Лекция 0x10

10 апреля

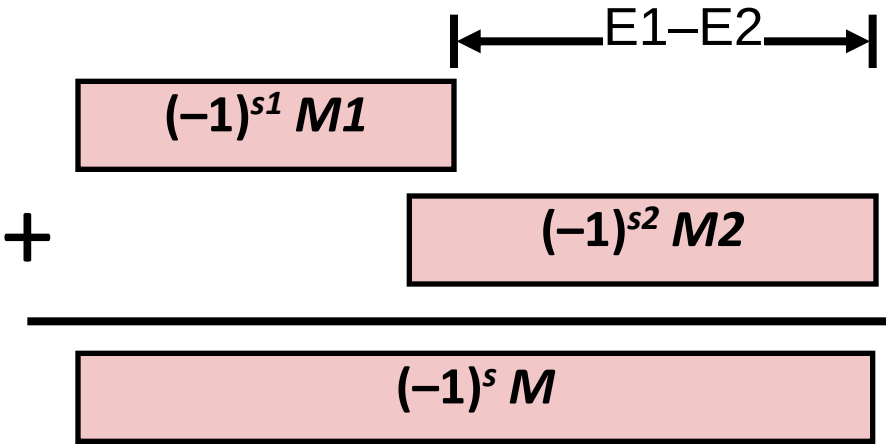
Умножение

- $(-1)^{s1}\ M1\ \ 2^{E1}\ \ x\ \ (-1)^{s2}\ M2\ \ 2^{E2}$
- Точный результат: $(-1)^{s}\ M\ \ 2^{E}$
  - Знаковый бит s: s1 ^ s2
  - Мантисса M: M1 x M2
  - Порядок 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
  - Порядок 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? — Да
    - Результатом может быть бесконечность или NaN
  - Коммутативность? — Да
  - Ассоциативность? — Нет
    - Переполнения и изменение результата при округлении
  - `0.0` – нейтральный элемент? — Да
  - Каждый элемент имеет обратный — Почти всегда
    - За исключением бесконечности и NaN
- Монотонность
  - $a \geq b \Rightarrow a+c \geq b+c$? — Почти всегда
    - За исключением бесконечности и NaN

Математические свойства умножения

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

- Монотонность
  - $a \geq b \ \& \ c \geq 0 \Rightarrow a * c \geq b * c$? — Почти всегда
    - 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 = …;
```

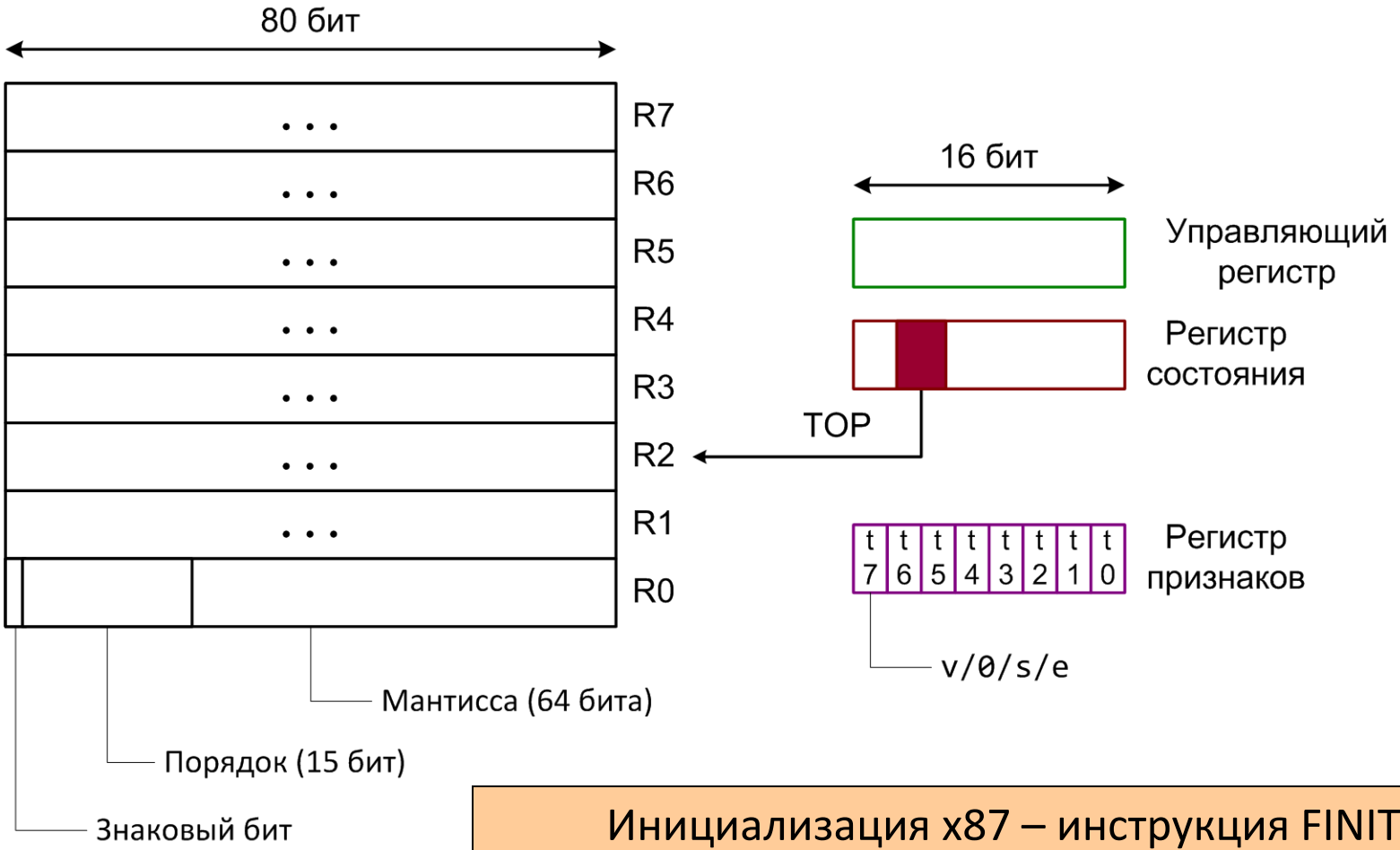
Предполагается, что
d и f не являются NaN

- `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`

Числа с плавающей точкой: промежуточные итоги

- IEEE 754 – четкое определение математических свойств
- Представляются числа вида $\pm M \times 2^E$
- 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

Упрощенная схема x87



Инициализация x87 – инструкция FINIT

```
CW = 0x037F   SW = 0x0000   Tag = 0xFFFF
```

Размер чисел с плавающей точкой

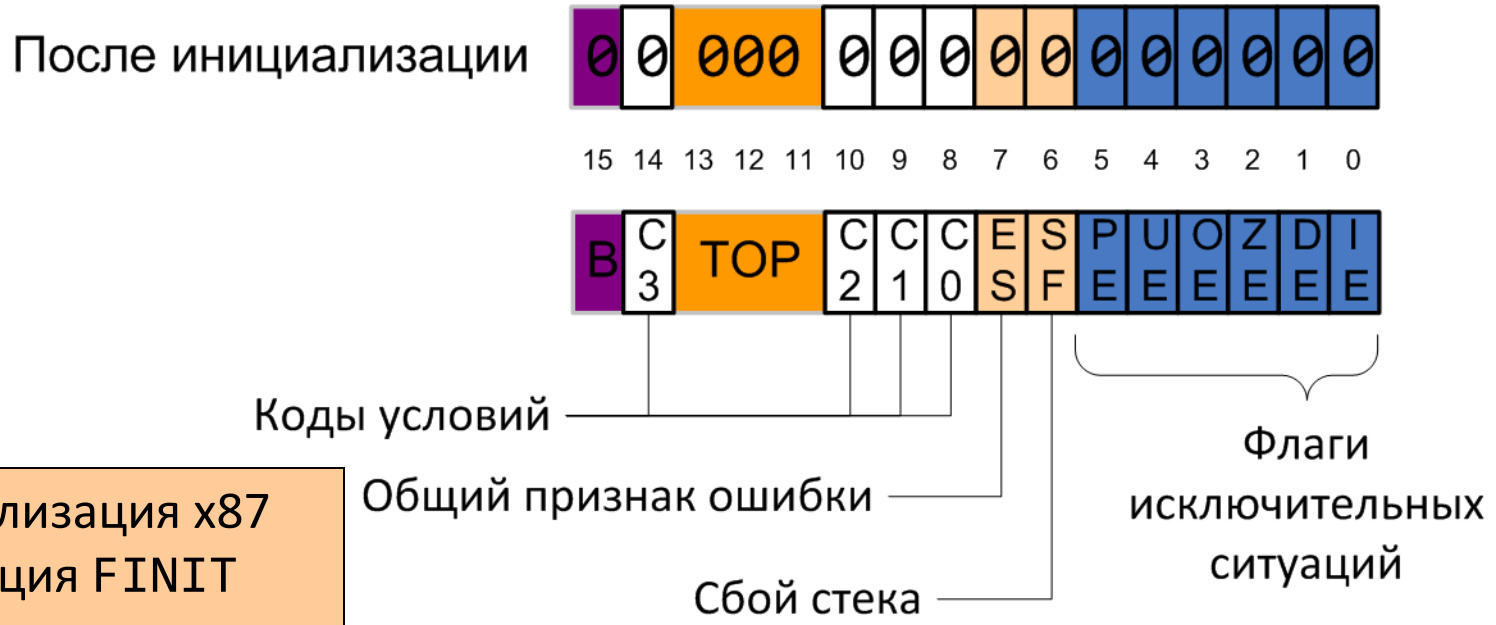
Регистры данных

Обмен данными только с памятью!

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

Слово (регистр) состояния

- SF – переполнение стека (C1 показывает направление)
- Исключительные ситуации: точность, переполнение, деление на ноль, денормализованный операнд, «неправильные» данные

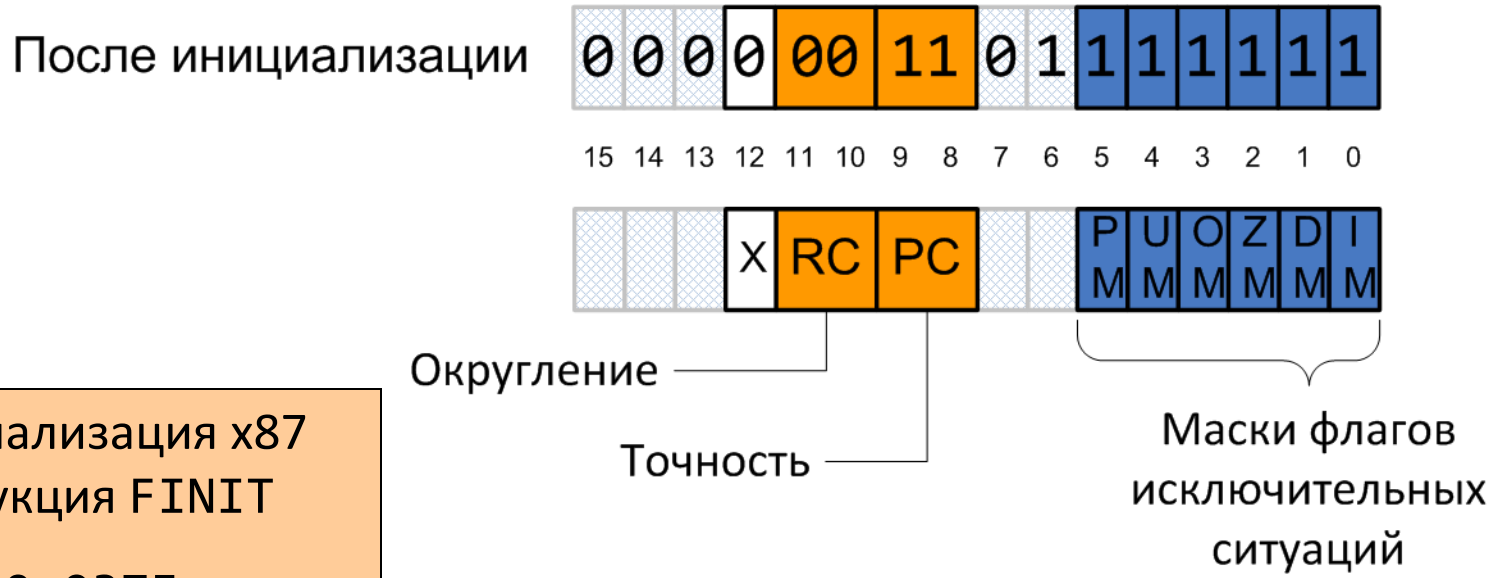


Инициализация x87 инструкция `FINIT`

`SW = 0x0000`

Управляющий регистр

- Точность: одинарная, двойная, расширенная
- Округление: к ближайшему четному, к нулю, к +/- бесконечности
- Флаг X – совместимость с 287
- Маски соответствуют исключениям в слове состояния

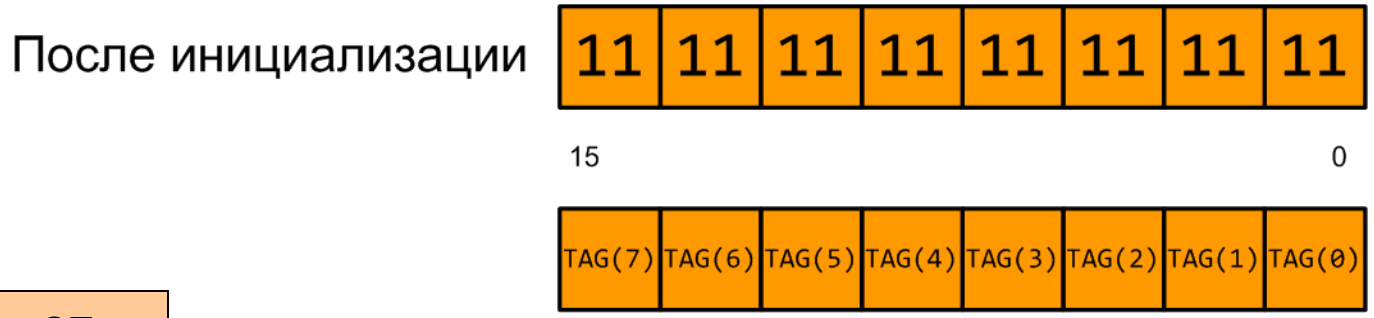


Инициализация x87 инструкция `FINIT`

`CW = 0x037F`

Регистр признаков (тагов)

- Состояние регистров
  - 0 – нормализованное число с плавающей точкой
  - 1 – число ноль
  - 2 – особые числа (NaN, ± ∞, денормализованное число)
  - 3 – регистр свободен
- Нумерация соответствует физическим регистрам



Инициализация x87 инструкция `FINIT`

`Tag = 0xFFFF`

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
```

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

```
__Infinity__
__NaN__
__QNaN__
__SNaN__
```

IEEE 754r – опубликован в 2008 году

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

Сложение двух чисел

```
%include 'io.inc'

section .data
 x dd 11.2
 y dd 0.7

section .bss
 z resd 1

section .text
global CMAIN
```

```
CMAIN:
   finit
   fld dword [x]
   fld dword [y]
   faddp
   fstp dword [z]
   PRINT_HEX 4, z
   NEWLINE
   xor eax, eax
   ret
```

x

11.2 ~ 1.11999980926513671875E1

0x41333333 = 01000001 00110011 00110011 00110011

Сложение двух чисел

```
%include 'io.inc'

section .data
 x dd 11.2
 y dd 0.7

section .bss
 z resd 1

section .text
global CMAIN
```

```
CMAIN:
   finit
   fld dword [x]
   fld dword [y]
   faddp
   fstp dword [z]
   PRINT_HEX 4, z
   NEWLINE
   xor eax, eax
   ret
```

y

0.7 ~ 6.99999988079071044921875E-1

0x3F333333 = 00111111 00110011 00110011 00110011

Сложение двух чисел

```
%include 'io.inc'

section .data
 x dd 11.2
 y dd 0.7

section .bss
 z resd 1

section .text
global CMAIN
```

```
CMAIN:
   finit
   fld dword [x]
   fld dword [y]
   faddp
   fstp dword [z]
   PRINT_HEX 4, z
   NEWLINE
   xor eax, eax
   ret
```

z

0x413E6666 = 0 1000001 0 0111110 01100110 01100110

1.189999961853027343 75E1 ~ 11.9

Печать числа

```
%include 'io.inc'

section .data
 x dd 11.2
 y dd 0.7

section .bss
 z resd 1

section .rodata
 lc db '%f', 10, 0

section .text
CEXTERN printf
global CMAIN
```

```
CMAIN:
   ; …
   ; пролог функции
   sub     esp, 20

   fld     dword [x]
   fld     dword [y]
   faddp
   fst     dword [z]
   fstp    dword [esp + 4]
   mov     dword [esp], lc
   call    printf

   add     esp, 20
   ; эпилог функции
   ; …
```

Печать числа

```
%include 'io.inc'

section .data
 x dd 11.2
 y dd 0.7

section .bss
 z resd 1

section .rodata
 lc db '%f', 10, 0

section .text
CEXTERN printf
global CMAIN
```

printf печатает мусор
где ошибка?!?!!1

```
CMAIN:
    ; …
    ; пролог функции
    sub     esp, 20

    fld     dword [x]
    fld     dword [y]
    faddp
    fst     dword [z]
    fstp    dword [esp + 4]
    mov     dword [esp], lc
    call    printf

    add     esp, 20
    ; эпилог функции
    ; …
```

Печать числа

```
%include 'io.inc'

section .data
 x dd 11.2
 y dd 0.7

section .bss
 z resd 1

section .rodata
 lc db '%f', 10, 0

section .text
CEXTERN printf
global CMAIN
```

ISO/IEC 9899:1999
§ 6.5.2.2 абзац №6

```
CMAIN:
    ; …
    ; пролог функции
    sub     esp, 20

    fld     dword [x]
    fld     dword [y]
    faddp
    fst     dword [z]
    fstp    qword [esp + 4]
    mov     dword [esp], lc
    call    printf

    add     esp, 20
    ; эпилог функции
    ; …
```

Польская обратная запись и x87

```
(w + x + y + z) / 4
```

```
w x + y + z + 4 /
```

```
section .data
 w dq  1e10
 x dq  1e10
 y dq  1e10
 z dq  1e10
 d dd  4
```

```
section .text
    ;…
    fld     qword [w]
    fld     qword [x]
    faddp
    fld     qword [y]
    faddp
    fld     qword [z]
    faddp
    fild    dword [d]
    fdivp ; st1 / st0
    ;…
```

```
w
x
+
y
+
z
+
4
/
```