

# Глава 1. Теоретические основы информатики

Лекция 1. Информация и общество

Лекция 2. Информация

Лекция 3. Представление информации

Лекция 4. Информационные процессы

Лекция 5. ЭВМ — техническое средство информатики

# Глава 1. Теоретические основы информатики

## Лекция 4. Информационные процессы

1.4.1. Хранение данных

1.4.2. Обработка данных (кодирование)

1.4.3. Передача данных

# Типы информационных процессов

Изменение во времени состояния объекта, процесса или явления в живой природе, обществе или технике носит название **процесса**.

Под **информационными процессами** понимаются любые действия, выполняемые с информацией.



### 1.4.1. Хранение данных

#### Носители информации, виды памяти

**Носители информации** — это физическая среда, непосредственно хранящая информацию.

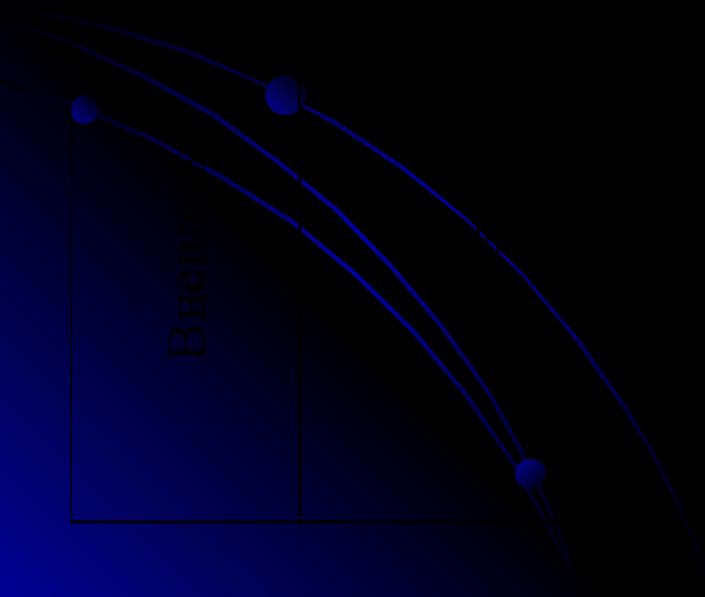
**Внутренняя память** располагается в одной оболочке с прибором, выполняющим вычислительные и логические функции и использующий для этого хранимые данные (например, мозг человека).

Во **внешней памяти (хранилище информации)** данные хранятся за пределами устройства обработки данных (библиотека, жесткий диск ПК) .

## 1.4. Информационные процессы

### 1.4.1. Хранение данных

Физические свойства и структурная организация памяти



## 1.4. Информационные процессы

# 1.4.1. Хранение данных

## Внутренняя память

**Внутренняя память представляет собой совокупность отдельных ячеек:**

- каждая ячейка памяти имеет свой собственный адрес;
- ячейка памяти предназначена для хранения нескольких байтов информации;
- ячейка памяти никогда не бывает пустой, но ее содержимое для данной программы может быть лишено смысла;
- при сохранении в ячейке памяти новой информации ее предыдущее содержимое теряется.

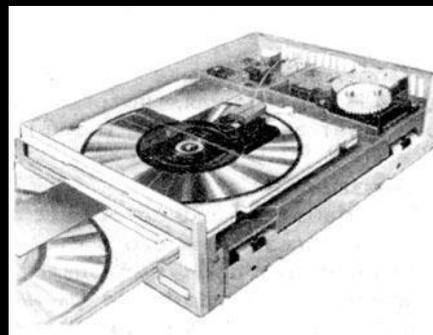
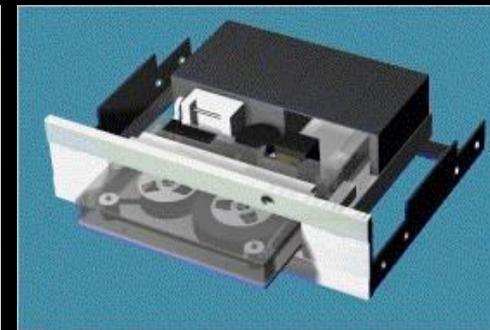
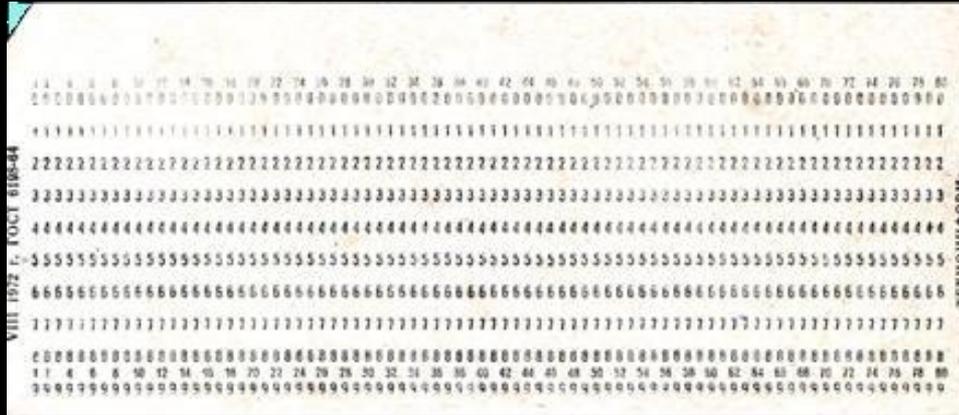


**ОЗУ** представляет собой память с произвольным доступом, которая обеспечивает хранение на время работы программу и данные, которыми она манипулирует. В силу этого ОЗУ еще называют временной памятью, подчеркивая, что содержимое этой памяти при выключении компьютера теряется.

**ПЗУ** предназначена для постоянного содержания информации в компьютере, которую можно считывать по мере надобности, но нельзя пополнять в ходе выполнения программы. Обычно в ПЗУ содержатся инструкции для приведения компьютера в рабочее состояние после включения.

# 1.4. Информационные процессы

## 1.4.1. Хранение данных Внешняя память



## 1.4.1. Хранение данных

Основные характеристики процесса хранения данных

1. Объем.
2. Надежность.
3. Время доступа.
4. Наличие защиты.



## 1.4.2. Обработка данных (кодирование)

### Общие сведения



Процесс перехода от исходных данных к результату и есть **процесс обработки данных**.

Тот объект (субъект), который осуществляет обработку, называется **исполнителем обработки** (может быть человек или техническое устройство, например, компьютер).

## 1.4.2. Обработка данных (кодирование)

### Цели преобразования данных

**1. Получение новой информации, нового знания**

**2. Изменение формы данных, но не содержания**

**2.1. Перевод текста с одного языка на другой**

**2.2. Кодирование — преобразование информации в форму удобную для ее хранения, передачи и обработки**

**2.3. Структурирование данных**

**2.4. Поиск**

# 1.4.2. Обработка данных (кодирование)

## Кодирование данных



# 1.4.2. Обработка данных (кодирование)

## Кодирование целых положительных чисел

Целые числа хранятся в двоичной системе счисления в форме с фиксированной запятой (естественный вид)

Положительные числа

Прямой код

$$120_{(10)} = 1111000_{(2)}$$

Длина	Кодировка
1 байт	01111000 (78)
2 байта (машинное слово)	00000000 01111000 (0078)
4 байта (двойное машинное слово)	00000000 00000000 00000000 01111000 (00000078)

# 1.4.2. Обработка данных (кодирование)

## Кодирование целых отрицательных чисел

Целые числа хранятся в двоичной системе счисления в форме с фиксированной запятой (естественный вид)

Отрицательные числа

Дополнительный код

$$-120_{(10)} = -1111000_{(2)}$$

01111000  
 10000111  
 10001000

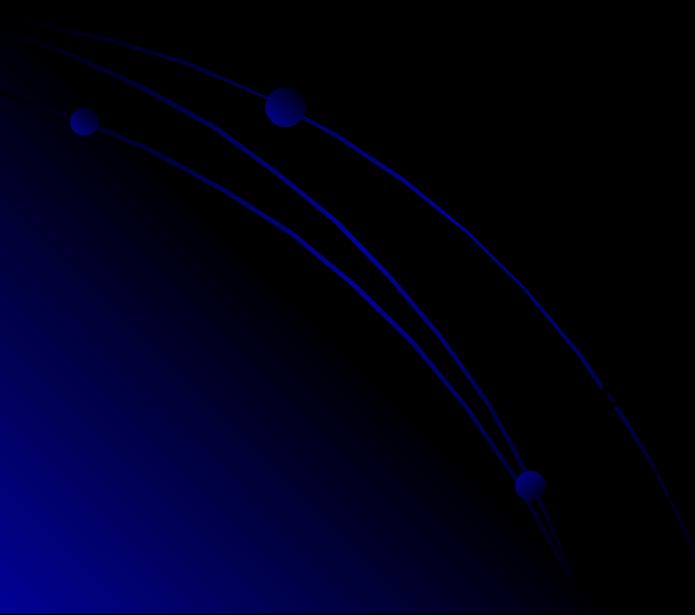
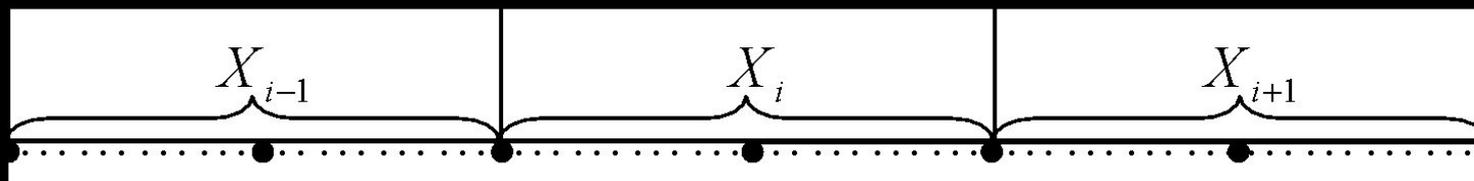
Замена  
 +  
 1

Длина	Кодировка
1 байт	10001000 (88)
2 байта (машинное слово)	11111111 10001000 (FF 88)
4 байта (двойное машинное слово)	11111111 11111111 11111111 10001000 (FF FF FF 88)

## 1.4. Информационные процессы

# 1.4.2. Обработка данных (кодирование)

## Представление вещественных чисел



# 1.4.2. Обработка данных (кодирование)

## Кодирование вещественных чисел

Вещественные числа хранятся в двоичной системе счисления в форме с плавающей запятой (нормализованный вид)

Нормализованный вид  $X_{(P)} = M_{(P)} \cdot P^k$

где  $M$  — мантисса числа,

$P$  — основание системы счисления,

$k$  — порядок числа

*Формат хранения 4-х байтного вещественного числа*

31	30	29	28	27	26	25	24	23	.....	1	0
1	0	0	1	0	1	0	1	1	.....	0	1

Знак порядка

Знак числа

# 1.4.2. Обработка данных (кодирование)

## Кодирование символьной информации

**Таблица кодировки** — это стандарт, ставящий в соответствие каждому символу алфавита свой порядковый номер.

**Двоичный код символа** — это его порядковый номер в двоичной системе счисления.

Однobaйтная таблица ASCII:

- символы с кодами от 0 до 31 используются в качестве управляющих кодов производителями компьютеров;
- символы с кодами от 32 до 127 являются стандартной кодировкой ASCII, включающей коды символов английского алфавита, знаки препинания, цифры, знаки арифметических действий и некоторые вспомогательные символы;
- символы с кодами от 128 до 255 отданы для создания в каждой стране своего стандарта.

# 1.4.2. Обработка данных (кодирование)

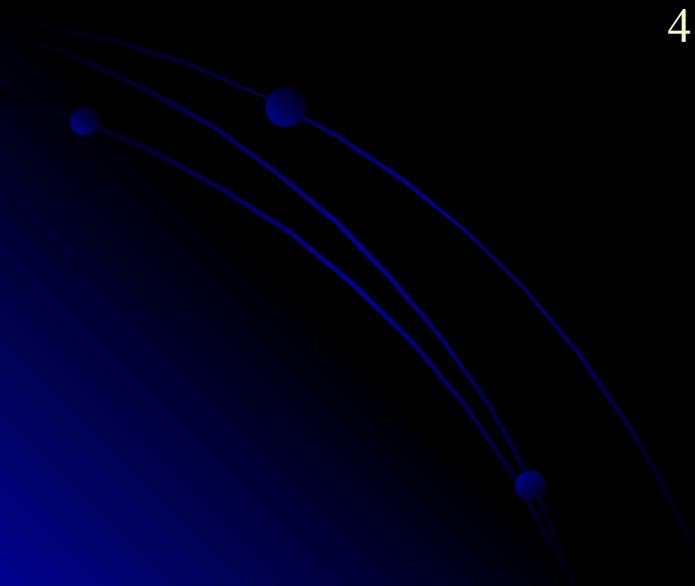
## Кодировка ASCII: общая часть

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
<b>00</b>	00 □	01 □	02 □	03 □	04 □	05 □	06 □	07 □	08 □	09 □	0A □	0B □	0C □	0D □	0E □	0F □
<b>10</b>	10 □	11 □	12 □	13 □	14 □	15 □	16 □	17 □	18 □	19 □	1A □	1B □	1C □	1D □	1E □	1F □
<b>20</b>	20	21 !	22 "	23 #	24 \$	25 %	26 &	27 (	28 )	29 '	2A *	2B +	2C ,	2D -	2E .	2F /
<b>30</b>	30 0	31 1	32 2	33 3	34 4	35 5	36 6	37 7	38 8	39 9	3A :	3B ;	3C <	3D =	3E >	3F ?
<b>40</b>	40 @	41 A	42 B	43 C	44 D	45 E	46 F	47 G	48 H	49 I	4A J	4B K	4C L	4D M	4E N	4F O
<b>50</b>	50 P	51 Q	52 R	53 S	54 T	55 U	56 V	57 W	58 X	59 Y	5A Z	5B [	5C \	5D ]	5E ^	5F _
<b>60</b>	60 `	61 a	62 b	63 c	64 d	65 e	66 f	67 g	68 h	69 i	6A j	6B k	6C l	6D m	6E o	6F p
<b>70</b>	70 p	71 q	72 r	73 s	74 t	75 u	76 v	77 w	78 x	79 y	7A z	7B {	7C 	7D }	7E ~	7F □

## 1.4.2. Обработка данных (кодирование)

Варианты кодировки второй половины ASCII для России

1. Кодировка MS-DOS CP 866
2. Windows CP1251
3. Кодировка UNIX KOI8-R
4. Кодировка ISO 88-59-5



# 1.4. Информационные процессы

## 1.4.2. Обработка данных (кодирование)

### Кодировка MS-DOS CP 866

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
80	410 А	411 Б	412 В	413 Г	414 Д	415 Е	416 Ж	417 З	418 И	419 Й	41A К	41B Л	41C М	41D Н	41E О	41F П
90	420 Р	421 С	422 Т	423 У	424 Ф	425 Х	426 Ц	427 Ч	428 Ш	429 Щ	42A Ъ	42B Ы	42C Ь	42D Э	42E Ю	42F Я
A0	430 а	431 б	432 в	433 г	434 д	435 е	436 ж	437 з	438 и	439 й	43A к	43B л	43C м	43D н	43E о	43F п
B0	2591 	2592 	2593 	2502 	2524 ┆	2561 ≡	2562 ≡	2556 ┆	2555 ┆	2563 ≡	2551 	2557 ┆	255D ┆	255C ┆	255B ┆	2510 ┆
C0	2514 L	2534 ┆	252C ┆	251C ┆	2500 —	253C ┆	255E ┆	255F ┆	255A ┆	2554 ┆	2569 ┆	2569 ┆	2560 ┆	2560 ┆	2560 ┆	2567 ┆
D0	2568 ┆	2564 ┆	2565 ┆	2559 ┆	2558 ┆	2552 ┆	2553 ┆	256B ┆	256A ┆	2518 ┆	250C ┆	2588 	2584 	258C 	2590 	2580 
E0	440 р	441 с	442 т	443 у	444 ф	445 х	446 ц	447 ч	448 ш	449 щ	44A ъ	44B ы	44C ь	44D э	44E ю	44F я
F0	401 Ё	451 ё	404 Є	454 є	407 Ӏ	457 ӓ	40E ӱ	45E ӱ	B0 °	2219 •	B7 ·	221A √	2116 №	A4 #	25A0 ■	A0

## 1.4. Информационные процессы

# 1.4.2. Обработка данных (кодирование)

## Кодировка Windows CP1251

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
<b>80</b>	402 Ѡ	403 ѡ	201A ,	453 ѓ	201E ,,	2026 ...	2020 †	2021 ‡	20AC €	2030 ‰	409 Љ	2039 ‹	40A Њ	40C Ќ	40B Ѣ	40F Ѥ
<b>90</b>	452 ђ	2018 ‘	2019 ’	201C “	201D ”	2022 •	2013 –	2014 —		2122 ™	459 љ	203A ›	45A њ	45C ќ	45B ћ	45F ѥ
<b>A0</b>	A0 	40E Ў	45E ў	408 Ј	A4 Ѧ	490 ѓ	A6 і	A7 ѕ	401 Ё	A9 ©	404 Є	AB «	AC ¬	AD -	AE ®	407 ї
<b>B0</b>	B0 °	B1 ±	406 І	456 і	491 ѓ	B5 μ	B6 ¶	B7 ·	451 ё	2116 №	454 є	BB »	458 ј	405 Ѕ	455 ѕ	457 ї
<b>C0</b>	410 А	411 Б	412 В	413 Г	414 Д	415 Е	416 Ж	417 З	418 И	419 Й	41A К	41B Л	41C М	41D Н	41E О	41F П
<b>D0</b>	420 Р	421 С	422 Т	423 У	424 Ф	425 Х	426 Ц	427 Ч	428 Ш	429 Щ	42A Ъ	42B Ы	42C Ь	42D Э	42E Ю	42F Я
<b>E0</b>	430 а	431 б	432 в	433 г	434 д	435 е	436 ж	437 з	438 и	439 й	43A к	43B л	43C м	43D н	43E о	43F п
<b>F0</b>	440 р	441 с	442 т	443 у	444 ф	445 х	446 ц	447 ч	448 ш	449 щ	44A ъ	44B ы	44C ь	44D э	44E ю	44F я

## 1.4. Информационные процессы

# 1.4.2. Обработка данных (кодирование)

## Кодировка UNIX KOI8-R

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
80	2500 —	2502 	250C Г	2510 └	2514 L	2518 J	251C ┌	2524 └	252C T	2534 └	253C ┌	2580 ■	2584 ■	2588 ■	258C ■	2590 ■
90	2591 ░	2592 ░	2593 ░	2320 	25A0 ■	2219 •	221A √	2248 ≈	2264 ≤	2265 ≥	A0	2321 	B0 °	B2 2	B7 .	F7 ÷
A0	2550 =	2551 	2552 F	451 ë	2553 Г	2554 Г	2555 Г	2556 Г	2557 Г	2558 L	2559 L	255A L	255B J	255C J	255D J	255E ┌
B0	255F ┌	2560 ┌	2561 ┌	401 Ë	2562 ┌	2563 ┌	2564 ┌	2565 ┌	2566 ┌	2567 ┌	2568 ┌	2569 ┌	256A ┌	256B ┌	256C ┌	A9 ©
C0	44E ю	430 а	431 б	446 ц	434 д	435 е	444 ф	433 г	445 х	438 и	439 й	43A к	43B л	43C м	43D н	43E о
D0	43F п	44F я	440 р	441 с	442 т	443 у	436 ж	432 в	44C ь	44B ы	437 з	448 ш	44D э	449 щ	447 ч	44A ъ
E0	42E Ю	410 А	411 Б	426 Ц	414 Д	415 Е	424 Ф	413 Г	425 Х	418 И	419 Й	41A К	41B Л	41C М	41D Н	41E О
F0	41F П	42F Я	420 Р	421 С	422 Т	423 У	416 Ж	412 В	42C Ь	42B Ы	417 З	428 Ш	42D Э	429 Щ	427 Ч	42A Ъ

# 1.4.2. Обработка данных (кодирование)

## Кодировка ISO 88-59-5

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
<b>80</b>	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
<b>90</b>	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
<b>A0</b>	A0	401 Ё	402 Ъ	403 Ѓ	404 Є	405 Ѕ	406 І	407 Ї	408 Ј	409 Љ	40A Њ	40B Ћ	40C Ќ	AD -	40E Ў	40F Џ
<b>B0</b>	410 А	411 Б	412 В	413 Г	414 Д	415 Е	416 Ж	417 З	418 И	419 Й	41A К	41B Л	41C М	41D Н	41E О	41F П
<b>C0</b>	420 Р	421 С	422 Т	423 У	424 Ф	425 Х	426 Ц	427 Ч	428 Ш	429 Щ	42A Ъ	42B Ы	42C Ь	42D Э	42E Ю	42F Я
<b>D0</b>	430 а	431 б	432 в	433 г	434 д	435 е	436 ж	437 з	438 и	439 й	43A к	43B л	43C м	43D н	43E о	43F п
<b>E0</b>	440 р	441 с	442 т	443 у	444 ф	445 х	446 ц	447 ч	448 ш	449 щ	44A ъ	44B ы	44C ь	44D э	44E ю	44F я
<b>F0</b>	2116 №	451 ё	452 ђ	453 ѓ	454 є	455 ѕ	456 і	457 ї	458 ј	459 љ	45A њ	45B ћ	45C ќ	A7 §	45E ў	45F џ

## 1.4. Информационные процессы

# 1.4.2. Обработка данных (кодирование)

## Двухбайтная система кодирования UNICODE

65536 СИМВОЛОВ

European Alphabets	African Scripts	Indic Scripts	East Asian Scripts	Central Asian Scripts
(see also <b>Comb. Marks</b> )	<b>Ethiopic</b>	Bengali	<b>Han Ideographs</b>	Kharoshthi
<b>Armenian</b>	Ethiopic	Devanagari	Unified CJK Ideographs (5MB)	Mongolian
Armenian	Ethiopic Supplement	Gujarati	CJK Ideographs Ext. A (2MB)	Phags-Pa
<i>Armenian Ligatures</i>	Ethiopic Extended	Gurmukhi	CJK Ideographs Ext. B (13MB)	Tibetan
<b>Coptic</b>	<b>Other African scripts</b>	Kannada	Compatibility Ideographs (.5MB)	
Coptic	N'Ko	Limbu	... Supplement (.5MB)	
<i>Coptic in Greek block</i>	Tifinagh	Malayalam	Kanbun	
<b>Cyrillic</b>	<b>Middle Eastern Scripts</b>	Oriya	(see also <b>Unihan Database</b> )	<b>Ancient Scripts</b>
Cyrillic	<b>Arabic</b>	Sinhala	<b>Radicals and Strokes</b>	<b>Ancient Greek</b>
Cyrillic Supplement	Arabic	Syloti Nagri	CJK Radicals	Ancient Greek Numbers
<b>Georgian</b>	Arabic Supplement	Tamil	KangXi Radicals	Ancient Greek Musical
Georgian	Arabic Presentation Forms A	Telugu	CJK Strokes	<b>Cuneiform</b>
Georgian Supplement	Arabic Presentation Forms B		Ideographic Description	Cuneiform
<b>Greek</b>	<b>Hebrew</b>	<b>Philippine Scripts</b>	<b>Chinese-specific</b>	Cuneiform Numbers
Greek	Hebrew	Buhid	Bopomofo	Old Persian
Greek Extended	<i>Hebrew Presentation Forms</i>	Hanunoo	Bopomofo Extended	Ugaritic
(see also <b>Ancient Greek</b> )	<b>Syriac</b>	Tagalog	<b>Japanese-specific</b>	<b>Linear B</b>
<b>Latin</b>	Syriac	Tagbanwa	Hiragana	Linear B Syllabary
Basic Latin	<b>Thaana</b>		Katakana,	Linear B Ideograms
Latin-1	Thaana	<b>South East Asian</b>	Katakana Phonetic Ext.	<b>Other Ancient Scripts</b>
Latin Extended A	<b>American scripts</b>	Buginese	<i>Halfwidth Katakana</i>	Aegean Numbers
Latin Extended B	Canadian Syllabics	Balinese	<b>Korean-specific</b>	Counting Rod Numerals
Latin Extended C	Cherokee	Khmer	Hangul Syllables (4MB)	Cypriot Syllabary
Latin Extended D	Deseret	Khmer Symbols	Hangul Jamo	Gothic
Latin Extended Additional	<b>Other Scripts</b>	Lao	Hangul Compatibility Jamo	Old Italic
<i>Latin Ligatures</i>	Shavian	Myanmar	<i>Halfwidth Jamo</i>	Ogham
<i>Fullwidth Latin Letters</i>	Osmanya	New Tai Lue	<b>Yi</b>	Runic
Small Forms	Glagolitic	Tai Le	Yi (.6MB)	Phoenician
(see also <b>Phonetic Symbols</b> )		Thai	Yi Radicals	

## 1.4.2. Обработка данных (кодирование)

### Задание для самостоятельной работы

1. Выписать свою фамилию
2. Закодировать ее в соответствии со стандартом MS-DOS CP 866
3. Определить как она будет отображаться на компьютере, если на нем установлена кодовая страница:
  - а) Windows CP1251,
  - б) UNIX KOI8-R,
  - в) ISO 88-59-5?
4. Определить правильную кодировку своей фамилии для кодовых страниц:
  - а) Windows CP1251,
  - б) UNIX KOI8-R,
  - в) ISO 88-59-5.

## 1.4. Информационные процессы

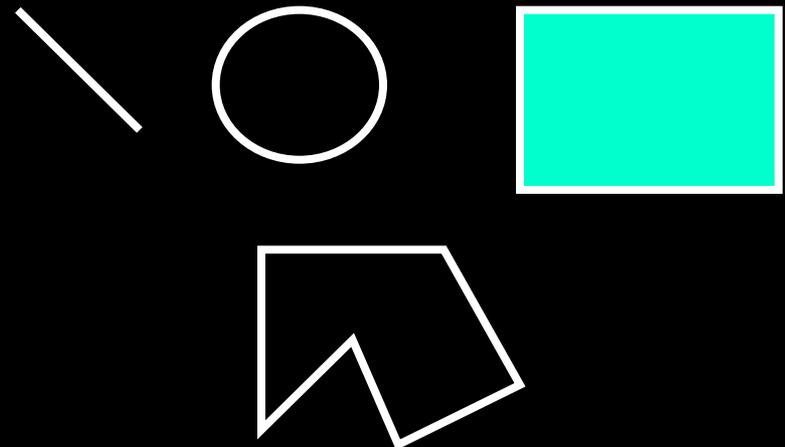
# 1.4.2. Обработка данных (кодирование)

## Кодирование графической информации

**Растровая графика** предполагает разбиение изображения на маленькие одноцветные элементы — видеопиксели, которые, сливаясь, дают общую картину.



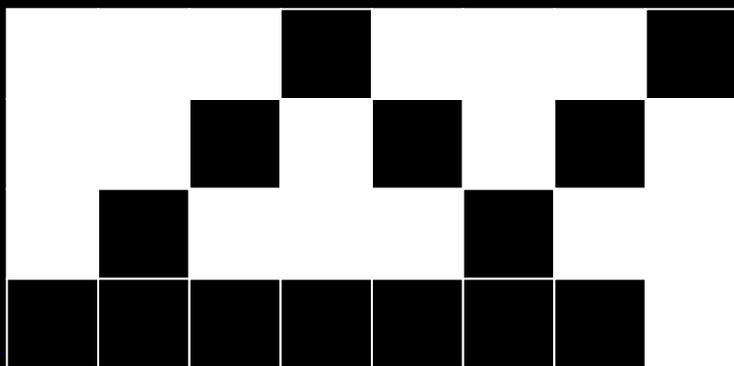
**Векторная графика** изображение рассматривается как совокупность простых элементов: прямых линий, дуг, окружностей, эллипсов, прямоугольников, закрасок и других, которые называются графическими примитивами.



# 1.4.2. Обработка данных (кодирование)

## Черно-белое графическое изображение

Каждая точка изображения кодируется битом:



Цвета

Белый	1
Черный	0

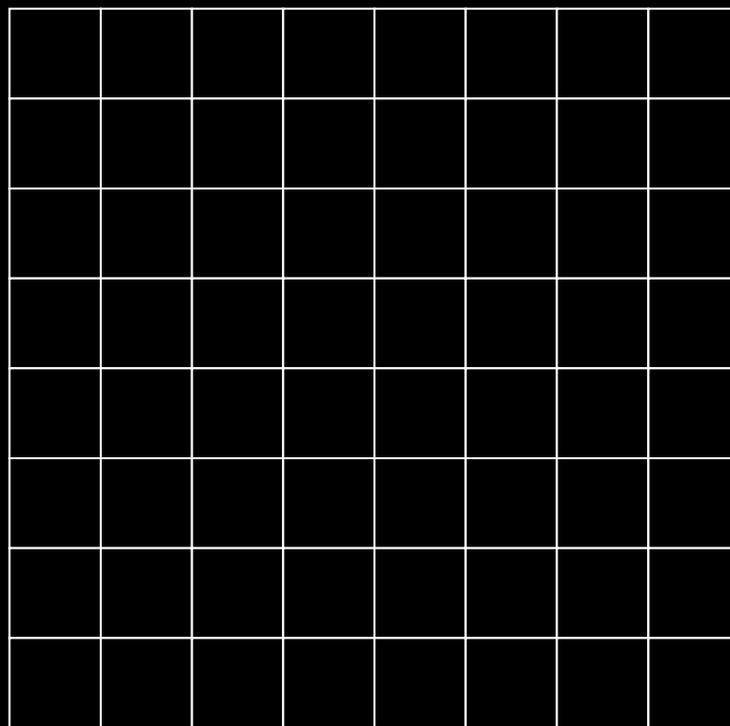
1 1 1 0 1 1 1 0  
1 1 0 1 0 1 0 1  
1 0 1 1 1 0 1 1  
0 0 0 0 0 0 0 1

Кодировка: EE D5 BB 01

## 1.4.2. Обработка данных (кодирование)

### Задание для самостоятельной работы

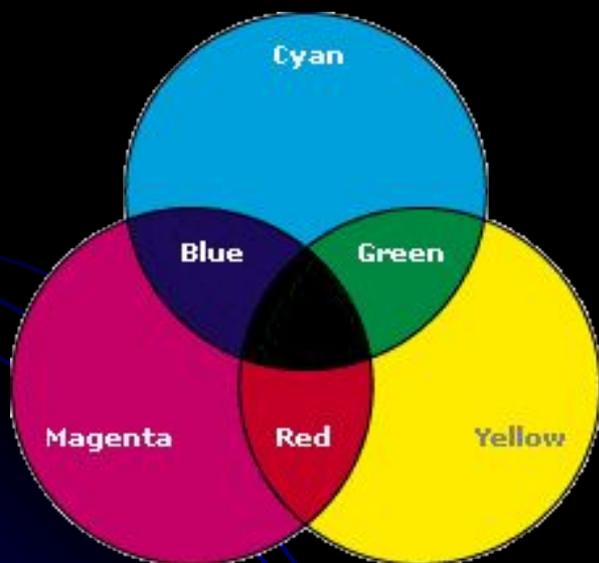
1. Нарисовать сетку  $8 \times 8$
2. Изобразить на сетки первую букву вашей фамилии (белыми квадратиками).
3. Записать двоичную кодировку черно-белого изображения.
4. Записать шестнадцатеричную кодировку изображения.
5. Какой размер памяти требуется для хранения такого изображения?



# 1.4.2. Обработка данных (кодирование)

Цветное (3-х битное) графическое изображение

Каждая точка изображения кодируется 3 битами:



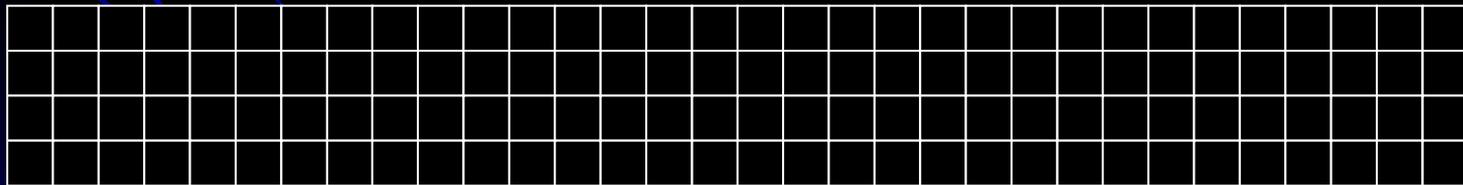
Цвета

Белый	111
Красный	100
Зеленый	010
Синий	001
Голубой	011
Пурпурный	101
Желтый	110
Черный	000

## 1.4.2. Обработка данных (кодирование)

### Задание для самостоятельной работы

1. Нарисовать сетку  $24 \times 24$
2. Изобразить на сетке произвольный цветной рисунок.
3. Записать двоичную кодировку цветного изображения.
4. Записать шестнадцатеричную кодировку изображения.
5. Какой размер памяти требуется для хранения такого изображения?



## 1.4.2. Обработка данных (кодирование)

Цветное (24-х битное) графическое изображение

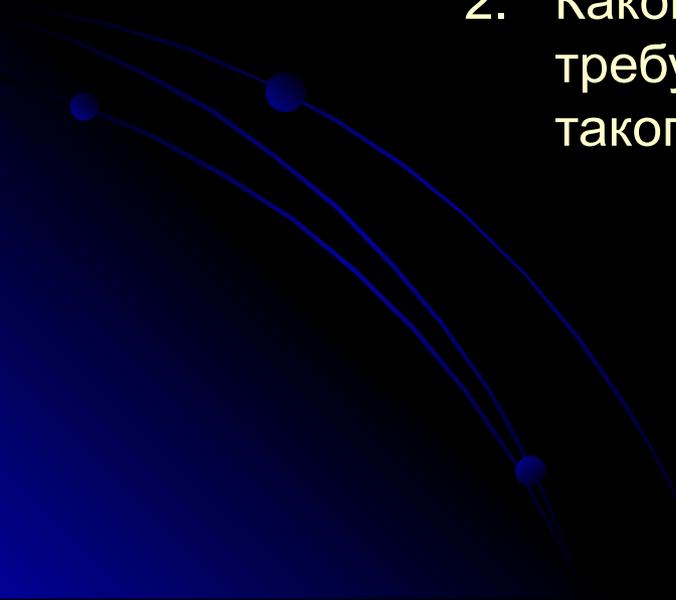
Каждая точка изображения кодируется 3 байтами:

Цвета

Белый	FF FF FF
Красный	FF 00 00
Зеленый	00 FF 00
Синий	00 00 FF
Голубой	00 FF FF
Пурпурный	FF 00 FF
Желтый	FF FF 00
Черный	00 00 00

## 1.4.2. Обработка данных (кодирование)

### Задание для самостоятельной работы

1. Записать шестнадцатеричную кодировку ранее нарисованного изображения в True Color
  2. Какой размер памяти требуется для хранения такого изображения?
- 

# 1.4.2. Обработка данных (кодирование)

## Кодирование звуковой информации



# 1.4.2. Обработка данных (кодирование)

## Кодирование звуковой информации

**Аудиоадаптер (звуковая плата)** — специальное устройство, подключаемое к компьютеру, предназначенное для преобразования электрических колебаний звуковой частоты в числовой двоичный код при вводе звука и для обратного преобразования (из числового кода в электрические колебания) при воспроизведении звука.

**Частота дискретизации** — это количество измерений входного сигнала за 1 секунду.

**Разрядность регистра** — число бит в регистре аудиоадаптера.

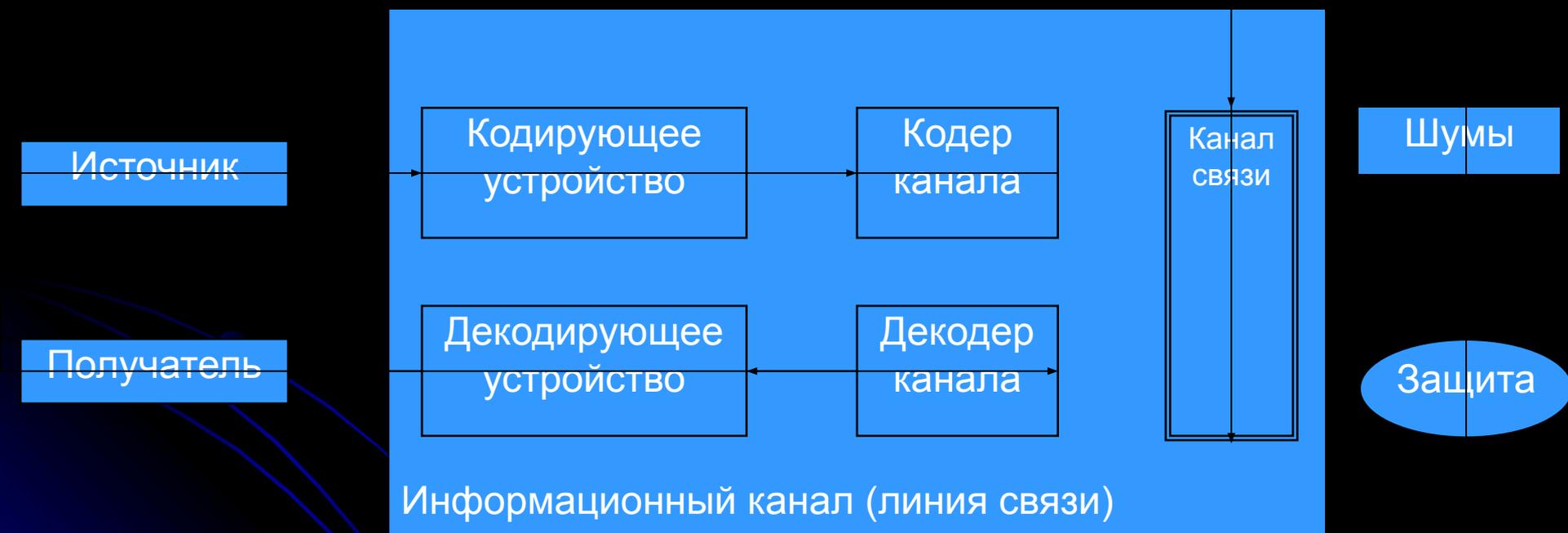
**Звуковой файл** — файл, хранящий звуковую информацию в числовой двоичной форме.

# 1.4.3. Передача данных

## Общая схема передачи данных

При передаче информации через линию связи приходится решать две проблемы:

- согласование метода кодирования информации с характеристиками канала связи;
- защиту передаваемой информации от возможных искажений.



Методы защиты от помех:

- экранирование электрических линий связей;
- улучшение избирательности приемного устройства;
- использование специальных методов кодирования информации.

# 1.4.3. Передача данных

## Характеристика канала связи

### 1. Ширина полосы пропускания

Интервал частот между максимальной частотой  $f_{\max}$  и минимальной частотой  $f_{\min}$ , используемый данным каналом связи для передачи сигналов с допустимыми искажениями, называется **шириной полосы пропускания**  $\Delta f = f_{\max} - f_{\min}$ .

Как правило  $f_{\min} = 0$ ,  $\Delta f = f_{\max}$

Согласно теореме Котельникова В. А.

$$\Delta t \leq 1/(2f_{\max})$$

## 1.4. Информационные процессы

# 1.4.3. Передача данных

## Характеристика канала связи

### 2. Пропускная способность канала связи

**Пропускная способность канала связи** — это среднее количество информации, передаваемое по каналу связи за единицу времени.

Если с передачей одного импульса связано количество информации  $I_0$ , а передается оно за время  $\tau_0$ , то отношение  $I_0$  к  $\tau_0$ , очевидно, будет отражать среднее количество информации, передаваемое по каналу за единицу времени:

$$C = I_0 / \tau_0$$

## 1.4.3. Передача данных

### Характеристика канала связи

#### 3. Скорость передачи информации

**Скорость передачи информации** — это отношение объема переданной информации к времени ее передачи.

Пусть по каналу связи за время  $t$  передано количество информации  $I$ . Тогда — скорость передачи информации  $J = I/t$  .  
Максимальная скорость передачи информации равна его пропускной способности.

### 1.4.3. Передача данных

#### Влияние шумов на пропускную способность информационного канала связи

**Шум** представляет собой разного рода помехи, искажающие передаваемый сигнал и приводящий к потере информации.

Рассмотрим частный случай передачи двух элементарных сигналов равной длительности. Каждый из них на входе канала связи несет  $I = 1$  бит информации. Из-за шумов при передаче может произойти искажение сигнала, в результате чего вместо 1 на выходе будет получен 0, а вместо 0 — 1. Пусть вероятности таких искажений для 0 и 1 равны  $p_{1 \rightarrow 0} = p_{0 \rightarrow 1} = p$ . Тогда вероятность того, что исходный сигнал придет без искажений, очевидно, равна  $1 - p$ . Следовательно, при интерпретации (распознавании) конечного сигнала возникает неопределенность, которая, может быть охарактеризована средней энтропией:

$$H = -p \cdot \log_2 p - (1 - p) \cdot \log_2 (1 - p)$$

Эта неопределенность приведет к уменьшению количества информации, содержащейся в сигнале, на величину  $H$ , т. е.:

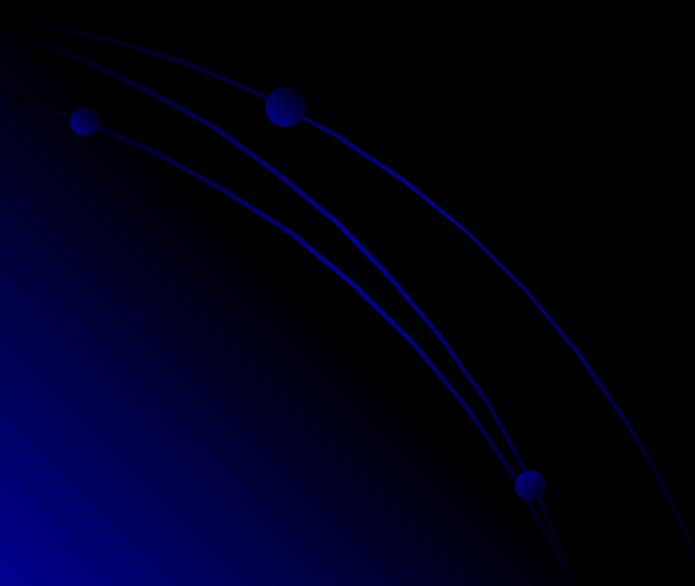
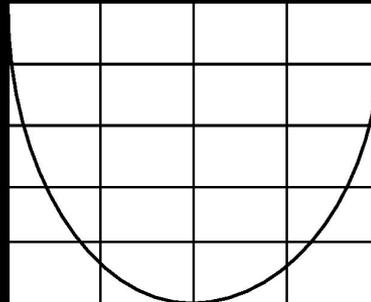
$$\bar{I}' = \bar{I} - H$$

### 1.4.3. Передача данных

#### Зависимость пропускной способности информационного канала от помех

Пропускная способность реального канала  $C_{PK}$  оказывается меньше, чем аналогичного идеального  $C$  :

$$C_{PK} = \bar{I}' / \tau_0 = (\bar{I} + p \cdot \log_2 p + (1 - p) \cdot \log_2 (1 - p)) / \tau_0 \leq C$$



### 1.4.3. Передача данных

#### Обеспечение надежности передачи данных

При помехоустойчивом кодировании сообщения для канала связи могут ставиться две задачи:

- кодирование обеспечивает только установление факта искажения информации — в этом случае исправление производится путем ее повторной передачи;
- кодирование позволяет локализовать и автоматически исправить ошибку передачи данных.

Надежность передачи при кодировании помехоустойчивым кодом обеспечивается тем, что вместе с разрядами сообщения  $K_c$  передается контрольных разрядов  $K_k$ . Избыточность сообщения для реального канала характеризуется относительной величиной:  $L = 1 + K_k / K_c$

**Избыточность сообщения** — это характеристика, показывающая, во сколько раз требуется удлинить сообщение, чтобы обеспечить его надежную (безошибочную) передачу (хранение).

## 1.4.3. Передача данных

### Обеспечение надежности передачи данных

#### 1. Бит четности

Кодовая цепочка должна содержать четное количество единиц

Информационный байт	Бит четности
01010100	1
11011011	0

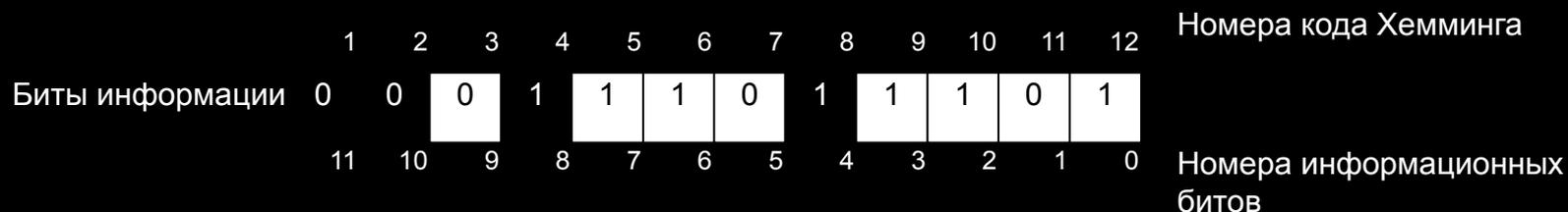
Избыточность сообщения  $L = 1 + 1/K_c$

## 1.4.3. Передача данных

### Обеспечение надежности передачи данных

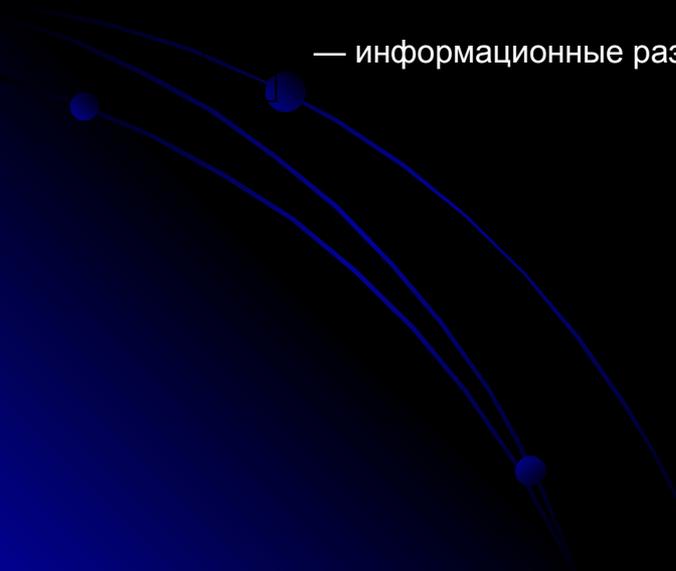
#### 1. Код Хемминга

Основная идея кода Хемминга состоит в добавлении к информационным битам нескольких битов четности, каждый из которых контролирует определенные информационные биты.



— информационные разряды

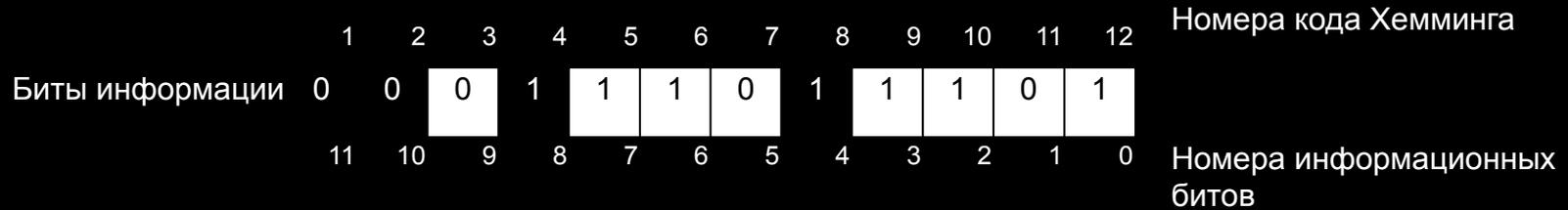
— контрольные разряды



# 1.4.3. Передача данных

## Обеспечение надежности передачи данных

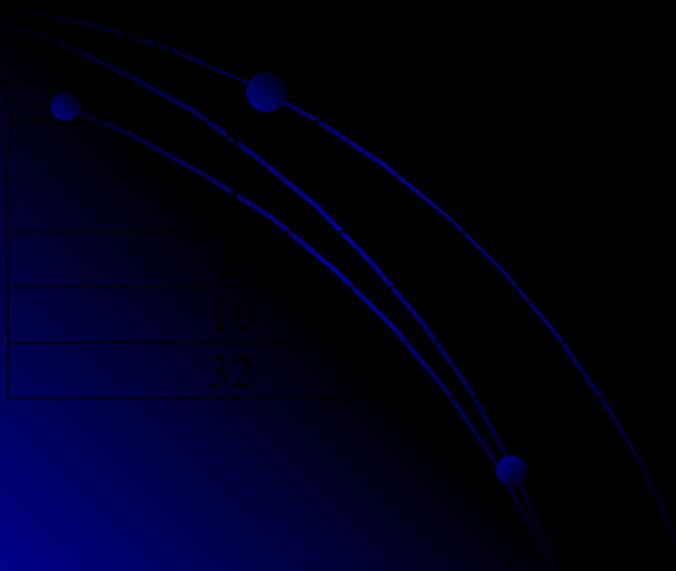
### 2. Код Хемминга



— информационные разряды

— контрольные разряды

Номера контролируемых битов проверочными



## 1.4.3. Передача данных

### Обеспечение надежности передачи данных

Алгоритм кодирования информации с помощью кода Хемминга состоит из 4-х шагов:

1. Добавление  $k+1$  проверочных битов (в позициях всех степеней числа 2) к передаваемым данным и их нумерация слева направо, начиная с 1.
2. Выбор в качестве определяемого бита бит с номером  $2^{k+1}$ .
3. Присваивание определяемому проверочному биту сумму по модулю 2 контролируемых им битов, кроме себя самого.
4. Уменьшение  $k$  на 1.
5. Выполнение шагов 2–4 до тех пор, пока  $k \geq -1$ .

#### Пример

Закодировать по коду Хемминга сообщение 01101101.

1. 01101101 → \_ \_ 0 \_ 1 1 0 \_ 1 1 0 1.
2. \_ \_ 0 1 1 0 \_ 1 1 0 1 → \_ \_ 0 \_ 1 1 0 □ 1 1 0 1 →  
→ \_ \_ 0 \_ 1 1 0 1 1 1 0 1, т. к.  $1 \oplus 1 \oplus 0 \oplus 1 = 1$ .
3. \_ \_ 0 1 1 0 1 1 1 0 1 → \_ \_ 0 □ 1 1 0 1 1 1 0 1 →  
→ \_ \_ 0 1 1 1 0 1 1 1 0 1, т. к.  $1 \oplus 1 \oplus 0 \oplus 1 = 1$ .
4. \_ \_ 0 1 1 1 0 1 1 1 0 1 → \_ □ 0 1 1 1 0 1 1 1 0 1 →  
→ \_ 0 0 1 1 1 0 1 1 1 0 1, т. к.  $0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 0$ .
5. \_ 0 0 1 1 1 0 1 1 1 0 1 → □ 0 0 1 1 1 0 1 1 1 0 1 →  
→ 0 0 0 1 1 1 0 1 1 1 0 1, т. к.  $0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 0$ .

Сообщение 01101101 кодом Хемминга будет закодировано как 000111011101.

## 1.4.3. Передача данных

### Обеспечение надежности передачи данных

Алгоритм кодирования информации с помощью кода Хемминга состоит из 4-х шагов:

1. Добавление  $k+1$  проверочных битов (в позициях всех степеней числа 2) к передаваемым данным и их нумерация слева направо, начиная с 1.
2. Выбор в качестве определяемого бита бит с номером  $2^{k+1}$ .
3. Присваивание определяемому проверочному биту сумму по модулю 2 контролируемых им битов, кроме себя самого.
4. Уменьшение  $k$  на 1.
5. Выполнение шагов 2–4 до тех пор, пока  $k \geq -1$ .

#### Пример

Закодировать по коду Хемминга сообщение 01101101.

1. 01101101 → \_ \_ 0 \_ 1 1 0 \_ 1 1 0 1.
2. \_ \_ 0 1 1 0 \_ 1 1 0 1 → \_ \_ 0 \_ 1 1 0 □ 1 1 0 1 →  
→ \_ \_ 0 \_ 1 1 0 1 1 1 0 1, т. к.  $1 \oplus 1 \oplus 0 \oplus 1 = 1$ .
3. \_ \_ 0 1 1 0 1 1 1 0 1 → \_ \_ 0 □ 1 1 0 1 1 1 0 1 →  
→ \_ \_ 0 1 1 1 0 1 1 1 0 1, т. к.  $1 \oplus 1 \oplus 0 \oplus 1 = 1$ .
4. \_ \_ 0 1 1 1 0 1 1 1 0 1 → \_ □ 0 1 1 1 0 1 1 1 0 1 →  
→ \_ 0 0 1 1 1 0 1 1 1 0 1, т. к.  $0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 0$ .
5. \_ 0 0 1 1 1 0 1 1 1 0 1 → □ 0 0 1 1 1 0 1 1 1 0 1 →  
→ 0 0 0 1 1 1 0 1 1 1 0 1, т. к.  $0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 0$ .

Сообщение 01101101 кодом Хемминга будет закодировано как 000111011101.

## 1.4.3. Передача данных

### Обеспечение надежности передачи данных

Избыточность кодов Хемминга



## 1.4.3. Передача данных

### Обеспечение надежности передачи данных

Алгоритм проверки и исправления сообщений передаваемых в коде Хемминга

1. Производится проверка всех битов четности.
2. Если все биты четности верны, то осуществляется переход к п. 5.
3. Вычисление суммы номеров всех неправильных битов четности.
4. Инвертация содержимого бита, номер которого равен сумме, найденной в п. 3.
5. Исключение битов четности, передача правильного информационного кода.

#### Пример

Дешифровать код Хемминга для сообщения 000101011101.

1. Первый бит четности неверен, т. к.  $0 \oplus 0 \oplus 1 \oplus 1 = 1 \neq 0$ .

Второй бит четности верен, т. к.  $0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 0$

Четвертый бит четности неверен, т. к.  $0 \oplus 1 \oplus 0 \oplus 1 = 0 \neq 1$

Восьмой бит четности верен, т. к.  $1 \oplus 1 \oplus 0 \oplus 1 = 1$

2. В сообщении 000101011101 содержится ошибка на которую указывают первый и четвертый биты четности.

3. Сумма номеров неправильных битов четности  $5=1+4$ .

4.  $0001\underline{0}1011101 \rightarrow 0001\underline{1}1011101$ .

5.  $\underline{0}\underline{0}0\underline{1}110\underline{1}1101 \rightarrow 01101101$ .

Сообщение 000101011101 кода Хемминга дешифровано как 01101101.

### 1.4.3. Передача данных

#### Задание для самостоятельной работы

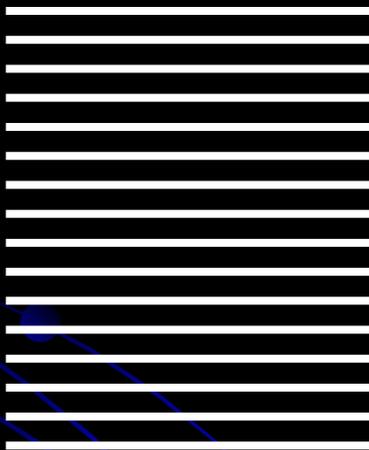
1. Выписать слово длиной 5-10 букв
2. Закодировать его в соответствии со стандартом MS-DOS CP 866
3. Перекодировать двоичный код слова кодом Хемминга, так чтобы в каждом пакете информации информационных битов было 8 (размер кода каждой буквы), а число контрольных битов 4
4. Для любых 5-6 пакетов в произвольном месте каждого из них заменить 0 на 1 или 1 на 0, но только 1 раз в 1 пакете
5. Выписать полученные пакеты на отдельный листочек и передать соседу, который не знает исходного сообщения.
6. Расшифровать полученные пакеты, исправляя ошибки, выписать полученное слово в соответствии со стандартом MS-DOS CP 866.
7. Сверить с соседом полученное слово.

# 1.4.3. Передача данных

## Способы передачи данных в компьютерах

### Параллельная передача данных

Шина данных



Характеристики:

1. Высокая скорость.
2. Существенное ограничение расстояния передачи.
3. Высокая стоимость линии.

### Последовательная передача данных

1. Синхронная
2. Асинхронная



# 1.4.3. Передача данных

## Связь компьютеров по телефонным линиям

