

Представление информации и цифровые устройства

Практическая работа на 60 минут. Изучите короткое объяснение в каждом разделе и сразу выполните задание. Записывайте ответы прямо на странице. В конце нажмите «Отправить работу — сохранить PDF».

ФИО

Лазарев Дмитрий Сергеевич

Группа

РУПО 26/11

Дата

2026-09-29

1

Формы представления чисел в ЭВМ

Числа в памяти компьютера представлены битами — последовательностями из нулей и единиц. Но одни и те же биты можно интерпретировать по-разному: как положительное целое число, число со знаком или вещественное число.

У **беззнакового целого числа** все биты используются для записи значения. Каждый двоичный разряд имеет свой вес: справа налево 1, 2, 4, 8 и так далее.

$$1011_2 = 1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 = 11_{10}$$

Знаковое целое число может быть отрицательным. Обычно используют дополнительный код: для получения отрицательного числа в заданной разрядности инвертируют биты положительного числа и прибавляют 1.

+3 в 8 битах: 00000011
Инверсия: 11111100
Прибавим 1: 11111101 → -3

Вещественные числа (например, 2,5) обычно хранятся в формате с плавающей точкой: знак, порядок и значащая часть. Некоторые десятичные дроби при этом представляются приближённо.

Главное: тип и разрядность определяют, как компьютер понимает набор битов и какой диапазон значений доступен.

Задание 1 • 8 минут

1. Переведите 13_{10} в двоичную систему. Покажите, из каких степеней двойки состоит число.
2. Запишите число -5 в 8-битном дополнительном коде. Покажите положительное число, инверсию и прибавление единицы.
3. Какой вид представления нужен для числа $3,14$: целое или с плавающей точкой?

Ваш ответ

- 1) $13 = 8 + 4 + 1 = 2^3 + 2^2 + 2^0 = 1101_2$
- 2) $+5 \rightarrow 00000101 \rightarrow$ инверсия $\rightarrow 11111010 \rightarrow +1 \rightarrow 11111011$
- 3) с плавающей точкой (вещественное)

2

Кодирование текстовой и графической информации

Как хранится текст

Компьютер хранит вместо буквы её **числовой код**. Стандарт Unicode задаёт коды символов разных языков. UTF-8 — распространённый способ записать эти коды в байтах. В UTF-8 символ может занимать от 1 до 4 байт.

Символ «А» → код Unicode U+0041
В UTF-8 → 01000001 (1 байт)

Как хранится изображение

Растровая картинка состоит из пикселей. У каждого пикселя есть цвет. В простой 24-битной RGB-записи цвет занимает 3 байта: красный, зелёный и синий каналы по 8 бит.

RGB(255, 0, 0) → красный цвет
Картинка $10 \times 10 = 100$ пикселей
 $100 \times 3 = 300$ байт без сжатия и служебных данных

Чем больше разрешение и глубина цвета, тем больше данных нужно для несжатого изображения. PNG и JPEG используют сжатие.

Задание 2 • 8 минут

1. Почему букву можно сохранить в памяти как число?
2. Вычислите размер несжатого изображения **20 × 10 пикселей** с глубиной цвета **24 бита**.
Ответ запишите в байтах; служебные данные не учитывайте.
3. Какой цвет обозначает RGB(0, 255, 0)?

Ваш ответ

- 1) компьютер хранит не саму букву, а её числовой код (стандарт unicode)
- 2) пикселей 200; на пиксель 3 байта; всего 600 байт ($20 \times 10 \times 3 = 600$)
- 3) зелёный цвет (максимальное значение во втором канале - green)

3

Кодирование звука и видео

Звук первоначально является непрерывным сигналом. При оцифровке компьютер регулярно измеряет его уровень — берёт отсчёты. **Частота дискретизации** показывает, сколько отсчётов делается за секунду, а **разрядность** — сколько бит отведено на один отсчёт.

Моно-звук: 8000 отсчётов/с, 8 бит на отсчёт
1 секунда: $8000 \times 8 = 64\,000$ бит = 8000 байт
(без сжатия и служебных данных)

Видео представляет собой последовательность изображений — кадров. Частота кадров показывает, сколько кадров сменяется за секунду. Обычно в видео также есть звук. Кодеки могут сжимать данные и сохранять не каждый кадр полностью.

$30 \text{ кадров/с} \times 2 \text{ секунды} = 60 \text{ кадров}$

Сжатие уменьшает размер файлов: например, MP3 применяют для звука, а H.264 — для видео. При сжатии с потерями часть исходной информации удаляется.

Задание 3 • 7 минут

1. Сколько байт займут **2 секунды** несжатого моно-звука с частотой дискретизации **16 000 Гц** и разрядностью **8 бит**?
2. Сколько кадров содержит **4-секундное** видео при **30 кадрах/с**?
3. Зачем звуковым и видеофайлам нужно сжатие?

Ваш ответ

- 1) 32 000 байт ($16\,000 \text{ отсчётов/с} \times 1 \text{ байт} \times 2 \text{ секунды} = 32\,000$)
- 2) 120 кадров ($30 \text{ кадров/с} \times 4 \text{ секунды} = 120$)
- 3) чтобы уменьшить размер файлов для экономии места и удобной передачи

4

Логические элементы. Базовые схемы

Логические элементы выполняют операции над сигналами 0 и 1. Они являются основой цифровых схем — от простого переключателя до процессора.

A	B	И (AND)	ИЛИ (OR)	Исключающее ИЛИ (XOR)
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

И выдаёт 1, только когда оба входа равны 1. **ИЛИ** выдаёт 1, если хотя бы один вход равен 1. **XOR** выдаёт 1, когда входы различаются. Элемент **НЕ (NOT)** меняет 0 на 1, а 1 — на 0.

$A = 1, B = 0$
 $A \text{ AND } B = 0$
 $A \text{ OR } B = 1$
 $A \text{ XOR } B = 1$
 $\text{NOT } A = 0$

Пример схемы: лампа должна загореться, только если включены два переключателя. Для такой логики подходит И (AND).

Задание 4 • 9 минут

1. Пусть **A = 1** и **B = 1**. Вычислите A AND B, A OR B, A XOR B и NOT A.
2. Какой элемент нужен, если сигнал должен появиться, когда сработал **хотя бы один** из двух датчиков?
3. Какой элемент нужен, если сигнал должен появиться, когда сработал **ровно один** из двух датчиков?

Ваш ответ

- 1) and = 1; or = 1; xor = 0; not a = 0 (по таблице истинности для a=1 и b=1)
- 2) или (or)
- 3) исключающее или (xor)

5

Триггеры. Регистры. Счётчики

Элементы из предыдущего раздела вычисляют значения. А чтобы компьютер мог **запоминать** информацию, нужны последовательностные схемы — устройства, состояние которых зависит и от прошлых событий.

Триггер — элемент памяти, способный хранить один бит: 0 или 1. **Регистр** — группа триггеров, хранящая двоичное слово, например 8 бит. **Счётчик** — схема, которая изменяет своё двоичное значение при поступлении импульсов.

Один триггер → 1 бит
8 триггеров → 8-битный регистр

Двухбитный счётчик:
00 → 01 → 10 → 11 → 00 → ...

Регистр нужен, чтобы временно хранить данные внутри процессора. Счётчик полезен, когда требуется посчитать тактовые импульсы или события.

Задание 5 • 7 минут

1. Сколько триггеров нужно, чтобы создать регистр на **16 бит**?
2. Двухбитный счётчик находился в состоянии **00**. Запишите его состояние после **трёх** импульсов, а затем после **четвёртого** импульса.
3. Какое из трёх устройств следует использовать для подсчёта нажатий кнопки?

Ваш ответ

- 1) 16 триггеров
- 2) после 3-го — 11; после 4-го — 00 (счётчик сбрасывается по кругу)
- 3) счётчик

6

Сумматоры. Шифраторы и дешифраторы

Сумматор складывает двоичные числа. Простейший полусумматор складывает два бита и выдаёт два результата: бит суммы и бит переноса в следующий разряд.

$1 + 1 = 10_2$
Бит суммы = 0
Бит переноса = 1

Шифратор преобразует номер активной входной линии в двоичный код. Например, шифратор «4 в 2» кодирует одну из четырёх линий двумя битами. В простом варианте предполагается, что активна только одна линия.

Дешифратор делает противоположное: по двоичному коду выбирает выходную линию. Например, дешифратор «2 в 4» получает два бита и активирует один из четырёх выходов.

Шифратор 4 → 2: вход № 3 → код 11_2
Дешифратор 2 → 4: код 10_2 → выход № 2

В примерах линии нумеруются с нуля: 0, 1, 2, 3. Шифратор и дешифратор выполняют разные направления преобразования.

Задание 6 • 8 минут

1. Полусумматор складывает **1 и 1**. Чему равны бит суммы и бит переноса?
2. У шифратора «4 в 2» активен вход **№ 2**. Какой двоичный код он выдаст?
3. На дешифратор «2 в 4» поступил код **01₂**. Какой выход станет активным?

Ваш ответ

- 1) сумма = 0; перенос = 1 (так как $1 + 1 = 10_2$)
- 2) двоичный код 10_2 (номер входа в двоичной системе)
- 3) выход № 1 (так как двоичный код 01_2 равен десятичной 1)

7

Мультиплексоры. Компараторы кодов

Мультиплексор (MUX) — цифровой переключатель. Он получает несколько входных сигналов, а управляющий код указывает, какой из них передать на один общий выход.

Мультиплексор «4 в 1»:

Код 00 → вход I0

Код 01 → вход I1

Код 10 → вход I2

Код 11 → вход I3

Компаратор кодов сравнивает два двоичных числа и выдаёт результат: $A = B$, $A > B$ или $A < B$. Это нужно, например, при проверке условий в цифровых устройствах.

$A = 0110_2 = 6_{10}$

$B = 0101_2 = 5_{10}$

Результат: $A > B$

Не путайте: мультиплексор **выбирает один сигнал**, а компаратор **сравнивает два числа**.

Задание 7 • 8 минут

1. На мультиплексор «4 в 1» подан управляющий код 10_2 . Какой вход выбран? Если на этом входе находится 1, что появится на выходе?
2. Компаратор сравнивает $A = 1010_2$ и $B = 1001_2$. Укажите, какое число больше, и переведите оба числа в десятичную систему.
3. Какое устройство нужно для выбора сигнала от одного из четырёх датчиков?

Ваш ответ

- 1) выбран вход I2, т.к управляющий двоичный код 10_2 соответствует десятичной двойке. Поскольку мультиплексор передаёт сигнал с выбранного входа без изменений, на выходе появится 1.
- 2) перевод в десятичную систему:
 $A = 1010_2 = 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 10_{10}$
 $B = 1001_2 = 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 9_{10}$
Сравнение: $10 > 9$, следовательно, число A больше числа B ($A > B$).
- 3) для выбора сигнала от одного из четырёх датчиков необходим мультиплексор «4 в 1». он позволяет коммутировать несколько линий связи в одну общую с помощью управляющего кода.

Перед сдачей • 5 минут вместе с оформлением

✓ Указаны ФИО и группа.

✓ Выполнены все семь заданий, ответы записаны по пунктам.

✓ В расчётах показаны промежуточные действия там, где они требуются.

После нажатия кнопки откроется окно печати. Выберите «Сохранить как PDF». В PDF попадут все ваши ответы полностью.