Системное программирование

Лекция 4-5

Представление чисел с плавающей точкой.

Типы данных математического сопроцессора.

Особенности представления чисел.

Команды сопроцессора

Основные понятия

FPU – Floating Point Unit, специальное устройство предназначенное для выполнения команд обработки вещественных данных. В современных процессорах является составной частью основного процессора.

Обеспечивает полную поддержку стандартов IEEE-754 и IEEE-854 по представлению и обработке данных с плавающей точкой.

Числа с плавающей точкой

- `(знак)(мантисса)*10(знак)(порядок)`

  `пример:-9.8765432*10-9`
- **нормализованное представление чисел**

  – целая часть мантиссы числа состоит из одной, не равной нулю, цифры
- для фиксированной разрядной сетки числа нормализованные числа имеют наибольшую точность.
- нормализованное представление исключает неоднозначность – каждое число с плавающей точкой может быть представлено различными (ненормализованными) способами

Стандарт IEEE 754

	Одинарный	Одинарный расширенный	Двойной	Двойной расширенный
Слово (бит)	32	≥43	64	≥ 79
Порядок (бит)	8	≥11	11	≥15
Смещение порядка	127	-	1023	-.
Значения порядка	-126÷127	≤-1022 ÷≥1023	-1022 ÷1023	≤-16382 ÷≥ 16383
Мантисса (бит)	24/23	≥31	52	≥63
Диапазон модулей нормализованных	$\approx 10^{-38} \div 10^{38}$	-	$\approx 10^{-308} \div 10^{308}$	-
Минимальное ненормализованно е	$\approx 10^{-45}$	-	$\approx 10^{-324}$	-

Представление вещественных чисел

$X = \pm mq^{\pm p}$

		Модуль порядка				Модуль мантиссы			
$\pm m$	$\pm p$	p_{n-1}	…	p_1	p_0	m_{-1}	m_{-2}	…	m_{-n}

	Смещённый порядок					Модуль мантиссы			
$\pm m$	p_n	p_{n-1}	…	p_1	p_0	m_{-1}	m_{-2}	…	m_{-n}

- **Нормализация числа – скрытая единица**

Для того, чтобы определить абсолютное значение числа с плавающей точкой, можно воспользоваться следующими формулами:

- одинарная точность: 1.(цифры мантиссы)*2(P-127)
- двойная точность: 1.(цифры мантиссы)*2(P-1023)
- расширенная точность: 1.(цифры мантиссы)*2(P-16383)

Представление вещественных чисел

$X = \pm mq^{\pm p}$

		Модуль порядка				Модуль мантиссы			
$\pm m$	$\pm p$	p_{n-1}	…	p_1	p_0	m_{-1}	m_{-2}	…	m_{-n}

	Смещённый порядок					Модуль мантиссы			
$\pm m$	p_n	p_{n-1}	…	p_1	p_0	m_{-1}	m_{-2}	…	m_{-n}

Рассмотрим число с одинарной точностью со смещённым порядком :

1 01111110 11000000000000000000000

знаковый бит равен 1 (отрицательное число)

смещённый порядок равен 126

мантисса – 1.11 (в двоичной системе счисления)

Значение этого числа равно:

-1.11 * 2(126-127) = -(1+1/2+1/4) * 2(-1) =-1,75 /2= -0,875

Представление вещественных чисел

- Алгоритм представления:

1. Перевести число из Р-ичной системы в двоичную
2. Представить двоичное число в нормализованной экспоненциальной форме
3. Рассчитать смещённый порядок числа
4. Разместить знак, порядок и мантиссу в соответствующие разряды

Типы данных сопроцессора

Тип данных	**Длина (бит)**	**Количество значащих цифр**	**Диапазон представления**
Целое слово	16	4-5	-32768 … 32767
Короткое целое	32	9-10	-2147483648 … 2147483647
Длинное целое	64	18-19	-9223372036854775808 … 9223372036854775807
Короткое вещественное	32	7-8	1.175494351*E-38 … 3.402823466*E+38
Длинное вещественное	64	15-16	2.2250738585072014*E-308 … 1.7976931348623158*E+308
Расширенное веществе[нное]	80	19-20	3.36210314311209350 63*E-4932 … 1.18973149535723176 5*E+4932

Особенности представления чисел

- Наименьшее положительное:

	Смещённый порядок					Модуль мантиссы			
0	0	0	…	0	1	0	0	…	0

- Наибольшее отрицательное:

	Смещённый порядок					Модуль мантиссы			
1	0	0	…	0	1	0	0	…	0

- Наибольшее положительное:

	Смещённый порядок					Модуль мантиссы			
0	1	1	…	1	0	1	1	…	1

- Наименьшее отрицательное:

	Смещённый порядок					Модуль мантиссы			
1	1	1	…	1	0	1	1	…	1

Особенности представления чисел

- Неоднозначность нуля:

	Смещённый порядок					Модуль мантиссы			
±	0	0	…	0	0	0	0	…	0

- Бесконечность INF:

	Смещённый порядок					Модуль мантиссы			
±	1	1	…	1	1	0	0	…	0

- Не число NAN:

	Смещённый порядок					Модуль мантиссы			
1	1	1	…	1	1	x	x	…	x

- Неопределенность:

	Смещённый порядок					Модуль мантиссы			
1	1	1	…	1	1	1	0	…	0

Особые случаи

- ***Неточный результат***

  В результате выполнения некоторых операций может возникнуть такая ситуация, когда невозможно точно представить результат.

  Обычно неточный результат является результатом округления и может не рассматриваться как ошибка.

Пример:

> результатом деления числа `1.0` на `3.0` является бесконечная периодическая двоичная дробь `0.010101...`

Такое число не может быть представлено точно ни в одном формате вещественных чисел.

Особые случаи

- ***Недействительная операция***

  Этот особый случай возникает при попытке выполнения таких запрещенных команд, как
  - деление нуля на нуль;
  - извлечения корня из отрицательного числа,
  - обращение к несуществующему регистру сопроцессора;
  - при попытке использования в качестве операндов команд нечисел, неопределенностей, бесконечности (для трансцендентных функций) или денормализованных чисел .

Особые случаи

- ***Переполнение***

  Если результат выполнения операции слишком велик и не может быть представлен в формате приемника результата.

Пример:

при сложении максимального числа расширенной точности самим с собой;

при преобразовании этого числа в формат с двойной или одинарной точностью.

Особые случаи

- ***Антипереполнение***

  Если результат слишком мал для его представления в формате приемника результата операции, но все же отличен от нуля.

Пример:

> при преобразовании наименьшего положительного числа с расширенной точностью в формат числа с двойной или одинарной точностью.

Особые случаи

- ***Денормализованный операнд***

  при выполнении операции может оказаться, что результат слишком мал по абсолютной величине для представления его в нормализованной форме. Можно было бы считать такой результат нулевым, однако это привело бы к снижению точности вычислений или даже к грубым ошибкам.

Пример:

вычисляется следующее выражение: `(y-x)+x;`

Если разность `(y-x)` вызывает ***антипереполнение*** и в качестве результата берется нулевое значение, то после вычисления всего выражения получится `x`.

Если же пойти на расширение диапазона представления чисел за счет снижения точности и сформировать результат вычисления разности
`y-x)` как ***денормализованное число***, выражение будет вычислено правильно и в результате получится `y`.

Однако при попытке деления на ненормализованное число или извлечения из него квадратного корня фиксируется особый случай

Денормализованные числа

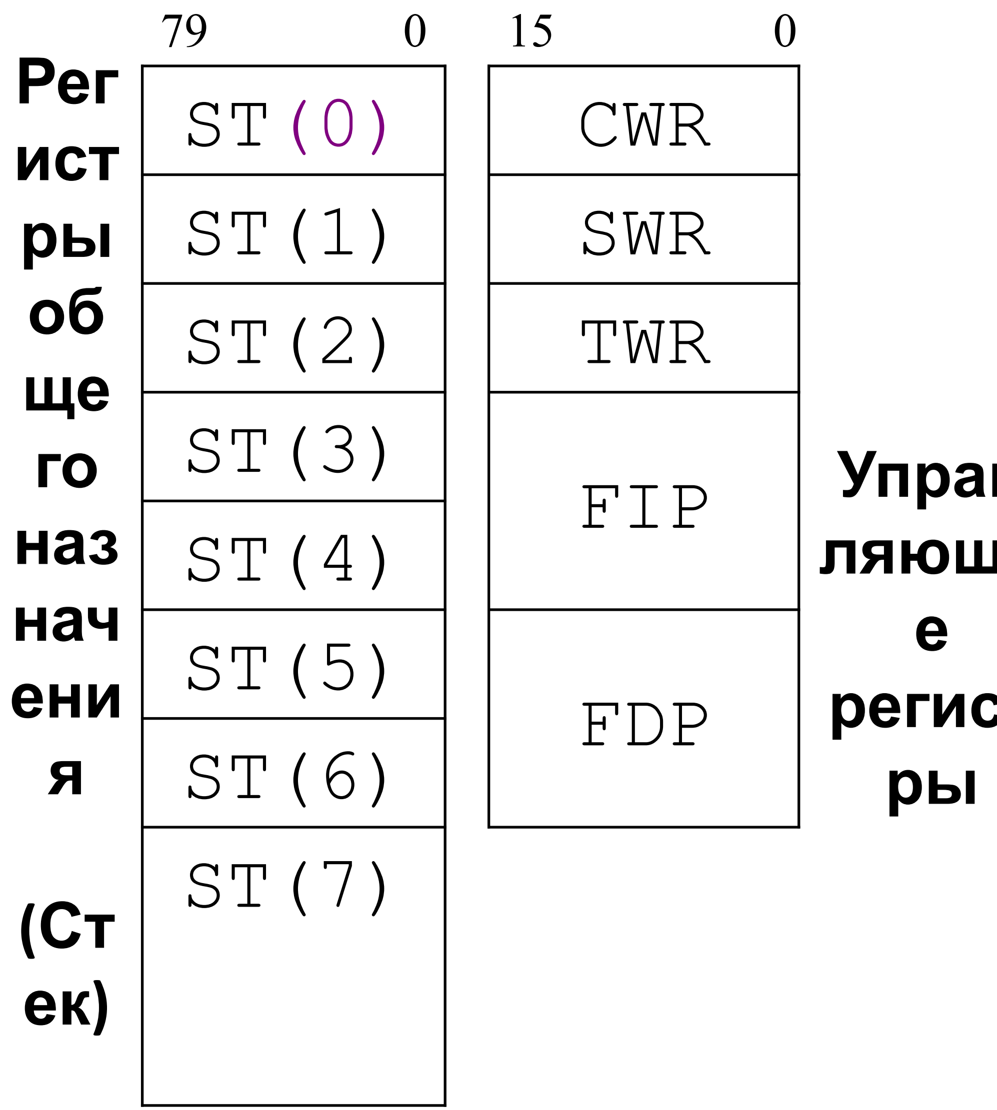
- Формат представления:

	Смещённый порядок					Модуль мантиссы			
±	0	0	…	0	0	x	x	…	x

- Для получения их значения не требуется использование неявной единицы – мантисса умножается на наименьшую для данного формата экспоненту.
- Позволяют представлять очень маленькие числа при вычислениях с расширенной точностью.
- Денормализованные числа находятся ближе к 0, чем наименьшее представимое нормализованное число.

Программная модель сопроцессора

- 8 регистров данных
  - 80 бит
- 3 специальных регистра
  - 16 бит
    - статус
    - управление
    - теги
- 2 регистра указателя
  - 48 бит
  - (команда, операнд)



Состояние FPU

Status Word Register

b	c3	top	c2	c1	c0	es	sf	pe	ue	oe	ze	de	ie
15	14	13 12 11	10	9	8	7	6	5	4	3	2	1	0

Флаги исключений

`I (b0) –` недопустимая операция
`D (b1) –` денормализованный операнд
`Z (b2) –` деление на ноль
`O (b3) –` переполнение (результат - ∞)
`U (b4) –` результат слишком мал для нормализации
`P (b5) –` потеря точности
`E (b6) –` любой из предыдущих
`S (b7) –` ошибка стека
`B (b8) –` занят
`c1 (b9) –` переполнение стека
`c3,c2,c0 (b14,b10,b8)` - сравнение, проверка

Команды сопроцессора

Передачи данных

Сравнения данных

Арифметические

Трансцендентные

Управления

FPU. Особенности

- Расширенная стековая машина
  - Операции со сдвигом стека
  - Два результата одноместной операции
  - Двуместные без сдвига стека
  - Двуместные, где один в памяти
  - Двуместные, где один «в глубине» стека
  - Двуместные с обратным порядком операндов
- Данные - 80 бит (помимо 64 и 32 бит)

Команды сопроцессора

- Без параметров:
  ОП `;ОП ST(1),ST(0) + pop`

- С одним параметром:
  ОП *источник* `; ОП ST(0),`*источник*

- С двумя параметрами:
  ОП *приёмник , источник*

  `;` *приёмник* `=` *приёмник* `ОП` *источник*

`f***p` – после операции производится выталкивание из стека

`f***r(p)` – реверсивное следование операндов в операциях – и /

Сравнение данных

Сравнение данных

- Вещественных

  fcom

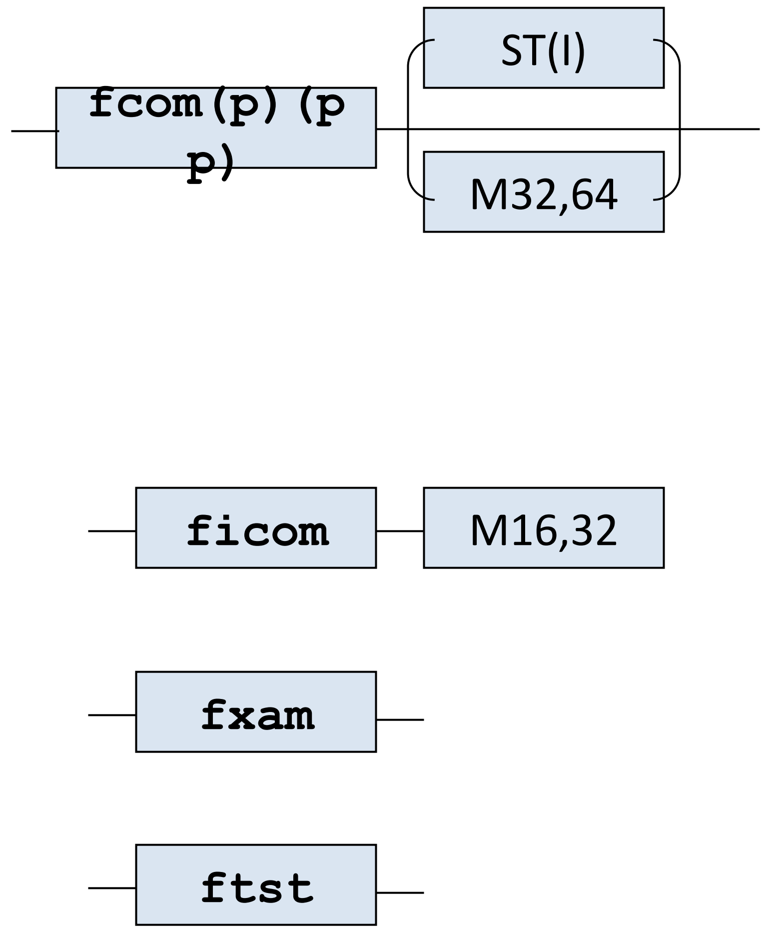
  fcomp

  fcompp
- Целых

  ficom(p)
- Анализ

  fxam
- С нулем

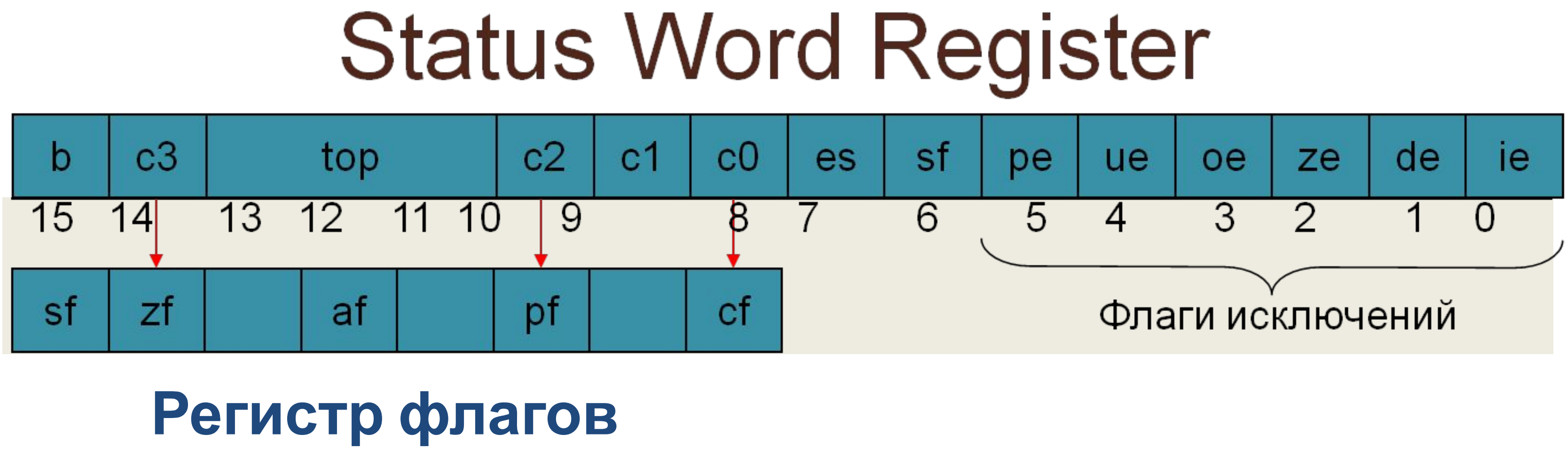
  ftst



FXAM

c1	c3	c2	c0	Содержимое ST
Знак ST	0	0	0	Неизвестный формат
	0	0	1	Не число
	0	1	0	Корректное вещественное число
	0	1	1	Бесконечность
	1	0	0	Нуль
	1	0	1	Пусто
	1	1	0	Денормализованное число

Особенности проверки чисел



	c3	c2	c0
ST > операнд	0	0	0
ST < операнд	0	0	1
ST = операнд	1	0	0
Не сравнимы	1	1	1

```
fstsw ax
sahf
j***
```

Загрузка и выгрузка

Передача данных

- Загрузка

  `fld`

  `fild`

- Выгрузка

  `fst, fstp`

  `fist, fistp`

- Обмен

  `fxch`

- Загрузка констант

  `fldz`

  `fld1`

  `fldpi`

  `fldl2t`

  `fldl2e`

  `fldlg2`

  `fldln2`

(если не указан источник то считается, что он соответствует `ST(1)`)

Все команды имеют один операнд: либо источник либо приемник

Команды сложения и умножения

Арифметические

- Вещественные
- Сложение
  `fadd(p)`
- Вычитание
  `fsub(r) fsub(r)p`
- Умножение
  `fmul(p)`
- Деление
  `fdiv(r) fdiv(r)p`

- Целочисленные
- Сложение
  `fiadd`
- Вычитание
  `fisub(r)`
- Умножение
  `fimul`
- Деление
  `fidiv(r)`

Вспомогательные арифметические

Арифметические

- **fsqrt** – Sqrt(st) □ st
- **fabs** – Abs(st) □ st
- **fchs** – +/- st □ st
- **fxtract** – Мантисса □ st, порядок □st(1)
- **fprem** – st mod st(1) □st
- **fscale** – st*2$^{st(1)}$ □ st
- **frndint** – [st] □ st

Трансцендентные

- **Тригонометрические**

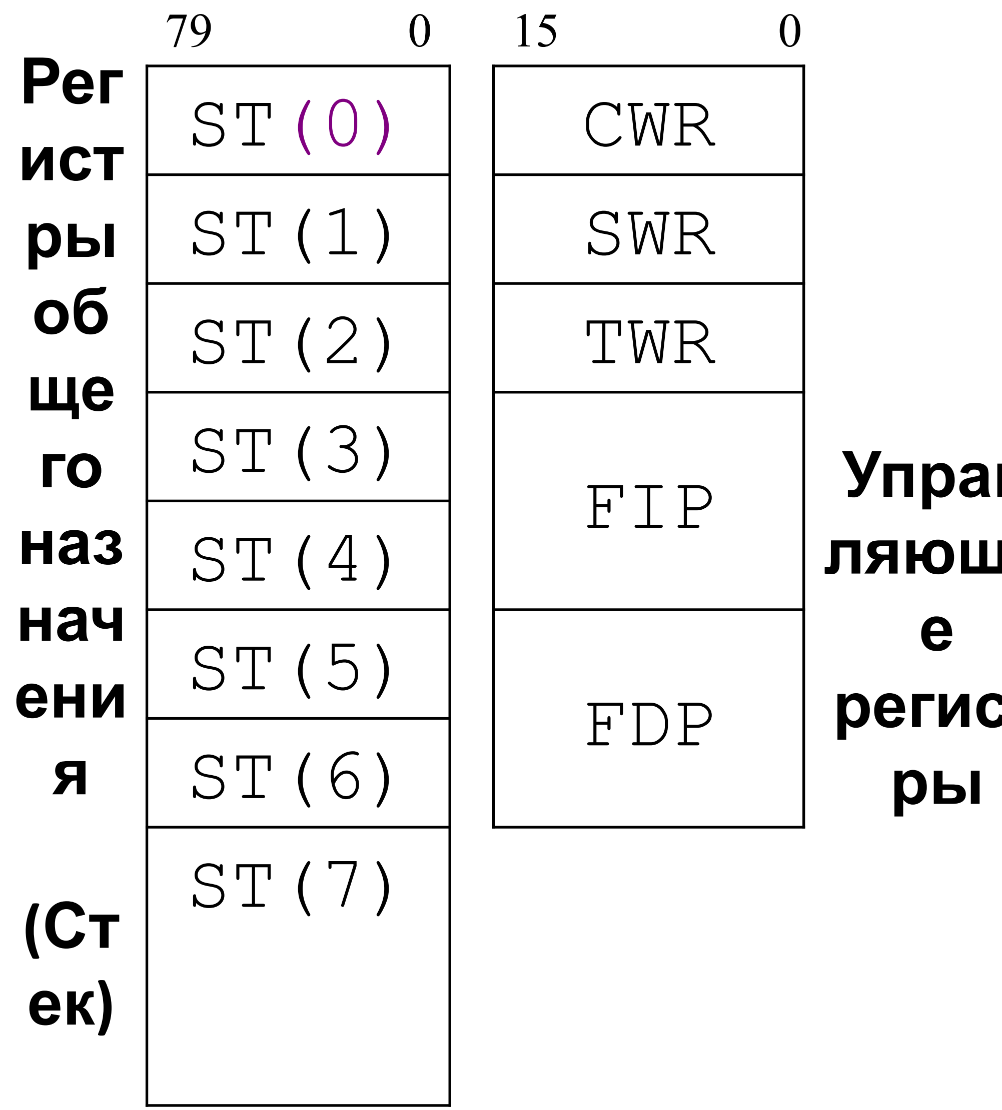
fsin sin(st) □st
fsincos sin(st) □ st, cos(st) □st(1)
fcos cos(st)
fptan tg(st) □ st, fld1
fpatan arctg(st/st(1)) □ st

- **Степенные и логарифмические**

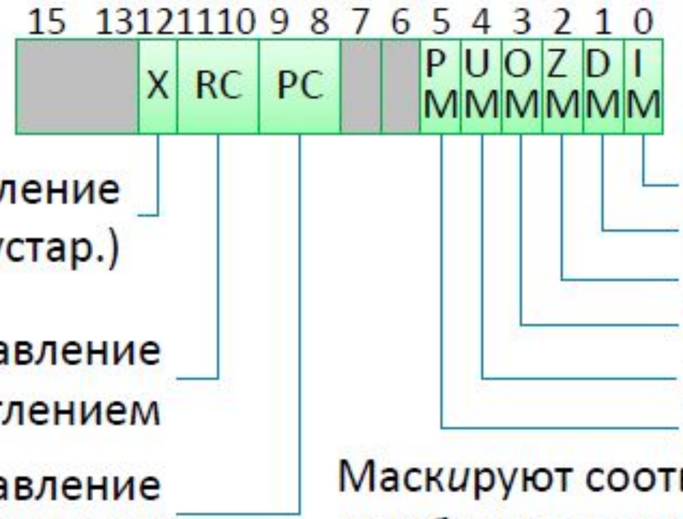
f2xm1 $2^{st}-1$ □ st; -1<x<1
fyl2xp1 st(1)*$\log_2$(st+1) □st(1), pop
fyl2x st(1)*$\log_2$(st) □st(1), pop

Программная модель сопроцессора

- 8 регистров данных
  - 80 бит
- 3 специальных регистра
  - 16 бит
    - статус
    - управление
    - теги
- 2 регистра указателя
  - 48 бит
  - (команда, операнд)



Регистр управления



Управление неопределенностью (устар.)

Round Control – управление округлением

Precision Control – управление точностью (IEEE 754)

Маски ошибок:
- кода операции
- операнда
- деления на 0
- переполнения
- потери значимости
- потери точности

Маскируют соответствующие ошибки исполнения (ошибка не вызывает прерывание)

RC	Режим
00	Округление к ближайшему
01	Округление вниз
10	Округление вверх
11	Округление вниз по модулю

PC	Разрядность мантиссы
00	24
01	Зарезервировано
10	53
11	64

Управление

finit - Сброс
fdecstp/fincstp - Сдвиг стека
ffree - Освобождение регистра
fclex - Сбросить статус
fstsw/fstcw - Считать статус/управление
fldcw - Записать управление
fnop - Пустая операция
fstenv - Сохранить состояние (кроме данных)
fldenv - Восстановить состояние (кроме данных)
fsave - Сохранить состояние полностью и сбросить
frstor - Восстановить состояние полностью
fwait/wait - Задержать ЦП

Регистр тегов

- Регистр TAG содержит двухбитный тег для каждого регистра x87 FPU

TAG7	TAG6	TAG5	TAG4	TAG3	TAG2	TAG1	TAG0

- Значение тегов описывает содержимое регистров

00	Число
01	Ноль
10	Специальное значение (NaN, неопределенность и т.д.)
11	Регистр пуст

- Тег 11 используется сопроцессором для определения ошибки переполнения или опустошения стека регистров
- Читать и изменять состояния TAG можно только командами
  - **FSTENV/FLDENV** – сохранить/загрузить среду x87
  - **FSAVE/FRSTORE** – сохранить/восстановить состояние x87