

## ЛЕКЦИЯ 5 ПРЕДАВАНЕ И КОДИРАНЕ НА СЪОБЩЕНИЯ

- ⌚ **Модел на система за предаване на съобщения**
- ⌚ **Определения (азбука, кодиране)**
- ⌚ **Два подхода при кодирането**
- ⌚ **Универсална азбука за кодиране**
- ⌚ **Кога е необходимо кодиране**
- ⌚ **Примери за кодиране**

ИНФО\_05

1/33

## ПРЕДАВАНЕ НА СЪОБЩЕНИЯ

Сама по себе си информацията, представена (**кодирана**) под формата на съобщение, **няма особена ценност, ако не** може да бъде **доставена на субект**, който **да я извлече** от носителя и **използва**.

От тази гледна точка **предаването** (**пrenaсянето** от едно място на друго) **на съобщения върху физически носител играе важна роля**.

**Предаване на съобщения може да се осъществява и между автомати.**

ИНФО\_05

2/33

## МОДЕЛ НА ШЕНОН

**Широко разпространение** е получил създаденият от френския учен **Клод Шенон** през 1948 г. **модел**, който е представен по-долу:



ИНФО\_05

3/33

## ИЗТОЧНИК НА СЪОБЩЕНИЯ

**Източникът на сигнали (информация)** **подготвя съобщенията**, които трябва да бъдат предадени **като редица от знакове** (принадлежащи на **азбуката на източника**).

Ще отбележим, че **днес повечето системи за предаване на данни имат дискретен (цифров) характер**.

ИНФО\_05

4/33

## КАНАЛ ЗА ПРЕНАСЯНЕ

Ролята на **канал за пренасяне** се играе от **физическа среда** (медия), в която се предават **сигналите** (знаковете).

**Каналът за пренасяне** (връзка) може да бъде **електрически или оптически проводник, радио или радиорелеен канал**.

**Сред характеристиките на тази среда най-важна е скоростта на пренасяне.**

ИНФО\_05

5/33

## ИЗТОЧНИК НА ШУМ

В резултат от смущения в канала е **възможно оригиналното съобщение да бъде „изкривено“** и някои пренасяни **сигнали да бъдат променени** (**подменени с други**).

**„Изкривяването“ на сигналите се моделира като се въвежда обект източник на шум в канала.**

ИНФО\_05

6/33

## ПРИМЕР: ГРЕШКИ ПРИ ПРЕДАВАНЕ НА СЪОБЩЕНИЯ

- 🔔 **Оригинал:** 00001101001100111101.
- 🔔 **Получено:** 00011101001110101101.
- 🔔 **Брой грешки:** 3.
- 🔔 **В зависимост от броя на грешките се говори за съобщения с една, две и повече грешки.**

ИНФО\_05

7/33

## БЛОКОВЕ ЗА КОДИРАНЕ И ДЕКОДИРАНЕ

Според К. Шенон, **основен проблем**, който възниква в процеса **на предаване на съобщения**, е **точно** или **приближено възстановяване на оригинала**.

За да се сведе до **приемливо равнище шумовият ефект между източника и канала** се разполага **блок за кодиране**, а **между канала и приемника – блок за декодиране**.

ИНФО\_05

8/33

## ПРИЕМНИК НА СИГНАЛИ

Приемникът на сигнали получава редица от знакове (на съответната азбука), по която възстановява съдържанието и структурата на съобщението.

Не е задължително азбуката на приемника да съвпада с азбуката на източника, но не е и задължително те да се различават.

ИНФО\_05

9/33

## ОПРЕДЕЛЕНИЯ: АЗБУКА

За дискретно представяне на информация се говори, когато съответното съобщение се представя от отделни знакове, избирани сред предварително определено множество (наречено азбука).

Знаковете на азбуката могат да имат различни представяния и носители (рисунок, трепение, звук и др.).

От формална гледна точка за да се представи чрез тях съобщение е необходимо:

- ① Да се различават един от друг.
- ② Броят им да е краен.

ИНФО\_05

10/33

## ОПРЕДЕЛЕНИЯ: КОД

Думата код произлиза от латинската **codex** (закон, сборник от правила).

Код се нарича всеки алгоритъм за превеждане на едно съобщение (наричано оригинал) в друго (вторично) съобщение.

От интерес за практиката са кодове, за които съществува алгоритъм (код), който възстановява еднозначно оригиналното съобщение по вторичното, т. е. без добавяне или загуба на данни.

Под код понякога се разбира и полученото вторично съобщение.

ИНФО\_05

11/33

## ОПРЕДЕЛЕНИЯ: КОДИРАНЕ

Процесът на трансформиране на оригиналното съобщение във вторично се нарича кодиране.

Обратният процес на възстановяване на оригинала по вторичното съобщение вместо кодиране се нарича декодиране.

Често думата кодиране се използва и като синоним на представяне, главно когато се договаряме как ще изглеждат знаковете на азбуката, която използваме.

ИНФО\_05

12/33

## ПРЕДСТАВЯНЕ НА 0 И 1 В ТЕХНИЧЕСКИ СИСТЕМИ

	0	1
ток със сила	0 mA	20 mA
ток с напрежение	$0 \div 0,8 \text{ V}$	$2 \div 5 \text{ V}$
светлина	няма	има
кондензатор	разреден	зареден

ИНФО\_05

13/33

## ДВАТА ПОДХОДА ПРИ КОДИРАНЕ

Основните подходи при (алгоритмите за) кодиране на съобщения са два:

- ① Кодиране на азбуката.
- ② Кодиране, зависещо от съдържанието на съответното (кодираното) съобщение.

ИНФО\_05

14/33

## ПРИМЕР: КОДИРАНЕ, ЗАВИСИМО ОТ СЪДЪРЖАНИЕ

Оригинал: Код се нарича всеки алгоритъм за превеждане на ...

К	О	Д	И	Р	А	Н	Е
5	7	2	4	8	1	6	3
К	О	Д	С	Е	Н	А	Р
И	Ч	А	В	С	Е	К	И
А	Л	Г	О	Р	И	Т	Ъ
М	З	А	П	Р	Е	В	Е
Ж	Д	А	Н	Е	Н	А	*

Код: НЕИЕН ДАГАА РИЪЕ\* СВОПН  
КИАМЖ АКТВА ДЧЛЗД ЕСРРЕ

ИНФО\_05

15/33

## КОДИРАНЕ НА АЗБУКИ

При този подход всяка буква от азбуката на оригинала се замества с една дума от вторичната азбука, наречена кодова дума.

Съответствието буква–дума се нарича кодова таблица.

- ① Вторичната азбука може да съвпада с оригиналната, но може и да бъде различна.
- ② Кодовите думи могат да бъдат с равен или с различен брой букви.
- ③ Кодовите думи могат да бъдат и еднбуквени.

ИНФО\_05

16/33

## ПРИМЕР: КОД НА ГАЙ ЮЛИЙ ЦЕЗАР

Римският император е водил своите дневници **със секретен код**.

В тях той **заменял всяка буква** на латинската азбука **с тази**, която е **на две позиции след нея** по азбучен ред, т. е. **заменял е А с С, В с D, ..., Х със Z, Y с А и Z с В**.

- ❶ Вторичната азбука **съвпада** с оригиналната.
- ❷ Кодовите думи са **еднобуквени**.

ИНФО\_05

17/33

## УНИВЕРСАЛНА АЗБУКА

Всеки знак от произволна азбука **се различава от останалите по някои свои физически характеристики**.

Щом като **знаковете** от всяка азбука **са краен брой и се различават един от друг**, то те **могат да бъдат наредени и номерирани** (например, **с целите числа 0, 1, 2, ..., n-1 или 1, 2, ..., n**).

По такъв начин **всеки знак се свързва взаимно-однозначно** със съответното му цяло число.

Така **целите неотрицателни числа** се явяват **универсална азбука за кодиране** на съобщения.

ИНФО\_05

18/33

## СЛЕДСТВИЯ

**Недостатък:** Целите числа **се пишат с цифри**, а **поредица от цифри не се разпада еднозначно на поредица от записани с тях числа**.

269368: 23 936 8, 269 368, 26 93 68 ... ?

**Предимство:** Числата могат да се запишат **в двоична ПБС**, т. е. **знаците 0 и 1 са достатъчни** за записване на произволно съобщение.

ИНФО\_05

19/33

## ЕДНОЗНАЧНОСТ НА ДЕКОДИРАНЕТО

- ❶ **Равномерно кодиране:** всички числа се пишат **с еднакъв брой цифри**, наречен **основа на кода**:  
269368 → (2) 26 93 68 или (3) 269 368
- ❷ **Разделител:** между числата се поставя специално **договорен знак**: 26,9,3,68
- ❸ **Префиксни кодове:** кодиращите числа се подбират така, че да **започват с различни поредици от цифри**.

ИНФО\_05

20/33

## КОГА Е НЕОБХОДИМО КОДИРАНЕ?

- ❶ За **защита от неправилен достъп**;
- ❷ За **защита срещу грешки** (със и без възможност за **възстановяване** на оригинала);
- ❸ За **намаляване на размера** на съобщението.

ИНФО\_05

21/33

## ЗАЩИТА СРЕЩУ НЕПРАВОМЕРЕН ДОСТЪП

Използва се когато **информацията** от оригиналното съобщение **не е предназначена за публичен достъп** (ограничава се броя на получателите).

В този случай **алгоритмите за кодиране** (декодиране) трябва да **отговарят на допълнителни условия**.

**Пример е кодът на Цезар**.

ИНФО\_05

22/33

## ПРИМЕР: КОД НА ЦЕЗАР

**Оригинал:** Аз обичам информатиката.

**Код:** ВЙ РГКЩВО КПЦРТОВФКМВФВ.

**Кодова таблица:**

А	Б	В	Г	Д	Е	Ж	З	И	Й
В	Г	Д	Е	Ж	З	И	Й	К	Л
К	Л	М	Н	О	П	Р	С	Т	У
М	Н	О	П	Р	С	Т	У	Ф	Х
Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ь	Ю	Я
Ц	Ч	Ш	Щ	Ъ	Ь	Ю	Я	А	Б

ИНФО\_05

23/33

## ВИДОВЕ ГРЕШКИ ПРИ ПРЕДАВАНЕ

Като **следствие от шума** в канала могат да се получат **два вида грешки**.

Нека **оригиналът е: 010**.

❶ **Получава се знак**, който **не принадлежи на азбуката: 012** (1,5 V).

❷ **Заменя се един знак** от азбуката **с друг: 011 или 000**.

**Защитата срещу шума бива два вида: регистриране и възстановяване**.

ИНФО\_05

24/33

## РЕГИСТРИРАНЕ НА ГРЕШКИТЕ

Основна идея на кодовете:

**01100... 11**  
n произволни

**010...01**  
t изчислени

Методите позволяват да се откриват не повече от предварително определен брой грешки.

ИНФО\_05

25/33

## ПРИМЕР: КОНТРОЛ ПО ЧЕТНОСТ

Регистрира единична грешка:

четен брой единици  
**01100...11 1**  
n произволни +1

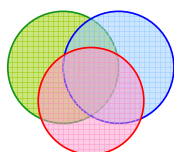
Вместо четен броят на единиците може да бъде и нечетен. Тогава се говори за контрол по нечетност.

ИНФО\_05

26/33

## КОРИГИРАНЕ НА ГРЕШКИ

Основна идея:



Разбиваме битовете на съобщението на подмножества и за всяко от тях изчисляваме контрол по четност (нечетност).

ИНФО\_05

27/33

## ПРИМЕР: ТАБЛИЧЕН КОНТРОЛ

Оригинал: 110 010 001

1	1	0	0
0	1	0	1
0	0	1	1
1	0	1	0

Код: 1100 0101 0011 1010

ИНФО\_05

28/33

## ПРИМЕР: ВЪЗСТАНОВЯВАНЕ

Получено е: 1100 0100 0011 1010

1	1	0	0	😊
0	1	0	1	😞
0	0	1	1	😊
1	0	1	0	😊
😊	😊	😊	😞	

Оригиналът е: 110 010 001

ИНФО\_05

29/33

## КОМПРЕСИРАНЕ

Цел: Намаляване на размера на съобщението за по-евтино съхраняване или по-бързо предаване.

Обикновено се използва кодиране, зависимо от съдържанието.

Основна идея: Дълги еднакви поредици се заменят с по-къси.

ИНФО\_05

30/33

## ПРИМЕР: ДЪЛГИ ПОРЕДИЦИ ОТ НУЛИ И ОТ ЕДИНИЦИ

ОСНОВНИ ИДЕИ:

- 1 Определяме горна граница на дължината (15 → 4 двоични цифри).
- 2 Редица от еднакви цифри заменяме с нейната дължина.
- 3 Винаги започваме с дължината на поредица от нули.

ИНФО\_05

31/33

## ПРИМЕР: КОМПРЕСИРАНЕ

Оригинал: (42) 0011...100...0111  
2 12 25=15+10 3

$2_{(10)}=0010_{(2)}$ ,  $12_{(10)}=1100_{(2)}$ ,  $15_{(10)}=1111_{(2)}$ ,  
 $0_{(10)}=0000_{(2)}$ ,  $10_{(10)}=1010_{(2)}$ ,  $3_{(10)}=0011_{(2)}$

Код: (24) 0010 1100 1111 0000 1010 0011

Компресия: 24:42 ≈ 0.5714 ≈ 57%

ИНФО\_05

32/33

