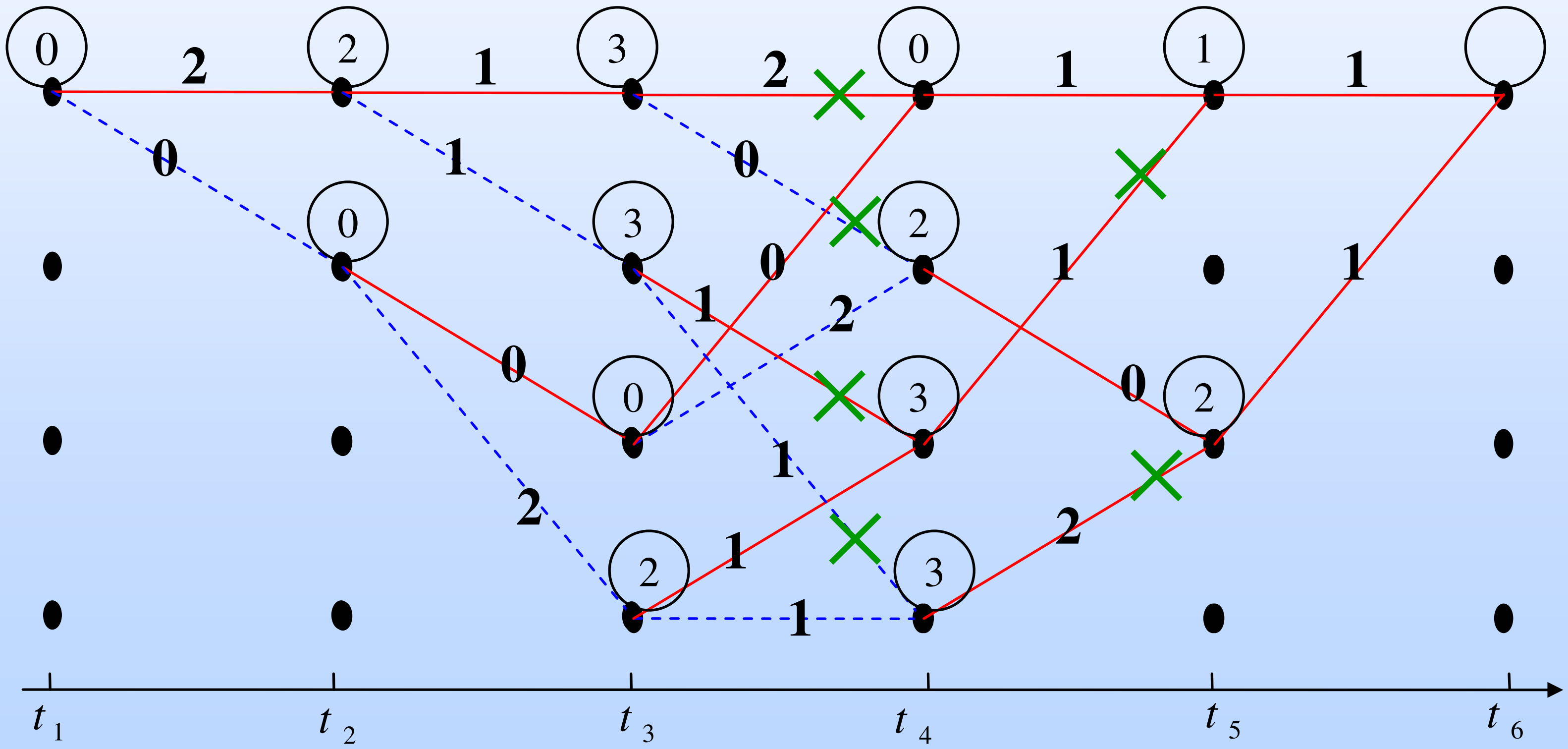
Example of Hard decision Viterbi decoding-cont'd

- **i=4**



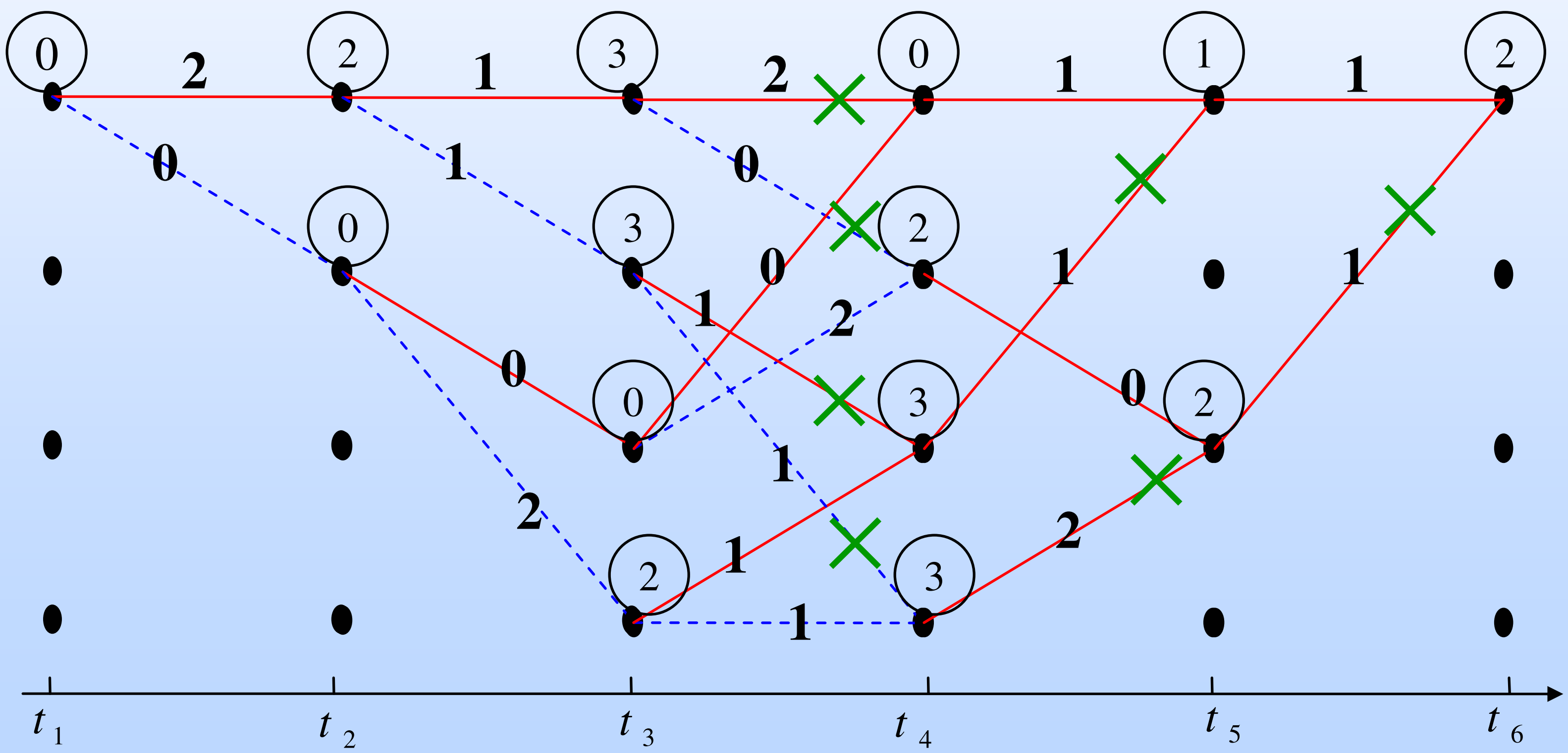
Example of Hard decision Viterbi decoding-cont’d

- **i=5**



Example of Hard decision Viterbi decoding-cont'd

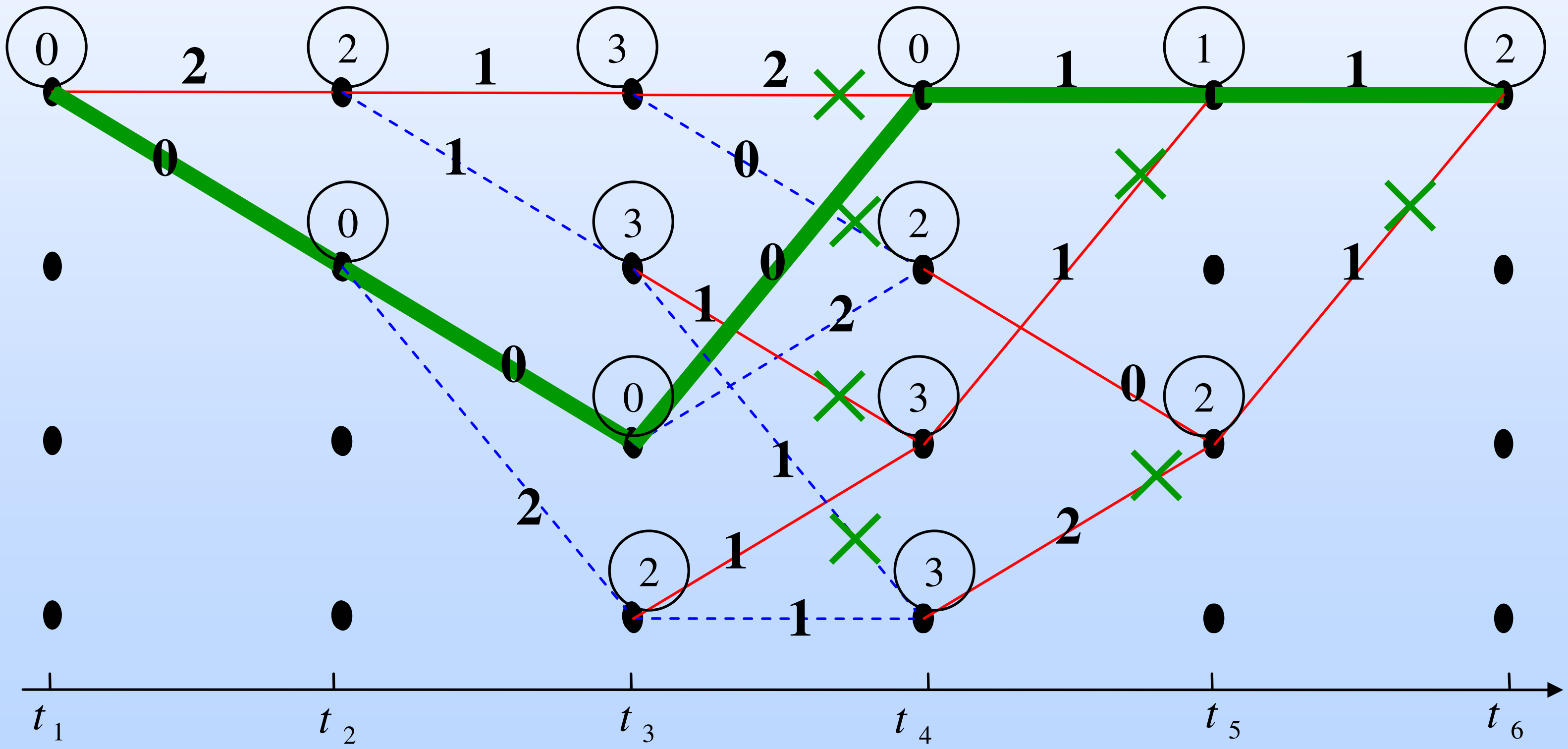
- **i=6**



Example of Hard decision Viterbi decoding-cont'd

- **Trace back and then:**

$$\hat{\mathbf{m}} = (100)$$
$$\hat{\mathbf{U}} = (11 \quad 10 \quad 11 \quad 00 \quad 00)$$



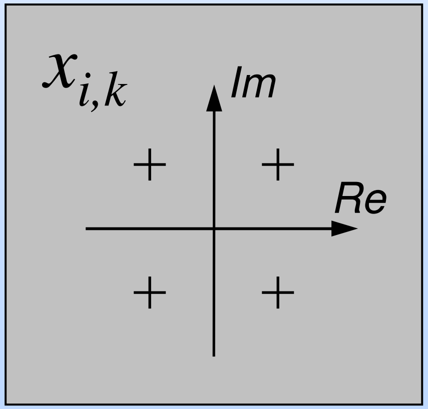
Generating the OFDM signal (1)

- **Symbol (QPSK) of sub-carrier *i* at time *k***
  - Other symbol-alphabets can be used as well (BPSK, m-QAM)
- **Baseband signal is generated by DSP**

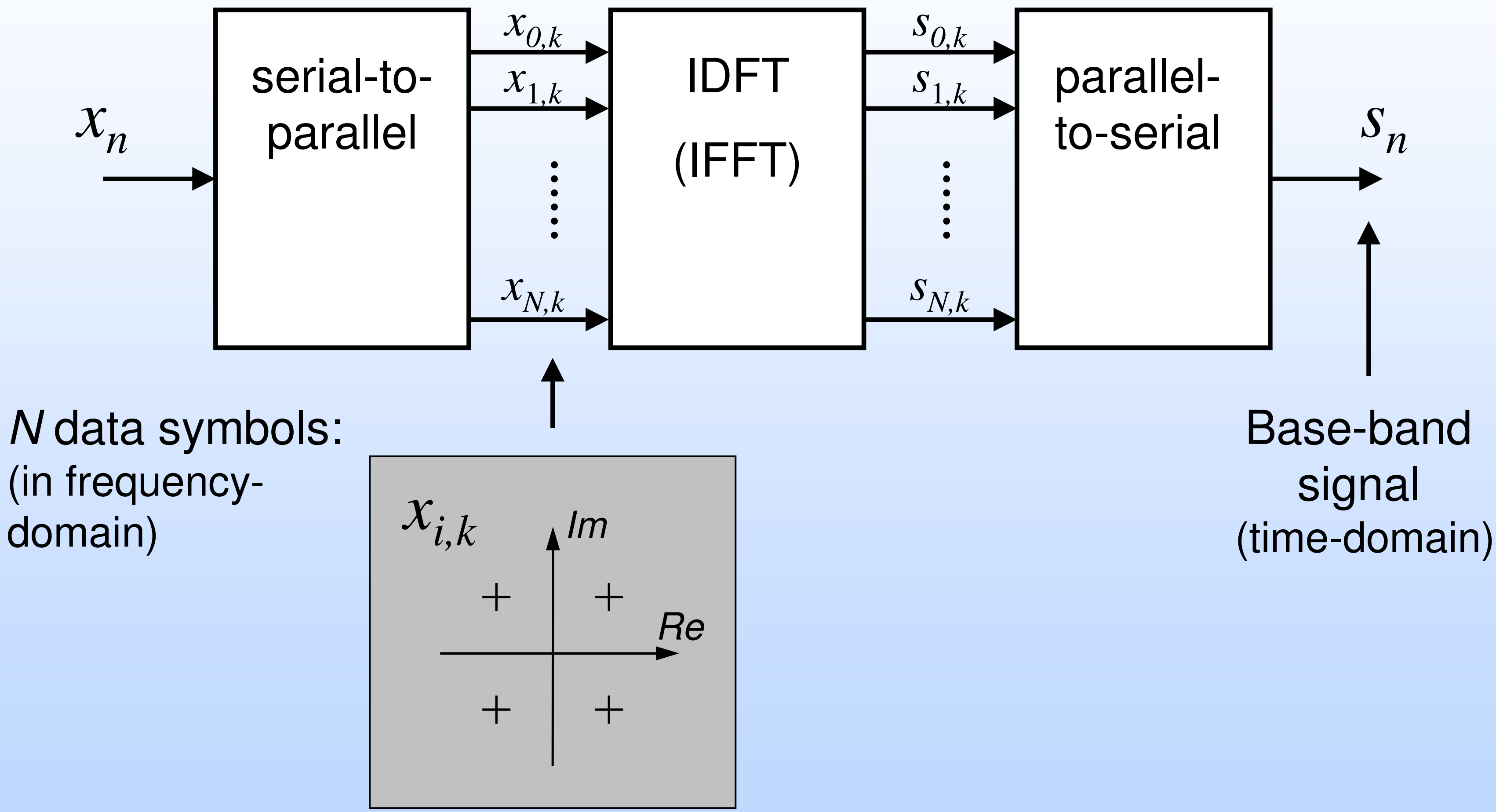
$$s_{BB,i,k}(t) = w(t-kT)\cdot x_{i,k}\cdot \exp\left[j2\pi i\Delta f\,(t-kT)\right]$$

Window function

Sub-carrier

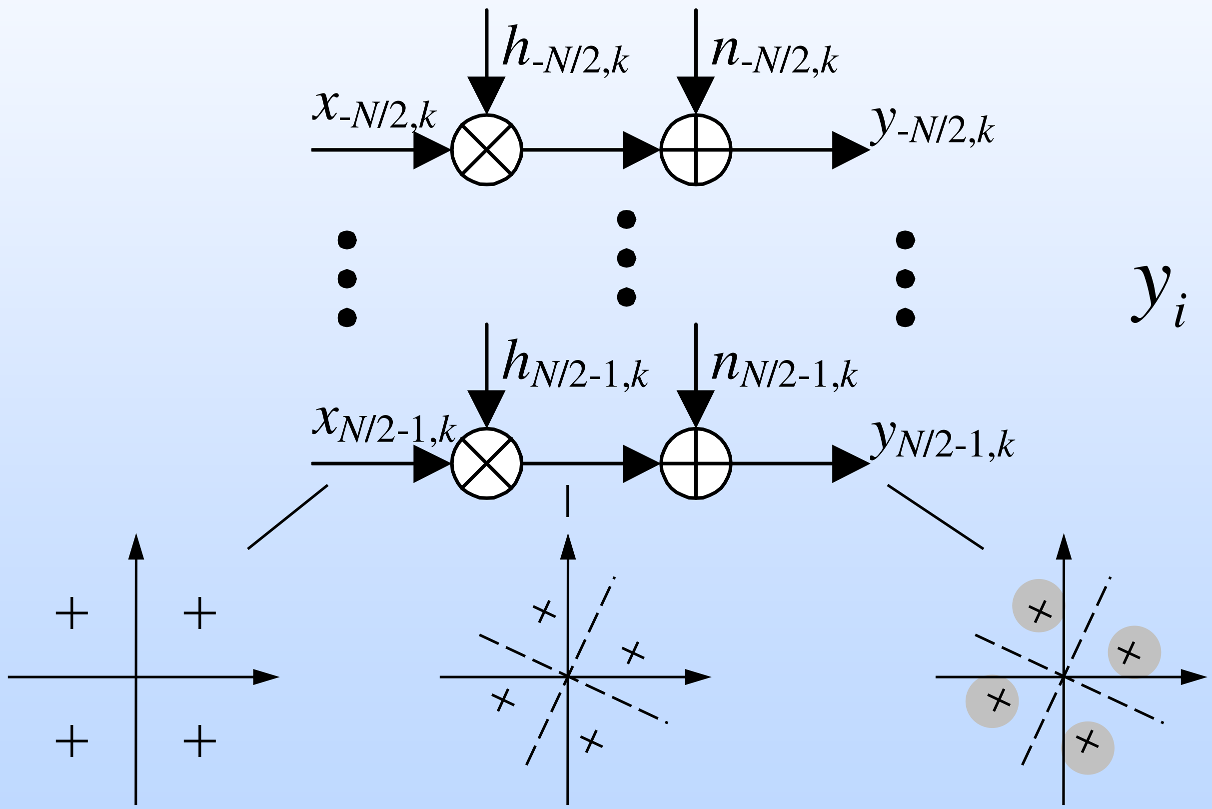


Generating the OFDM signal (2)



OFDM System Model

- **Multiplication of data symbols with (complex-valued) channel transfer-function:**

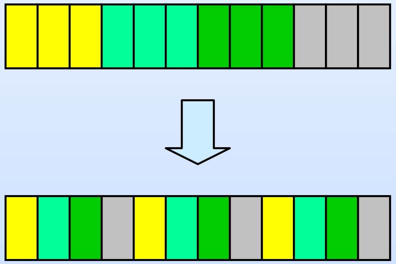


$$y_i = x_i h_i + n_i$$

Bit interleaving

Transmitter

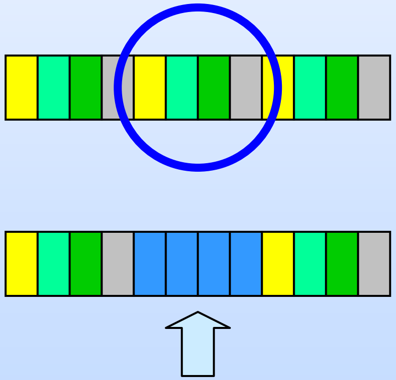
Bits are interleaved ...



... and will be de-interleaved in the receiver

Channel

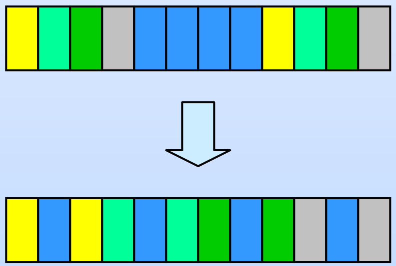
Fading affects many adjacent bits



Bit errors in the receiver

Receiver

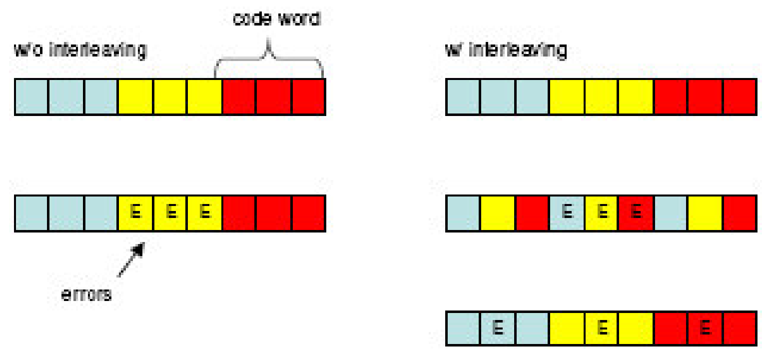
After de-interleaving of bits, bit errors are spread!



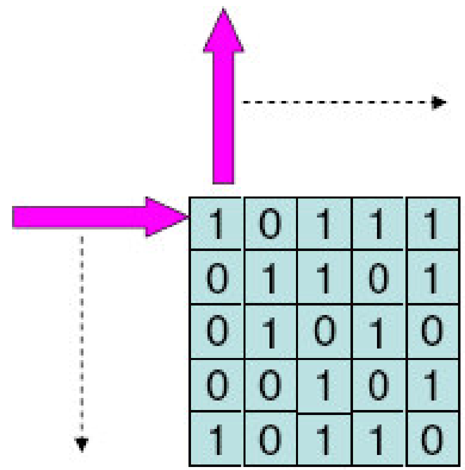
(better for FEC)

Coding / Interleaving

- Interleaving
  - Scatters error bursts
  - Can be done in time or in frequency domain

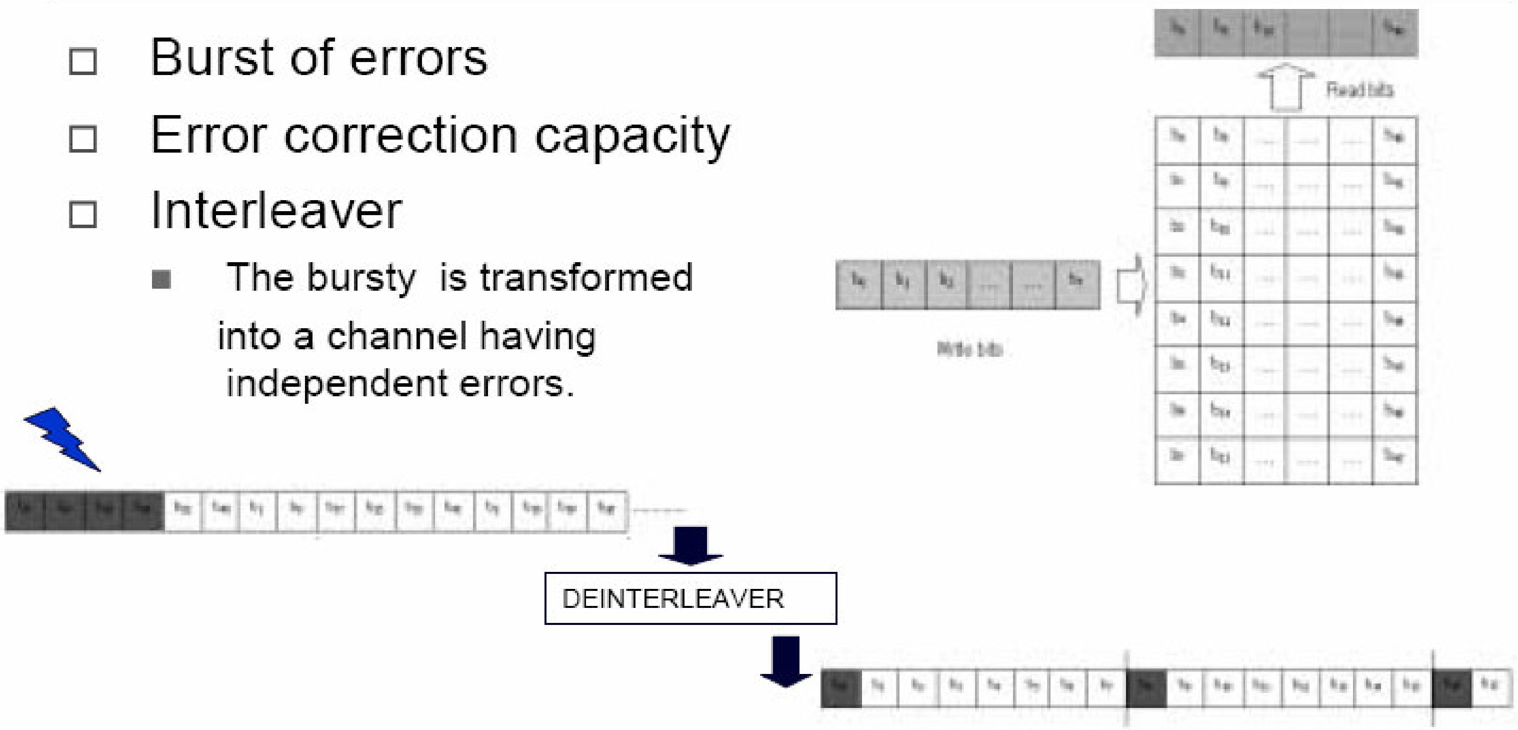


- One of the simplest form is block interleaving
  - Write row-by-row
  - Read column-by-column (or another way around)
  - Additional matrix permutation is possible



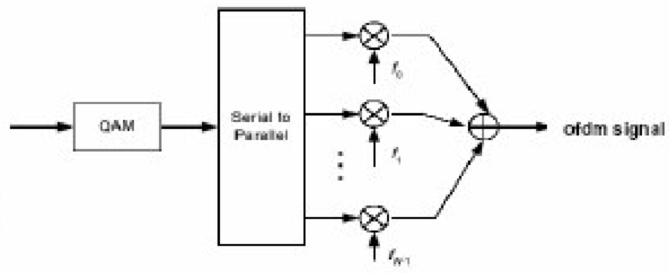
Interleaver

- Burst of errors
- Error correction capacity
- Interleaver
  - The bursty is transformed into a channel having independent errors.



Homework

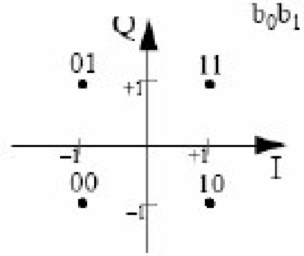
- Derive expression for OFDM-signal
- Use 4 subchannels and 4QAM



OFDM transmitter

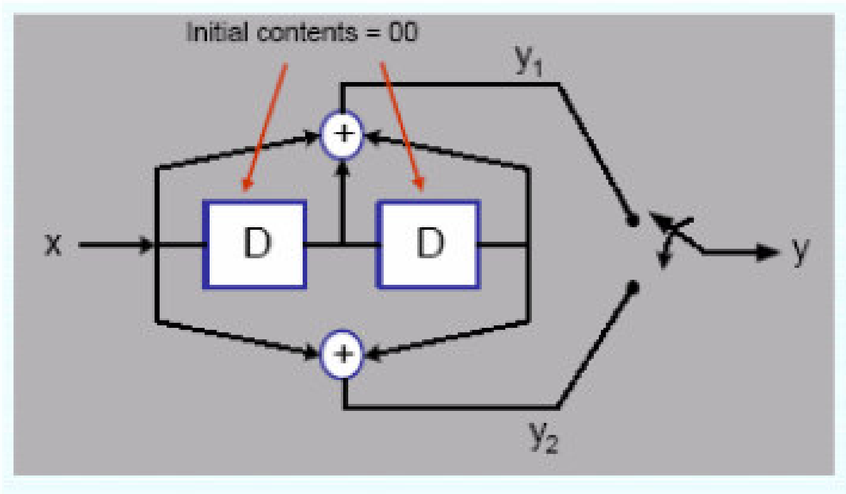
- Input data sequence:
  11 01 00 10
- Subcarrier frequencies are:
  $-2f_c\ -1f_c\ 1f_c\ 2f_c$

4QAM



- Please use Viterbi algorithm to decode the received sequence:

  Z=[11 10 10 10 01]



Please draw the trellis and state diagram