

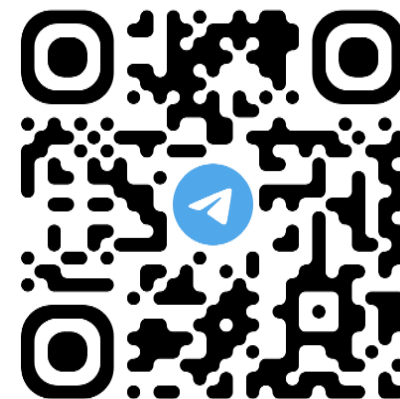


# Числа с плавающей точкой

Лекция 4

# Содержание

- 1 Примеры
- 2 Стандарт IEEE754
- 3 Нестандартизированные типы данных
- 4 Тренды



<https://t.me/+2kGsBQZcdBQxNDAY>

# Примеры

## Буря в пустыне, 1991

- Батарея ПВО США не сбила иракскую ракету, поразив собственную инфраструктуру (28 погибло, 100 раненых)
- Накопление погрешности при учете времени на 24-битном ЦПУ
  - Дискретизация – 0.1 сек. с момента запуска батареи
  - Накопленная погрешность в 0.34 сек. за 100 ч не позволила перехватить летящую со скоростью 1676 м/с цель

## Запуск ракеты Ariane 5 , 1996

- ПО с Ariane 4 переиспользовано на Ariane 5
- Переполнение из-за более высокой скорости по сравнению с предыдущим поколением
  - Конверсия `fp64` в `int16` с неполной защитой от переполнения на программном уровне
- Аналогичная ситуация - на резервном процессоре с последующей потерей устойчивости и уничтожением ракеты

# IEEE754

## История

- **1976:** John Palmer (Intel) встретился с Prof. William Kahan для обсуждения возможностей по разработке корпоративного стандарта для чисел с плавающей точкой
- **1978:** После участия во 2м собрании IEEE754 (09/1977) Prof. William Kahan вместе с командой разрабатывает первый черновик для следующего IEEE754 собрания (04/1978)
- **1980:** Микропроцессорные компании реализовали стандарт: Intel (8087 ко-процессор для 8086), Motorola (68881 ко-процессор для 68000), Zilog (Z8070 для Z8000), National Semiconductor (16081)
- **1985:** IEEE754 принят в качестве стандарта
- **1989:** IEEE754 принят в качестве международного стандарта ISO/IEC 60559
- **2008:** Следующая ревизия стандарта
- **2019:** Следующая ревизия стандарта

[https://ethw.org/Milestones:IEEE\\_Standard\\_754\\_for\\_Binary\\_Floating-Point\\_Arithmetic,\\_1985](https://ethw.org/Milestones:IEEE_Standard_754_for_Binary_Floating-Point_Arithmetic,_1985)

<https://people.eecs.berkeley.edu/~wkahan/ieee754status/754story.html>

# IEEE754

- **Стандарт IEEE754** (IEC 60559) описывает форматы представления чисел в виде бинарной и десятичной плавающей точки и операции над ними
  - Далее FP и DFP
- **Определяет вычисления** для аппаратуры, программного обеспечения или их комбинации
- **Для**
  - Разработки следующих стандарту архитектур для миграции приложений между ними
  - Разработки приложений с поддержкой портирования на другую архитектуру с получением аналогичных численных результатов
  - Разработки приложений, удовлетворяющих требованиям надежности, с возможностью детектирования и обработки числовых аномалий во время исполнения
  - Разработки математических функций, удовлетворяющих требованиям по точности

# IEEE754

## Стандарт специфицирует

- Форматы FP и DFP для вычислений и обмена данными
- Операции сложения, вычитания, умножения, деления, объединенного умножения и сложения (fma), квадратного корня, сравнения и др.
- Операции конверсии между целыми и FP форматами
- Операции конверсии между различными FP форматами
- Операции конверсии между FP и их строковым представлением
- FP исключения и их обработку, включая ситуацию «не чисел» (Not A Number, NaN)

## Стандарт не специфицирует

- Форматы целых чисел
- Интерпретацию полей знака и мантиссы в NaN

# IEEE754

Форматы, определяемые стандартом

Формат	Название	Основание
binary32	Одинарная точность, fp32	2
binary64	Двойная точность, fp64	2
binary128	Четверная точность fp128	2
decimal64		10
decimal128		10

Дополнительно определены форматы для обмена

- Binary16
- Binary{k},  $k \geq 128$ ,  $k$  – кратно 32, например 256

# IEEE754

## Формат FP

- $(-1)^s 2^e m$ ,  $s$  – знак,  $e$  – экспонента,  $m$  – мантисса
- $s = 0$  (положительное число) или  $1$  (отрицательно число)
- $e \in [e_{\min}, e_{\max}]$
- $m = d_0.d_1d_2 \dots d_{p-1}$ ,  $d_i \in \{0,1\}, m < 2$
- Представление кодирует +/- inf, quit & signaling NaN

## Формат DFP

- Бит для знака
- Поле  $G$  размера  $w+5$  битов, объединяющее экспоненту и 4 бита мантиссы
- Мантисса  $T$  размером  $J \times 10$  битов, совместно с битами из  $G$  кодирует  $3 \times J + 1$  десятичных знаков
- Представление кодирует +/-inf, quit & signaling NaN

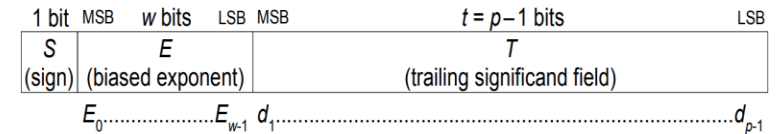


Figure 3.1—Binary interchange floating-point format

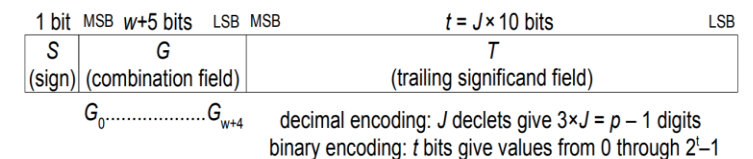


Figure 3.2—Decimal interchange floating-point formats

Далее сосредоточимся на  
FP

# IEEE754

## Параметры представления

Parameter	binary16	binary32	binary64	binary128
Размер, биты	16	32	64	128
Знак, биты	1	1	1	1
Экспонента, w, биты	5	8	11	15
Мантисса без «скрытого» бита, p-1, биты	10	23	52	112
$e_{max}$	15	127	1023	16383
$e_{min} = 1 - e_{max}$	-14	-126	-1022	-16382
Bias, $2^{w-1} - 1$ $E = e + \text{Bias}$	15	127	1023	16383

### Диапазоны:

- Binary16:  $\sim[-65000, 65000]$
- Binary32:  $\sim[-10^{38}, 10^{38}]$
- Binary64:  $\sim[-10^{308}, 10^{308}]$
- Binary64:  $\sim[-10^{4932}, 10^{4932}]$

# IEEE754

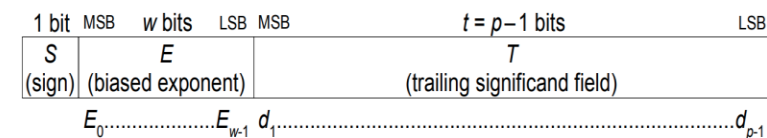


Figure 3.1—Binary interchange floating-point format

Значения смещенной мантиисы ( $E = e + \text{bias}$ ) и экспоненты	Представление f	Диапазон f
$E = 2^w - 1, T > 0$	qNaN: $d_1 = 1$ sNaN: $d_1 = 0, d_i = 1$ для некоторого i	f - qNaN или sNaN безотносительно знака S $f = 2^{e_{\max}+1} \times T$
$E = 2^w - 1, T = 0$	$f = (-1)^s (+\infty)$	f - бесконечность со знаком $f = 2^{e_{\max}+1}$
$E \in [1, 2^w - 2]$	$f = (-1)^s 2^{E-\text{bias}} (1 + 2^{1-p} * T)$ Неявный ведущий бит мантиисы - 1	f – нормальное/нормализованное число $f \in [2^{e_{\min}}, 2^{e_{\max}}(2 - 2^{1-p})]$
$E = 0, T > 0$	$f = (-1)^s 2^{e_{\min}} (0 + 2^{1-p} * T)$ Неявный ведущий бит мантиисы - 0	f – субнормальное число $f \in [2^{e_{\min}} \times 2^{1-p}, 2^{e_{\min}})$
$E = 0, T = 0$	$f = (-1)^s (+0)$	f – нуль со знаком знак может нести дополнительную информацию

# IEEE754

## Округление

- Операция округления
  - Вход – бесконечно представимое число
  - Выход - число, модифицированное согласно режиму округления
  - Сигнализирование об исключительных ситуациях (например, переполнение)

Режим	Результат
К ближайшему	<ul style="list-style-type: none"><li>• Ближайшее к входу</li><li>• roundTiesToEven: в случае двух кандидатов округление к четному</li><li>• roundTiesToAway: в случае двух кандидатов округление к большему по модулю</li><li>• <math>2^{e_{\max}}(2 - 0.5 \times 2^{1-p}) \rightarrow \text{inf}</math> без изменения знака</li></ul>
$+\infty$ (TowardPositive)	<ul style="list-style-type: none"><li>• Ближайшее к входу; не меньше, чем вход</li></ul>
$-\infty$ (TowardNegative)	<ul style="list-style-type: none"><li>• Ближайшее к входу; не больше, чем вход</li></ul>
0 (TowardZero)	<ul style="list-style-type: none"><li>• Ближайшее к входу; не больше по модулю, чем вход</li></ul>

# IEEE754

## Операции

- По типу результата и исключениям
  - General-computational:** целочисленный или FP результат для заданного режима округления; возможны FP исключения
  - Quiet-computational:** FP результат без сигнализации FP исключений
  - Signaling-computational:** нет FP результата с сигнализацией FP исключений
  - Non-computational:** нет FP результата, нет FP исключений
- По формату входов и выходов:
  - Homogeneous:** FP входы и FP выходы имеют один и тот же формат
  - formatOf:** формат FP выхода отличается от формата FP входов

Тип операции	Примеры
Homogeneous general-computational	<ul style="list-style-type: none"><li>roundToIntegralTiesToEven</li><li>roundToIntegralTiesToAway</li><li>nextUp</li></ul>
formatOf General-computational	<ul style="list-style-type: none"><li>formatOf-addition</li><li>formatOf-squareRoot</li><li>formatOf-convertFormat</li></ul>
Quit-computational	<ul style="list-style-type: none"><li>Copy</li><li>Negate</li><li>abs</li></ul>
Signaling-computational	<ul style="list-style-type: none"><li>compareQuietEqual</li><li>compareSignalingEqual</li></ul>
Non-computational	<ul style="list-style-type: none"><li>Is754version1985</li><li>Is754version2008</li><li>isSignMinus</li><li>totalOrder</li></ul>

# IEEE754

## Исключения

- Операция над операндами не может привести к результату, который устроил бы приложения
- Исключение обрабатывается с помощью встроенных или альтернативных механизмов

Исключение (Exception)	Комментарии
Invalid	<ul style="list-style-type: none"><li>• Результат не определен для заданных операндов</li><li>• <math>0 \times \infty</math>, <math>0 / 0</math>, <math>\sqrt{x}</math> для <math>x &lt; 0</math>, fp-&gt;int конверсия</li></ul>
Division by zero	<ul style="list-style-type: none"><li>• Точный бесконечный результат для операции над конечными операндами</li><li>• <math>x/y</math>, <math>x</math>-конечное, <math>y=0</math></li></ul>
Overflow	<ul style="list-style-type: none"><li>• Результат операции превосходит не может быть представлен в заданном формате с учетом режима округления</li></ul>
Underflow	<ul style="list-style-type: none"><li>• Результат операции – малый (в диапазоне <math>\pm 2^{emin}</math>) ненулевой до или после округления</li></ul>
Inexact	<ul style="list-style-type: none"><li>• Результат операции отличается от того, который мог бы быть получен в случае бесконечных мантиссы и экспоненты</li></ul>

# IEEE754

## Сложение на примере $0.5 - 0.4375 = 0.0625$ в **binary32**

- $a = 0.5 = 2^{-1} \times 1.000$ ,  $b = -0.4375 = 2^{-2} \times 1.110$
- Приведем экспоненту  $b$  к экспоненте  $a$  ( $e_a > e_b$ )
  - $b = 2^{-1} \times 0.111$
- Сложим мантиссы
  - $2^{-1} \times 1.000 - 2^{-1} \times 0.111 = 2^{-1} \times 0.001$
- Нормализуем результат, проверим на overflow/underflow
  - $2^{-1} \times 0.001 = 2^{-4} \times 1.000$
  - $-126 \leq -4 \leq 127$  – исключения не возникают
- Округляем результат (не требуется, исключение не возникает)
- $2^{-4} \times 1.000 = 0.0625$

## Умножение на примере $0.5 \times -0.4375 = -0.21875$ в **binary32**

- $a = 0.5 = 2^{-1} \times 1.000$ ,  $b = -0.4375 = 2^{-2} \times 1.110$
- Сложим смещенные экспоненты
  - $(-1 + 127) + (-2 + 127) - 127 = -3 + 127$
- Перемножим мантиссы
  - $1.000 \times 1.110 = 1.110$
  - Нормализуем результат, проверим на overflow/underflow
    - $2^{-3} \times 1.110$
    - $1 \leq (-3 + 127) \leq 254$  – исключения не возникают
- Округляем результат (не требуется, исключение не возникает)
- Формируем бит результата, -1
  - $-2^{-3} \times 1.110 = -0.21875$

# IEEE754

## Еще примеры

- $0.1 = 0x3DCCCCCD = 0.1000000014901161\dots$  в binary32
- **Представимы ли числа  $2^{23}$ ,  $2^{23} - 1$ ,  $2^{24} + 1$  в binary32?**
- **Если эти числа не представимы, то каков будет результат при округлении к ближайшему**
  - roundTiesToEven
  - roundTiesToAway
- $a = -2^{15} \times 1.0, b = 2^{15} \times 1.0, c = 2^{-10} \times 1.0$ , binary32
  - Вычислить  $(a+b) + c$ ,  $a + (b + c)$

# IEEE754

## Катастрофическая потеря точности

- Результат вычитания двух (положительных) приблизительно равных чисел много менее аккуратен, чем операнды
  - $\tilde{x} = x(1 + \sigma_x)$ ,  $\tilde{y} = y(1 + \sigma_y)$ , где  $\sigma_x$  и  $\sigma_y$  - относительные ошибки представления  $x$  и  $y$ 
    - Ошибка связана с результатами предыдущих вычислений, измерениями, дискретизацией, моделированием
  - $\tilde{x} - \tilde{y} = (x - y) \left(1 + \frac{x\sigma_x - y\sigma_y}{x - y}\right)$ ,
    - если  $x$  и  $y$  приблизительно равны, то ошибка результата вычитания нарастает
- Методы вычисления смещенной оценки дисперсии  $DX = E(X - EX)^2$ ,  $DX \geq 0$ , на базе выборки  $(x_1, \dots, x_n)$ 
  - “Быстрый”:  $DX = RS - M^2$ , где  $RS = \frac{1}{n} \sum_{i=1}^n x_i^2$ ,  $M = \frac{1}{n} \sum_{i=1}^n x_i$
  - Двухпроходной:  $M = \frac{1}{n} \sum_{i=1}^n x_i$ ,  $DX = \frac{1}{n} \sum_{i=1}^n (x_i - M)^2$
  - Однопроходной:  $M_{n+1} = M_n + \frac{x_{n+1} - M_n}{n}$ ,  $DX_{n+1} = DX_n + \frac{(x_{n+1} - M_n)(x_{n+1} - M_{n+1}) - DX_n}{n}$
  - Какие методы подвержены проблеме катастрофической потери точности?**

# Нестандартизированные типы данных

Тип	Знак, биты	Экспонента, биты	Мантисса, биты	Комментарии
bfloat16	1	8	7	<ul style="list-style-type: none"><li>• К ближайшему (четному), зависит от реализации</li><li>• Inf/[s q]NaN</li></ul>
tfloat32	1	8	10	<ul style="list-style-type: none"><li>• К ближайшему (четному)</li><li>• Inf/ “canonical” NaN</li></ul>
<a href="#">fp8 e4m3</a>	1	4	3	<ul style="list-style-type: none"><li>• Округление определяется реализацией</li><li>• Inf не определено</li><li>• NaN доступен в виде одного паттерна</li></ul>
<a href="#">fp8 e5m2</a>	1	5	2	<ul style="list-style-type: none"><li>• Округление определяется реализацией</li><li>• Inf/NaN</li></ul>

Нестандартизированные типы данных:

- Модифицируют требования IEEE754
- Применяются в задачах расчетов на базе нейронных сетей для ускорения вычислений, снижения нагрузки на объем данных, балансируя точность тренировки
- Реализованы в GPU и CPU

# Тренды

- **MSFP (Microsoft Floating Point)**

- Набор FP типов данных для специализированной аппаратуры, MSFT16, MSFT12
- Формат аналогичен IEEE754 с общей для набора чисел экспонентой

- **POSIT**

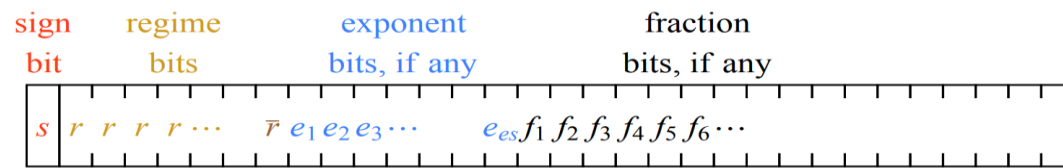
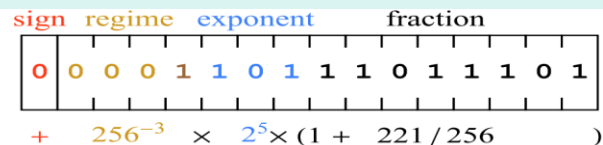


Table 1. Run-length meaning *k* of the regime bits

Binary	0000	0001	001x	01xx	10xx	110x	1110	1111
Numerical meaning, <i>k</i>	-4	-3	-2	-1	0	1	2	3



# Домашнее задание

- Исследовать проблему катастрофической потери точности, реализовав три метода для вычисления оценки дисперсии в `float[32|64]` с использованием как минимум 3 выборок:
  - 1000 чисел из нормального распределения (среднее = 1, среднеквадратическое отклонение = 1)
  - 1000 чисел из нормального распределения (среднее = 10, среднеквадратическое отклонение = 0.1)
  - 1000 чисел из нормального распределения (среднее = 100, среднеквадратическое отклонение = 0.01)
  - ...