*МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ*

*НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ*

*„КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”*

*НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС*

*„ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ”*

**Лабораторна робота №3**

з курсу «Архітектура

обчислювальних систем**»**

Тема: «Дослідження роботи

арифметичного сопроцесору.»

*Виконав:*

*студент II курсу*

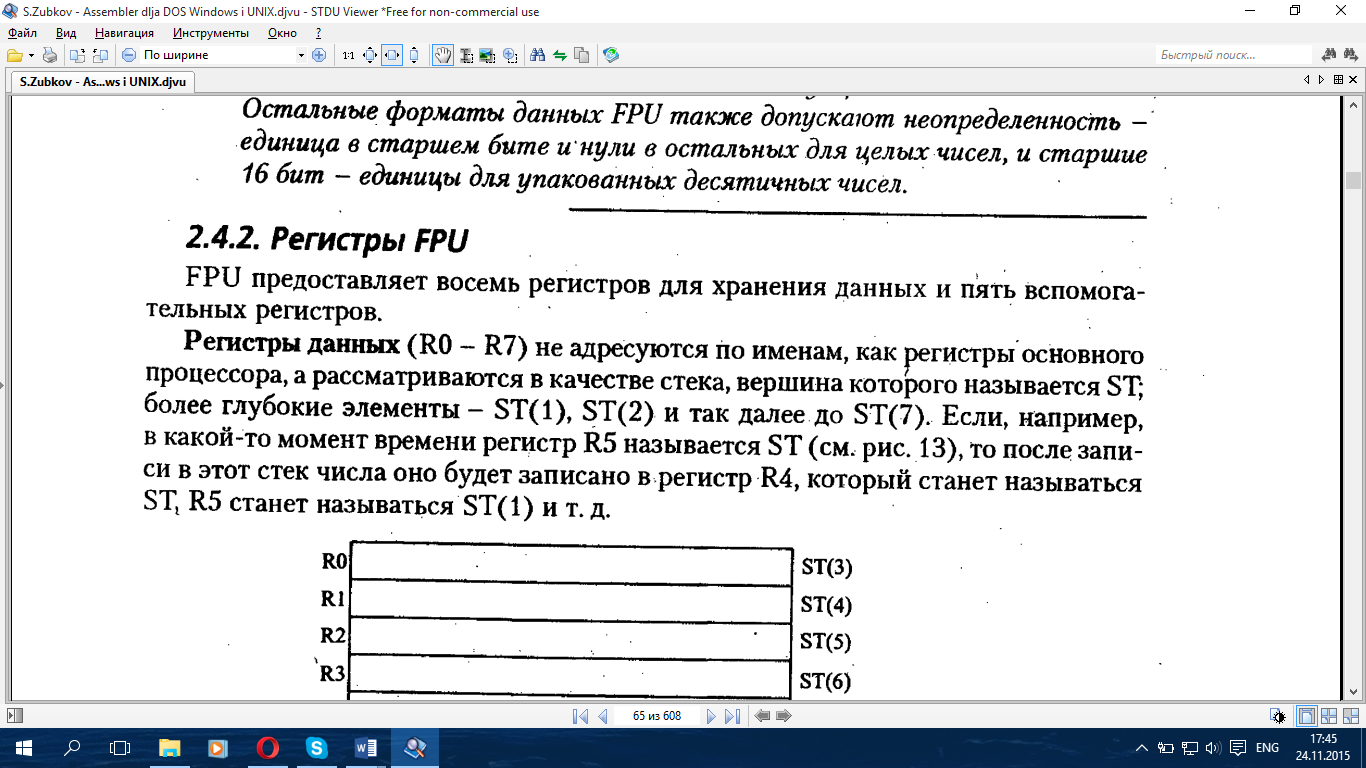
*групи ДА-51*

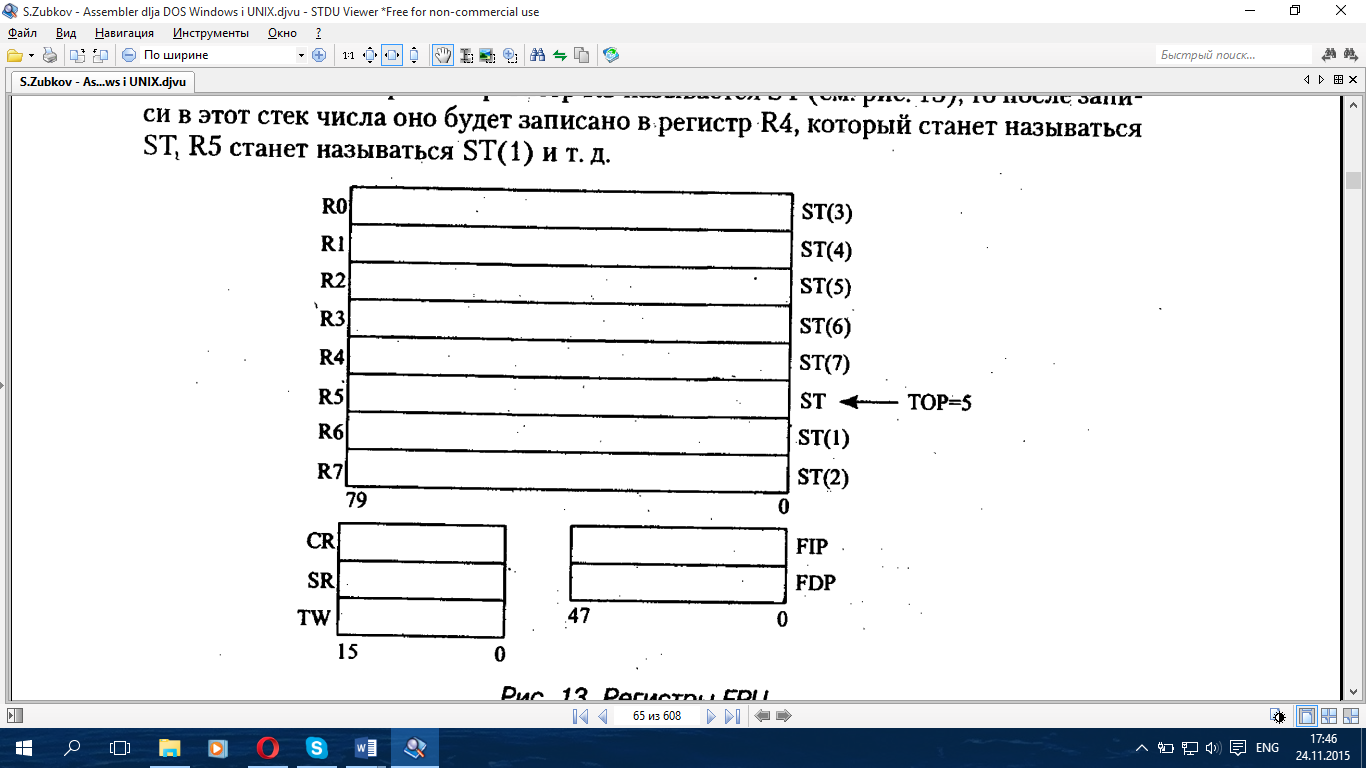
*Болобан Олег*

Київ

2017

***Теоретичні відомості***





*Представление чисел с плавающей точкой*

*Плавающая точка (floating point)* - метод представления действительных чисел, при котором число хранится в виде мантиссы и показателя степени, а значение числа вычисляется по формуле:

x = (-1)^{sign} \times mant \times base^{exp},

где x - число, sign - бит, отвечающий за знак числа, mant - мантисса, base - основание степени, exp - показатель степени.

Такой метод является компромиссом между точностью и диапазоном представляемых значений.

Используются следующие форматы представления чисел:

* *половинной точности (half precision)* (16 бит),
* *одинарной точности (single precision)* (32 бита),
* *двойной точности*(*double precision*) (64 бит)
* *четверной точности (quadruple precision)* (128 бит),
* *расширенной точности (extended precision)* (80 бит).

## *Нормальная и нормализованная формы*

*Нормальной* называется форма представления числа, при которой абсолютное значение мантиссы десятичного числа находится на полуинтервале [0,1).

Недостатком такой записи является тот факт, что числа нельзя записать однозначно: 0.01 = 0.001 \times 10^1.

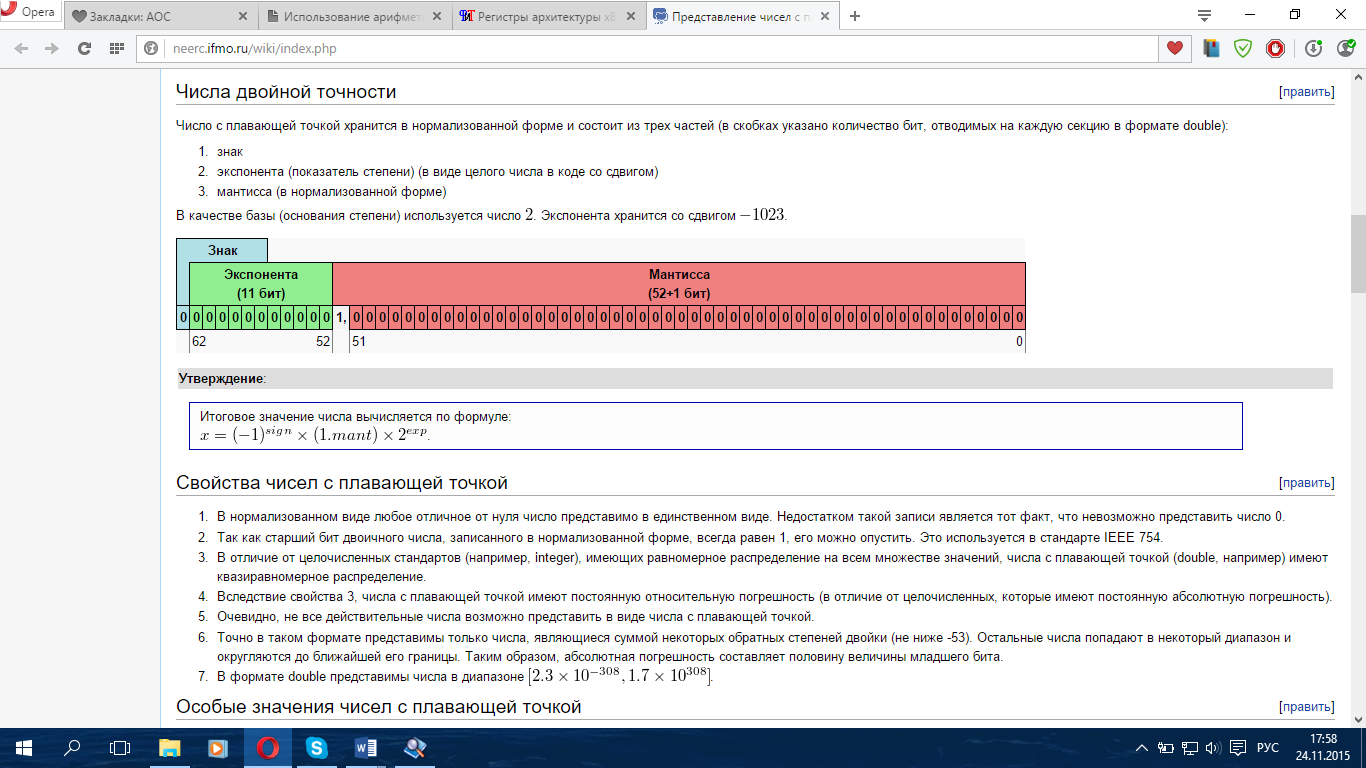
*Нормализованной* называется форма представления числа, при которой абсолютное значение мантиссы десятичного числа лежит на полуинтервале [1, 10), а двоичного на полуинтервале [1, 2).

## *Числа двойной точности*

Число с плавающей точкой хранится в нормализованной форме и состоит из трех частей (в скобках указано количество бит, отводимых на каждую секцию в формате double):

1. знак
2. экспонента (показатель степени) (в виде целого числа в коде со сдвигом)
3. мантисса (в нормализованной форме)

В качестве базы (основания степени) используется число 2. Экспонента хранится со сдвигом -1023.



Итоговое значение числа вычисляется по формуле:  x = (-1)^{sign} \times (1.mant) \times 2^{exp}.

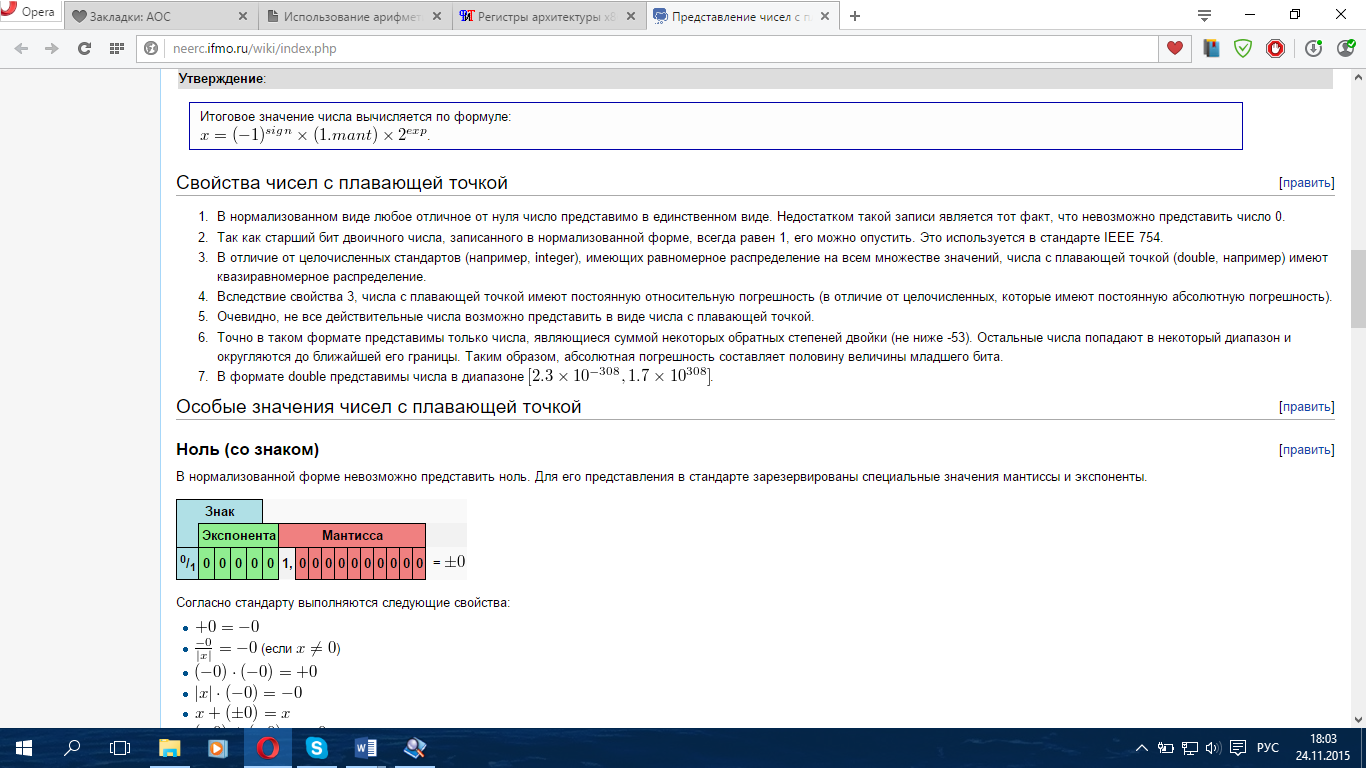
## *Свойства чисел с плавающей точкой*

1. В нормализованном виде любое отличное от нуля число представимо в единственном виде. Недостатком такой записи является тот факт, что невозможно представить число 0.
2. Так как старший бит двоичного числа, записанного в нормализованной форме, всегда равен 1, его можно опустить. Это используется в стандарте IEEE 754.
3. В отличие от целочисленных стандартов (например, integer), имеющих равномерное распределение на всем множестве значений, числа с плавающей точкой (double, например) имеют квазиравномерное распределение.
4. Вследствие свойства 3, числа с плавающей точкой имеют постоянную относительную погрешность (в отличие от целочисленных, которые имеют постоянную абсолютную погрешность).
5. Очевидно, не все действительные числа возможно представить в виде числа с плавающей точкой.
6. Точно в таком формате представимы только числа, являющиеся суммой некоторых обратных степеней двойки (не ниже -53). Остальные числа попадают в некоторый диапазон и округляются до ближайшей его границы. Таким образом, абсолютная погрешность составляет половину величины младшего бита.
7. В формате double представимы числа в диапазоне [2.3 \times 10^{-308}, 1.7 \times 10^{308}].

## *Особые значения чисел с плавающей точкой*

### *Ноль (со знаком)*

В нормализованной форме невозможно представить ноль. Для его представления в стандарте зарезервированы специальные значения мантиссы и экспоненты.

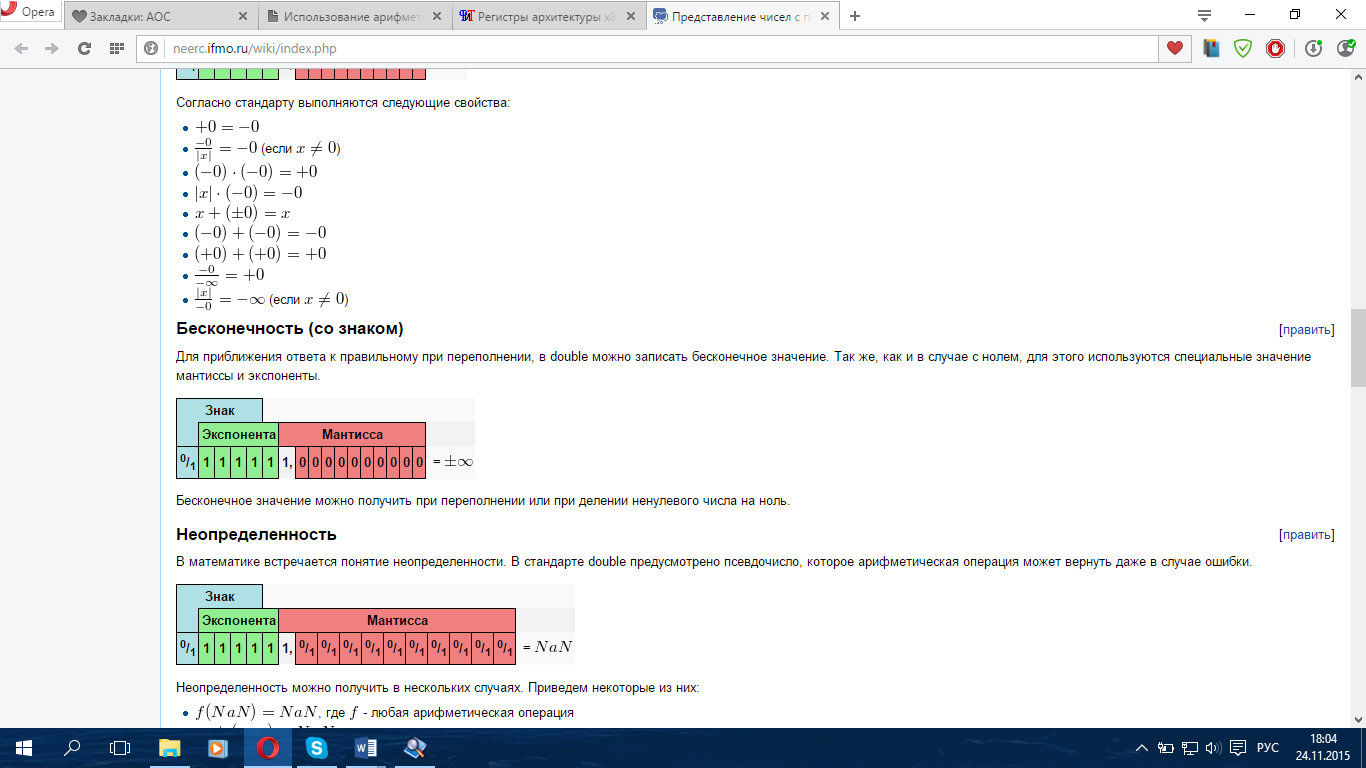


Согласно стандарту выполняются следующие свойства:

* +0 = -0
* \frac{-0}{ \left| x \right| } = -0\,\! (если x\ne0)
* (-0) \cdot (-0) = +0\,\!
* \left| x \right| \cdot (-0) = -0\,\!
* x + (\pm 0) = x\,\!
* (-0) + (-0) = -0\,\!
* (+0) + (+0) = +0\,\!
* \frac{-0}{-\infty} = +0\,\!
* \frac{\left|x\right|}{-0} = -\infty\,\! (если x\ne0)

### *Бесконечность (со знаком)*

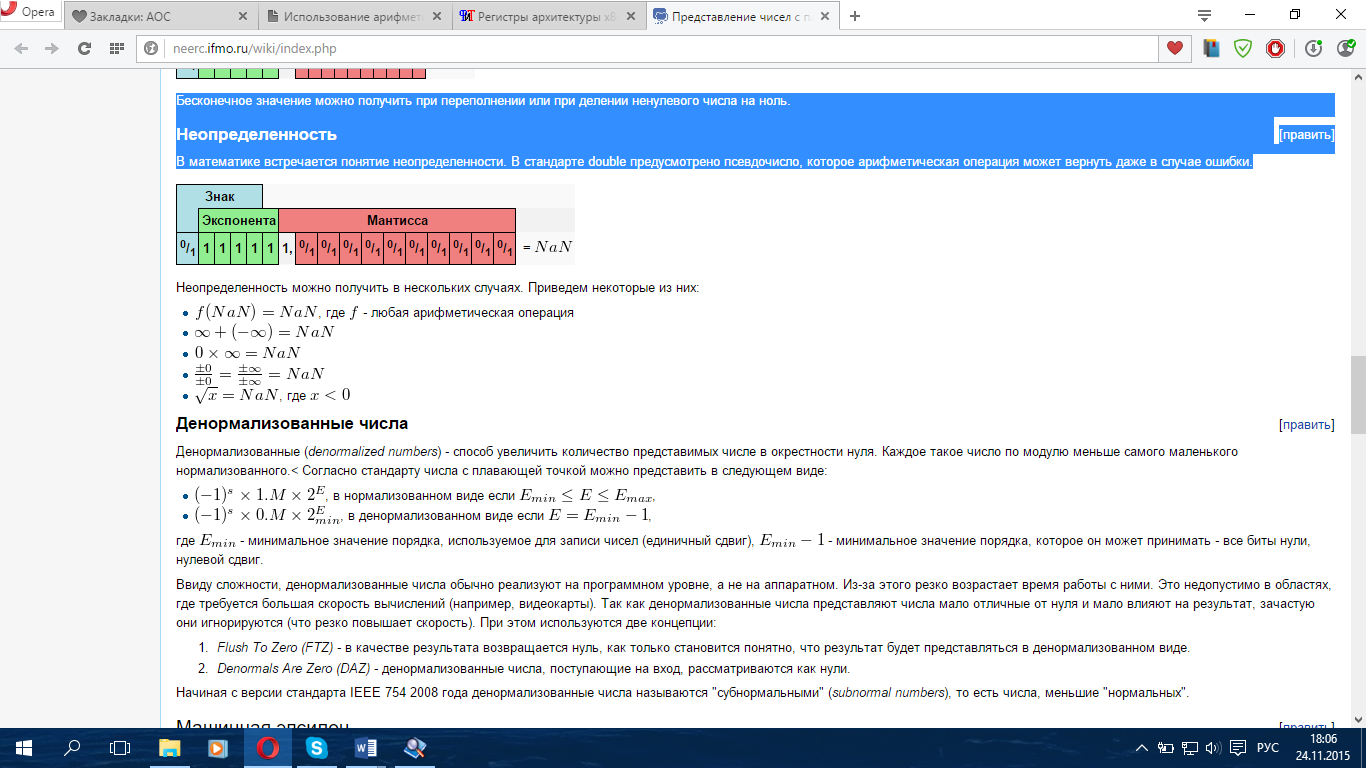
Для приближения ответа к правильному при переполнении, в double можно записать бесконечное значение. Так же, как и в случае с нолем, для этого используются специальные значение мантиссы и экспоненты.



Бесконечное значение можно получить при переполнении или при делении ненулевого числа на ноль.

### *Неопределенность*

В математике встречается понятие неопределенности. В стандарте double предусмотрено псевдочисло, которое арифметическая операция может вернуть даже в случае ошибки.



Неопределенность можно получить в нескольких случаях. Приведем некоторые из них:

* f(NaN) = NaN, где f - любая арифметическая операция
* \infty + (-\infty) = NaN
* 0 \times \infty = NaN
* \frac{\pm0}{\pm0} = \frac{\pm \infty}{\pm \infty} = NaN
* \sqrt{x} = NaN, где x < 0

### *Денормализованные числа*

Денормализованные (*denormalized numbers*) - способ увеличить количество представимых числе в окрестности нуля. Каждое такое число по модулю меньше самого маленького нормализованного. Согласно стандарту числа с плавающей точкой можно представить в следующем виде:

* (-1)^s \times 1.M \times 2^E, в нормализованном виде если E_{min} \leq E \leq E_{max},
* (-1)^s \times 0.M \times 2^E_{min}, в денормализованном виде если E = E_{min} - 1,

где E_{min} - минимальное значение порядка, используемое для записи чисел (единичный сдвиг), E_{min} - 1 - минимальное значение порядка, которое он может принимать - все биты нули, нулевой сдвиг.

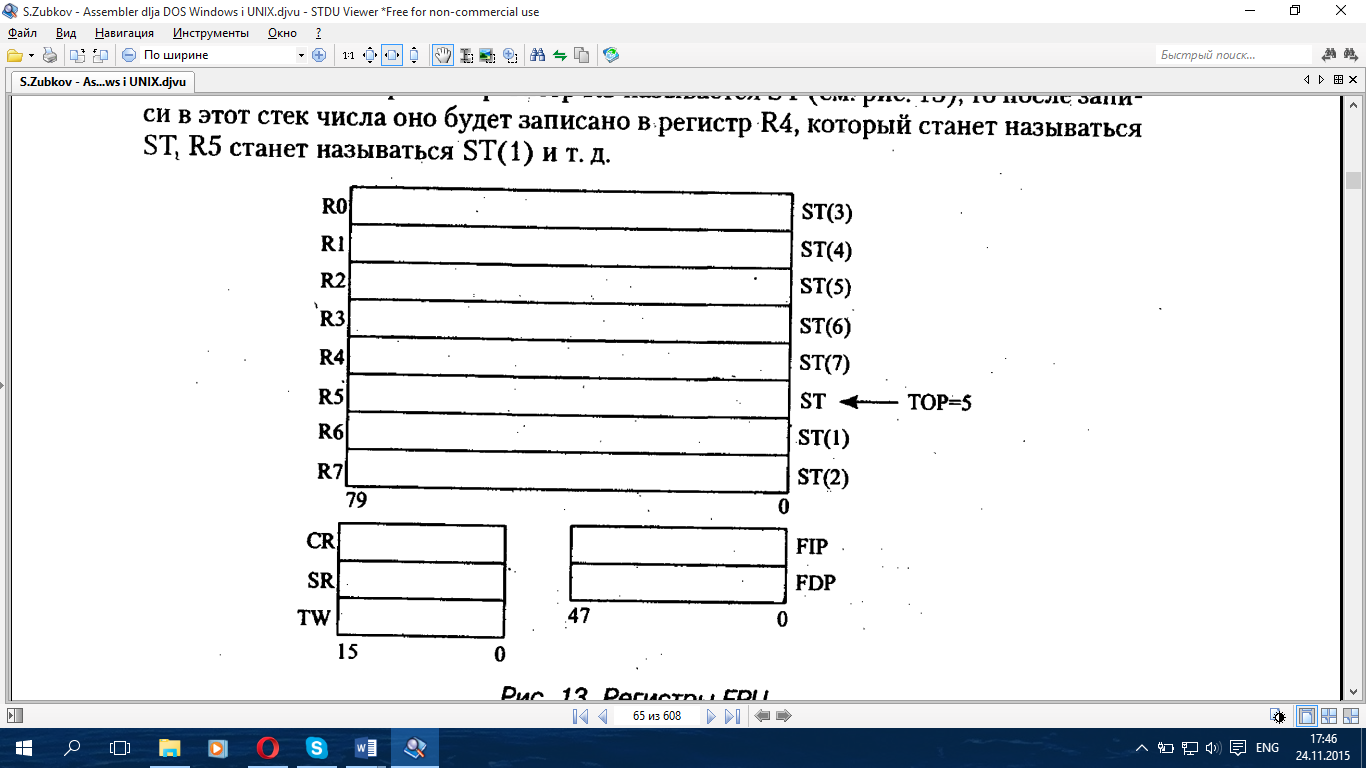
Ввиду сложности, денормализованные числа обычно реализуют на программном уровне, а не на аппаратном. Из-за этого резко возрастает время работы с ними. Это недопустимо в областях, где требуется большая скорость вычислений (например, видеокарты). Так как денормализованные числа представляют числа мало отличные от нуля и мало влияют на результат, зачастую они игнорируются (что резко повышает скорость). При этом используются две концепции:

1. *Flush To Zero (FTZ)* - в качестве результата возвращается нуль, как только становится понятно, что результат будет представляться в денормализованном виде.
2. *Denormals Are Zero (DAZ)* - денормализованные числа, поступающие на вход, рассматриваются как нули.

Начиная с версии стандарта IEEE 754 2008 года денормализованные числа называются "субнормальными" (*subnormal numbers*), то есть числа, меньшие "нормальных".

***Виконання роботи***

*Регістри арифметичного сопроцесору:*



*Лістинг програми*

*.model small*

*.stack 100h*

*.data*

*a dd 186*

*b dd -87*

*result dd ?*

*.code*

*start:*

*mov ax,@data*

*mov ds,ax*

*finit*

*fst a;ST(0)=186*

*fst b;ST(1)=186, ST(0)=-87*

*fadd a,b; a+b*

*fsub ;a-b*

*fdiv b;a/b*

*fld [result];ST(0)=186/(-87)*

*mov ax, 4C00h*

*int 21h*

*end start*

***Висновки:***

В даній лабораторній роботі було розглянуто формати представлення чисел з плаваючою комою. Також були розглянуті основи роботи з сопроцесором, були розглянуті такі основні команди: FINIT, FLD, FILD, FST, FIST. Над заданими в роботі числами були виконані такі операції: додавання, віднімання, множення, ділення.