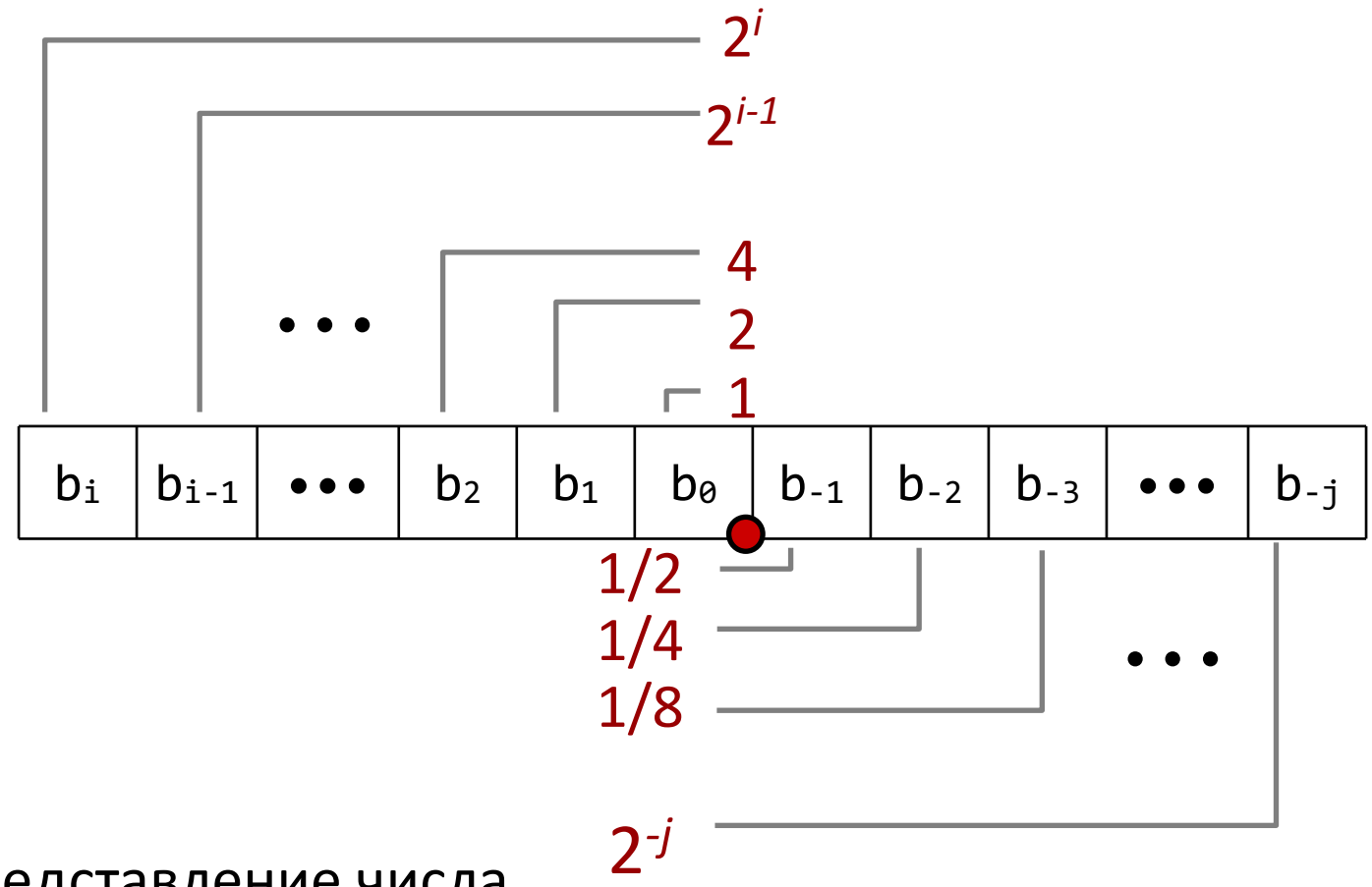
Лекция 14

30 марта

Дробные двоичные числа



- Представление числа
  - Биты справа от “двоичной точки” представляют отрицательные степени 2
  - Точное представление для рациональных чисел вида : $\sum_{k=-j}^{i} b_k \times 2^k$

Примеры дробных двоичных чисел

Число	**Представление**
5 3/4	101.11_2
2 7/8	10.111_2
63/64	0.111111_2

- Деление на 2 может выполняться сдвигом вправо, …
- … а умножение на 2 – сдвигом влево
- Числа вида $0.11111\ldots_2$
  - На один «шаг» меньше чем 1.0
  - Используется специальное обозначение $1.0 - \varepsilon$

Представимые рациональные числа

- Ограничение
  - Можно представить рациональные числа только вида $x/2^k$
  - Другие рациональные числа представляются повторяющимися группами бит

- Число — Представление

1/3	$0.0101010101[01]\ldots_2$
1/5	$0.00110011001[1001]\ldots_2$
1/10	$0.000110011001[1001]\ldots_2$

Представление чисел с плавающей точкой

- Численное представление

$$(-1)^s \times M \times 2^E$$

  - Знаковый бит s определяет, является число положительным или отрицательным
  - Мантисса M – дробное число в полуинтервале [1.0,2.0).
  - Экспонента E определяет степень 2 в третьем множителе

- Кодировка
  - Наибольший значащий бит s – знаковый бит s
  - Поле exp кодирует экспоненту E
  - Поле frac кодирует мантиссу M

Размеры чисел

- Одинарная точность: 32 бита. Тип – `float`.
  - Знак s — 1 бит
  - Мантисса M — 23 бита
  - Экспонента E — 8 битов
- Двойная точность: 64 бита. Тип – `double`.
  - Знак s — 1 бит
  - Мантисса M — 52 бита
  - Экспонента E — 11 битов
- Нормализация чисел
  - Нормализованное значение – мантисса не принимает «крайние» значения (одни нули или одни единицы)
  - Денормализованное значение – мантисса либо ноль, либо 11…11

Нормализованное число

- Значение: `float f = 15213.0;`

  $15213_{10} = 11101101101101_2$

  $= 1.1101101101101_2 \times 2^{13}$

- Мантисса

  M = 1.1101101101101_2

  frac = $11011011011010000000000_2$

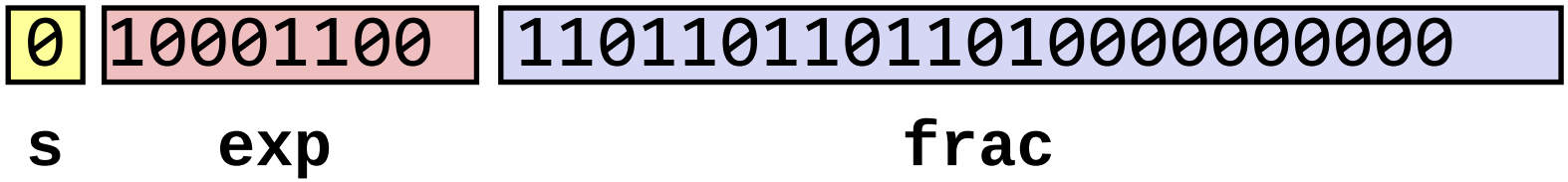
- Экспонента

  E = 13

  Смещение = 127

  Exp = E + Смещение = 140 = 10001100_2

- Итого:



Денормализованное число

- Условие: exp = 000…0
- Значение экспоненты: `E = –Смещение + 1` (вместо `E = 0 – Смещение`)
- Мантисса кодируется с ведущим `0`: `M = 0.`xxx…x_2
  - xxx…x: биты поля `frac`
- Примеры
  - exp = 000…0, frac = 000…0
    - Представляет число ноль
    - Различные кодировки для +0 и –0
  - exp = 000…0, frac ≠ 000…0
    - Кодируются числа близкие к 0.0
    - Распределены по числовой прямой с равным шагом

Особые числа

- Условие: exp = 111…1
- Пример: exp = 111…1, frac = 000…0
  - Представляет бесконечно большое число ∞
    (как положительное, так и отрицательное)
  - Требуются для операций в которых может произойти переполнение

    `1.0/0.0 = –1.0/–0.0 = +∞`

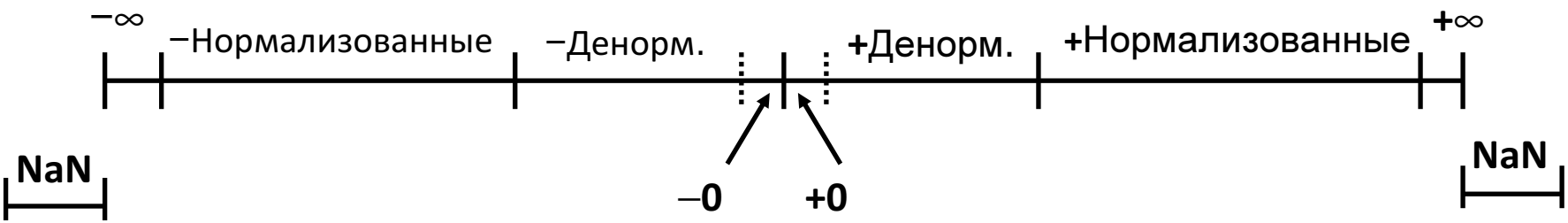
    `1.0/–0.0 = –∞`
- Пример: exp = 111…1, frac ≠ 000…0
  - Not-a-Number (NaN)
  - Используется в ситуациях, когда значение операции не определено

    `sqrt(–1)`

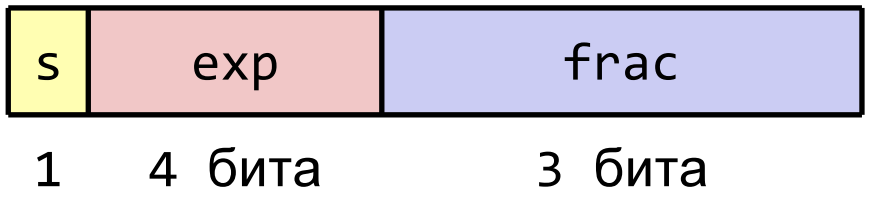
    `∞ – ∞`

    `∞ × 0`

Диапазоны значений



Пример



- 8-разрядные числа с плавающей точкой
  - Знаковый бит – старший бит
  - Следующие четыре бита – экспонента, смещение – 7
  - Последние три бита – дробная часть (мантисса)

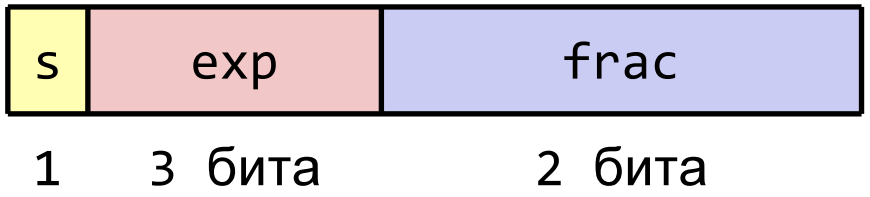
- Выполнены все требования стандарта IEEE 754 к формату числа
  - Реализованы нормализованные и денормализованные числа
  - Представлены значения 0, NaN, бесконечность

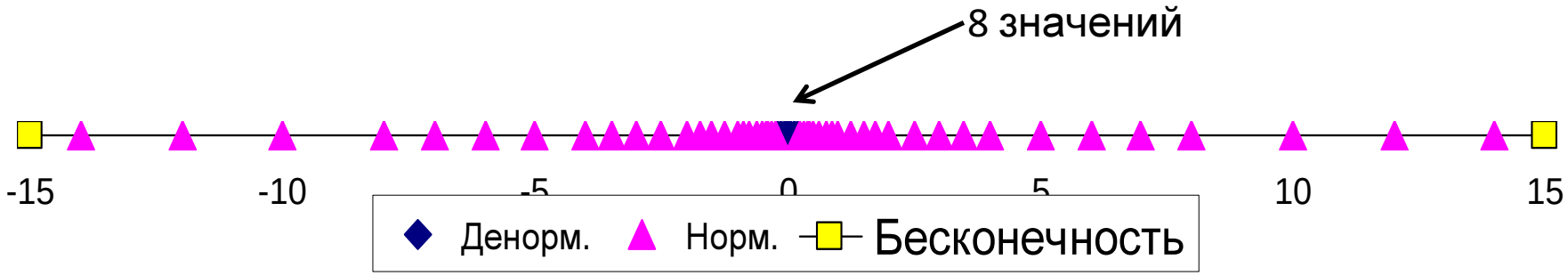
Диапазоны значений (только для положительных чисел)

	s	exp	frac	E	Значения	
Денормализованные числа	0	0000	000	-6	0	
	0	0000	001	-6	1/8*1/64 = 1/512	Ближайшее к 0
	0	0000	010	-6	2/8*1/64 = 2/512	
	…					
	0	0000	110	-6	6/8*1/64 = 6/512	
	0	0000	111	-6	7/8*1/64 = 7/512	Наибольшее денорм.
Нормализованные числа	0	0001	000	-6	8/8*1/64 = 8/512	Наименьшее норм.
	0	0001	001	-6	9/8*1/64 = 9/512	
	…					
	0	0110	110	-1	14/8*1/2 = 14/16	
	0	0110	111	-1	15/8*1/2 = 15/16	Ближайшее к 1 «снизу»
	0	0111	000	0	8/8*1 = 1	
	0	0111	001	0	9/8*1 = 9/8	Ближайшее к 1 «сверху»
	0	0111	010	0	10/8*1 = 10/8	
	…					
	0	1110	110	7	14/8*128 = 224	
	0	1110	111	7	15/8*128 = 240	Наибольшее норм.
	0	1111	000	n/a	inf	

Распределение значений по числовой прямой

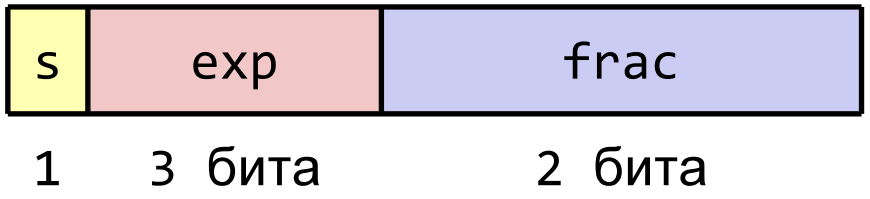
- 6-разрядный формат
  - e = 3 бита экспоненты
  - f = 2 бита мантиссы
  - Смещение $2^{3-1}-1 = 3$
- Распределение сильно «сгущается» в окрестности 0

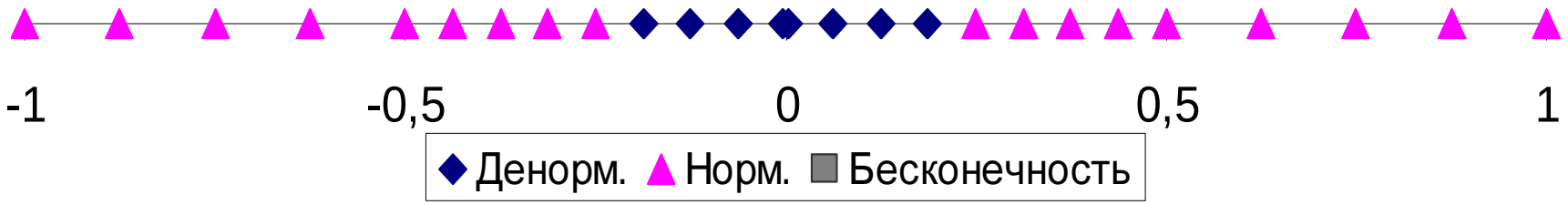




Распределение значений по числовой прямой (вид вблизи)

- 6-разрядный формат
  - e = 3 бита экспоненты
  - f = 2 бита мантиссы
  - Смещение 3





Некоторые числа

Описание	*exp*	*frac*	*Численное значение*
• Ноль	00…00	00…00	0.0
• Наименьшее «+» денорм.	00…00	00…01	$2^{-\{23,52\}} \times 2^{-\{126,1022\}}$
– Одинарная точность ≈ 1.4×10^{-45}			
– Двойная точность ≈ 4.9×10^{-324}			
• Наибольшее денорм.	00…00	11…11	$(1.0 - \varepsilon) \times 2^{-\{126,1022\}}$
– Одинарная точность ≈ 1.18×10^{-38}			
– Двойная точность ≈ 2.2×10^{-308}			
• Наименьшее «+» норм.	00…01	00…00	$1.0 \times 2^{-\{126,1022\}}$
– Немногим больше чем наибольшее денормализованное			
• Единица	01…11	00…00	1.0
• Наибольшее норм.	11…10	11…11	$(2.0 - \varepsilon) \times 2^{\{127,1023\}}$
– Одинарная точность ≈ 3.4×10^{38}			
– Двойная точность ≈ 1.8×10^{308}			

Точность {одинарная,двойная}

Особенности кодировки

- FP ноль совпадает с целочисленным нулем
  - Все биты = 0
- Допустимо (в большинстве случаев) использовать беззнаковое целочисленное сравнение
  - Сперва сравниваем знаковые биты
  - Необходимо рассматривать −0 = 0
  - NaNs
    - В целочисленной интерпретации больше, чем любые другие числа
    - Что необходимо выдавать в качестве результата сравнения?
  - В противном случае …
    - Денормализованные vs. Нормализованные
    - Нормализованные vs. Бесконечность

Операции над числами с плавающей точкой

- $x +_f y = Round(x + y)$
- $x \times_f y = Round(x \times y)$
- Основная идея
  - Сперва вычислить точный результат
  - Поместить результат в требуемый размер точности
    - Переполнение, если экспонента слишком большая
    - Возможно придется округлять поле frac

Округление

- Способы округления

	1.40	1.60	1.50	2.50	–1.50
– К нулю	1	1	1	2	–1
– К наименьшему (−∞)	1	1	1	2	–2
– К наибольшему (+∞)	2	2	2	3	–1
– К ближайшему (✓)	1	2	2	2	–2

Округление к ближайшему целому числу

- Основной способ округления
  - Все остальные способы дают статистическое смещение
    - Пример: суммирование положительных чисел будет давать устойчивую недо- или пере- оценку результата
- Применимо при округлении в произвольной позиции дроби
  - Когда число расположено точно посредине двух значений к которым можно округлить
    - Округляют к тому числу, у которого наименьшая значащая цифра четная
  - Например, округление до ближайших сотых

1.2349999	1.23	
1.2350001	1.24	
1.2350000	1.24	(середина — округляем к большему)
1.2450000	1.24	(середина — округляем к меньшему)

Округление двоичных чисел

- Двоичные дробные числа
  - “Четные” числа у которых младший значащий бит 0
  - “Середина” – когда биты справа от позиции к которой происходит округление = $100\ldots_2$
- Примеры
  - Округление до ближайшей 1/4 (2 бита справа от бинарной точки)

Число	Двоичное	Окр.	Действие	Окр. число
2 3/32	10.00011_2	10.00_2	(<1/2—down)	2
2 3/16	10.00110_2	10.01_2	(>1/2—up)	2 1/4
2 7/8	10.11100_2	11.00_2	(1/2—up)	3
2 5/8	10.10100_2	10.10_2	(1/2—down)	2 1/2

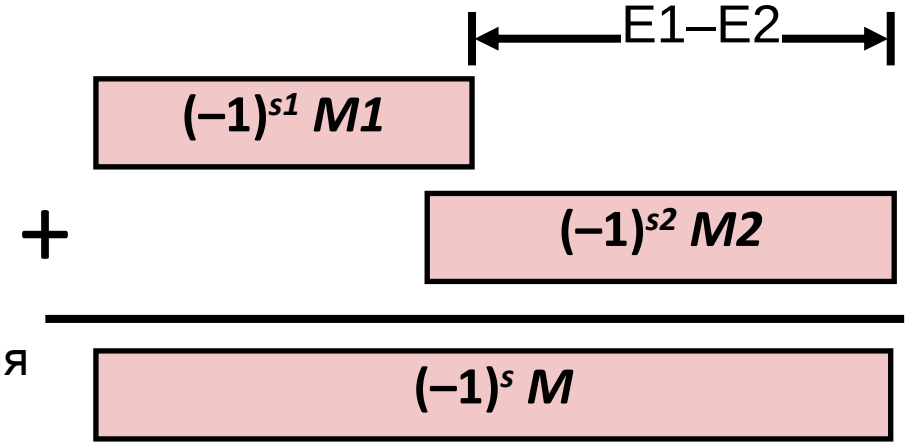
Умножение

- $(-1)^{s1}\ M1\ \ 2^{E1}\ \ x\ \ (-1)^{s2}\ M2\ \ 2^{E2}$
- Точный результат: $(-1)^{s}\ M\ \ 2^{E}$
  - Знаковый бит s: s1 ^ s2
  - Мантисса M: M1 x M2
  - Экспонента E: E1 + E2

- Исправление
  - Если $M \geq 2$, сдвигаем M вправо (делим на 2), увеличивая E
  - Если E выходит за пределы, переполнение
  - Округляем M до соответствующего размера поля frac

Сложение

- $(-1)^{s1}\ M1\ \ 2^{E1}\ \ +\ \ (-1)^{s2}\ M2\ \ 2^{E2}$
  - Пусть E1 > E2
- Точный результат: $(-1)^{s}\ M\ \ 2^{E}$
  - Знаковый бит s, мантисса M:
    - Результат выравнивания и сложения
  - Экспонента E: E1



- Исправление
  - Если M ≥ 2, сдвигаем M вправо, увеличивая E
  - Если M < 1, сдвигаем M влево на k позиций, уменьшая E на k
  - Переполнение если E выходит за пределы
  - Округляем *M* до соответствующего размера поля frac

Математические свойства сложения

- Выполняются ли свойства Абелевых групп
  - Замкнутость?
    - Результатом может быть бесконечность или NaN
  - Коммутативность?
  - Ассоциативность?
    - Переполнения и изменение результата при округлении
  - `0.0` – нейтральный элемент?
  - Каждый элемент имеет обратный
    - За исключением бесконечности и NaN
- Монотонность
  - $a \geq b \Rightarrow a+c \geq b+c$?
    - За исключением бесконечности и NaN

Математические свойства умножения

- Выполняются ли свойства коммутативных колец
  - Замкнуто ли относительно умножения?
    - Результат может быть бесконечность или NaN
  - Умножение коммутативно?
  - Умножение ассоциативно?
    - Возможность переполнения, неточности округления
  - 1.0 – мультипликативная единица?
  - Умножение дистрибутивно над сложением?
    - Возможность переполнения, неточности округления

- Монотонность
  - $a \geq b \ \& \ c \geq 0 \ \Rightarrow a * c \geq b * c$?
    - Исключение – бесконечность и NaN

Числа с плавающей точкой в языке Си

- Язык Си вводит два уровня точности
  - –float одинарная точность
  - –double двойная точность
- Приведение типа
  - –Приведение типа между int, float, и double включает изменение битового представления
  - – double/float → int
    - Отбрасывается дробная часть (аналогично округлению к нулю)
    - Поведение не определено, когда значение вне допустимого диапазона или NaN: как правило устанавливается TMin
  - – int → double
    - Точное приведение, поскольку long и int 32 бита ≤ 53 бита
  - – int → float
    - Будет округляться согласно принятым соглашениям

Задачи

- Для каждого Си-выражения объяснить:
  - почему оно верно для любого значения переменных, …
  - … либо почему ложно

```
int x = …;
float f = …;
double d = …;
```

Предполагается, что
d и f не являются NaN

- `x == (int)(double) x`
- `x == (int)(float) x`
- `f == (float)(double) f`
- `d == (float) d`
- `f == -(-f);`
- `2/3 == 2/3.0`
- `d < 0.0` ⇒ `((d*2) < 0.0)`
- `d > f` ⇒ `-f > -d`
- `d * d >= 0.0`
- `(d+f)-d == f`

Промежуточные итоги

- IEEE754 – четкое определение математических свойств
- Представляются числа вида $M \times 2^E$
- Семантика операций не зависит от особенностей аппаратуры
  - Сперва точное вычисление, затем округление
- Отличия от «настоящей» арифметики
  - Нарушаются свойства ассоциативности и дистрибутивности
  - Создаются сложности для компилятора и серьезных математических вычислений

Сопроцессор x87

- 8 регистров для данных
  - Организованны в виде стека + кольцо
  - Указатель верхушки стека TOP
- Регистр состояния
- Управляющий (контрольный) регистр

Размер чисел с плавающей точкой


```
dd 1.234567e20 ; Константы одинарной точности
dq 1.234567e20 ; Двойной точности
dt 1.234567e20 ; Расширенной точности
```

NASM и числа с плавающей точкой

```
db -0.2                          ; «Четверть»
dw -0.5                          ; IEEE 754r/SSE5
                                 ; половинная точность
dd 1.2                           ; одинарная точность
dd 1.222_222_222                 ; допускается использовать
                                 ; знак подчеркивания
dd 0x1p+2                        ; 1.0x2^2 = 4.0
dq 0x1p+32                       ; 1.0x2^32 = 4 294 967 296.0
dq 1.e10                         ; 10 000 000 000.0
dq 1.e+10                        ; синоним для 1.e10
dq 1.e-10                        ; 0.000 000 000 1
dt 3.14159265358979238462 ; число Пи
do 1.e+4000                      ; IEEE 754r четверная точность
```

- IEEE 754r – опубликован в 2008 году

NASM и числа с плавающей точкой

- __float8__
- __float16__
- __float32__
- __float64__
- __float80m__
- __float80e__
- __float128l__
- __float128h__

- __Infinity__
- __NaN__

```
dq +1.5, -__Infinity__, __NaN__
mov eax, __float32__(3.1415926)
```