

Арифметика с плавающей точкой

Объявление вещественных переменных

Free Pascal	NASM	размер
x: real размер: single или double (определяется реализацией)		
a: single число одинарной точности	a dd 1.8 a resd 1	32 разряда
b: double число двойной точности	b dq 1.8 b resq 1	64 разряда
c: extended число расширенной точности	c dt 1.8 c resq 1	80 разрядов

$$N = (-1)^s * M * 2^{p-b}$$

s — знаковый бит — 1 разряд

1 ≤ M < 2 — мантисса. Занимает 23, 52 или 64 разряда

(в 32- и 64- разрядном представлении целая часть мантиссы = 1 не хранится в памяти.)

p — порядок, занимает 8, 11 или 15 разрядов

b — смещение: 127, 1023 или 16383 соответственно

Загрузка данных на FP-регистр (на вершину стека)

(при этом в регистре SR уменьшится top, бывший st0 станет st1)

fld op ; op – sti, m32, m64, m80.

Выгрузка данных из FP-регистра (с вершины стека)

fst op ; op – sti, m32, m64 – копирование, стек не изменится

fstp op ; op – sti, m32, m64, m80 – выталкивание, при этом в регистре SR уменьшится top, бывший st1 станет st0

Загрузка целого числа на вершину стека

fld op ; op – целое, m32, m64, m80, будет преобразовано в вещественное

Выгрузка числа с вершины стека с округлением до целого

fist op ; op – целое, m16, m32, стек не изменится

fistp op ; op – целое, m16, m32, m64, будет вытолкнуто из стека

Обмен элементов стека

fxch op1, op2 ; op1, op2 – sti, регистры не должны быть пустыми

fxch ; поменять значения st0 и st1

Загрузка констант на вершину стека

fld1 ; 1.0

fldz ; 0.0

fldpi ; π

fldl2e ; log_ee

fldl2t ; log₂10

fldln2 ; ln 2

fldlg2 ; lg 10

Арифметические операции

”Невыталкивающий” формат, количество элементов в стеке не изменится

КОД op1, op2 ; op1 := op1 операция op2, операнды – регистры *sti*, хотя бы один из них - *st0*
КОД op ; *st0* := *st0* операция op op – второй операнд - m32, m64 или регистры *sti*,
КОД ; операнды неявные *st0* := *st0* операция *st1*

”Выталкивающий” формат (команды оканчиваются на p).

Из стека будет удален верхний элемент, операнды (неявные или явно указанные) – только регистры *sti*
Второй операнд – явно или неявно указанный – это *st0*

КОДp op1, op2 ; op := op1 операция op2, op1 – регистры *sti*, op2 – это *st0*
КОДp op ; op := op операция *st0*, op – регистр *sti*
КОДp ; из стека выталкиваются *st0* и *st1*, результат (*st1* операция *st0*) записывается
; в стек как новый *st0*

КОД: fadd (сложение), **fsub** (вычитание), **fmul** (умножение), **fdiv** (деление),
r – с реверсом (обратным порядком) операндов:
fsubr (вычитание второго операнда из первого, запись результата в *st0*)
fdivr (деление второго операнда на первый, запись результата в *st0*)

С преобразованием целого (в названии команды - *i*)

КОД op ; *st0* := *st0* операция op, op – m16, m32 - целое

КОД: fiadd (сложение), **fisub** (вычитание), **fimul** (умножение), **fidiv** (деление),
r – с реверсом (обратным порядком) операндов:
fisubr (вычитание второго операнда из первого, запись результата в *st0*)
fidivr (деление второго операнда на первый, запись результата в *st0*)

Сравнение

fcom st0, op ; op - это *sti*, вычисляется (*st0* – op), устанавливаются флаги, стек не меняется
fcom op ; m32, m64 или *sti*, вычисл. (*st0* – op), установл. флаги, стек не меняется
fcom ; вычисл. *st0* – *st1*, устанавливаются флаги, стек не меняется

fcomp st0, op ; op - это *sti*, вычисл. (*st0* – op), установл. флаги, *st0* выталкивается из стека
fcomp op ; m32, m64 или *sti*, вычисл. (*st0* – op), установл. флаги, *st0* выталкивается из стека
fcomp ; вычисл. (*st0* – *st1*), устанавливаются флаги, *st0* выталкивается из стека

fcompp ; *st0* – *st1*, устанавливаются флаги, *st0* и *st1* выталкиваются из стека

У арифметического сопроцессора своя система флагов, команд условного перехода по ним нет. Из регистра SR (State Register) используем флаги **C3** (аналог ZF), **C0** (аналог CF) и **C2** (устанавливается в 1, если числа не сравнимы, иначе =0). Чтобы воспользоваться результатом сравнения

fstsw ax ; копируем SR в регистр ax

sahw ; копируем ah (некоторые флаги) в регистр флагов, CF:=C0, ZF:=C3, PF:=C2

; После этих команд можно использовать условные переходы (как при беззнаковом сравнении)

fucomi sti ; сравнение *st0* и *sti*, установка ZF и CF

fucomip sti ; сравнение *st0* и *sti*, уст. ZF и CF, выталкивание верхушки стека

Некоторые математические функции

fsin ; st0:= sin (st0)

fcos ; st0:= cos (st0)

fsqrt ; st0:=sqrt(st0)

fptan ; st1 := tg(st0), st0 := 1,

; далее либо командой fdivrp получаем в st0 tg,

; либо командой fdivrp получаем в st0 ctg

f2xm1 ; st0 := $2^{st0} - 1$

fy12x ; st0 := st1 * $\log_2(st0)$

fy12xp1 ; st0 := st1 * $\log_2(st0+1)$