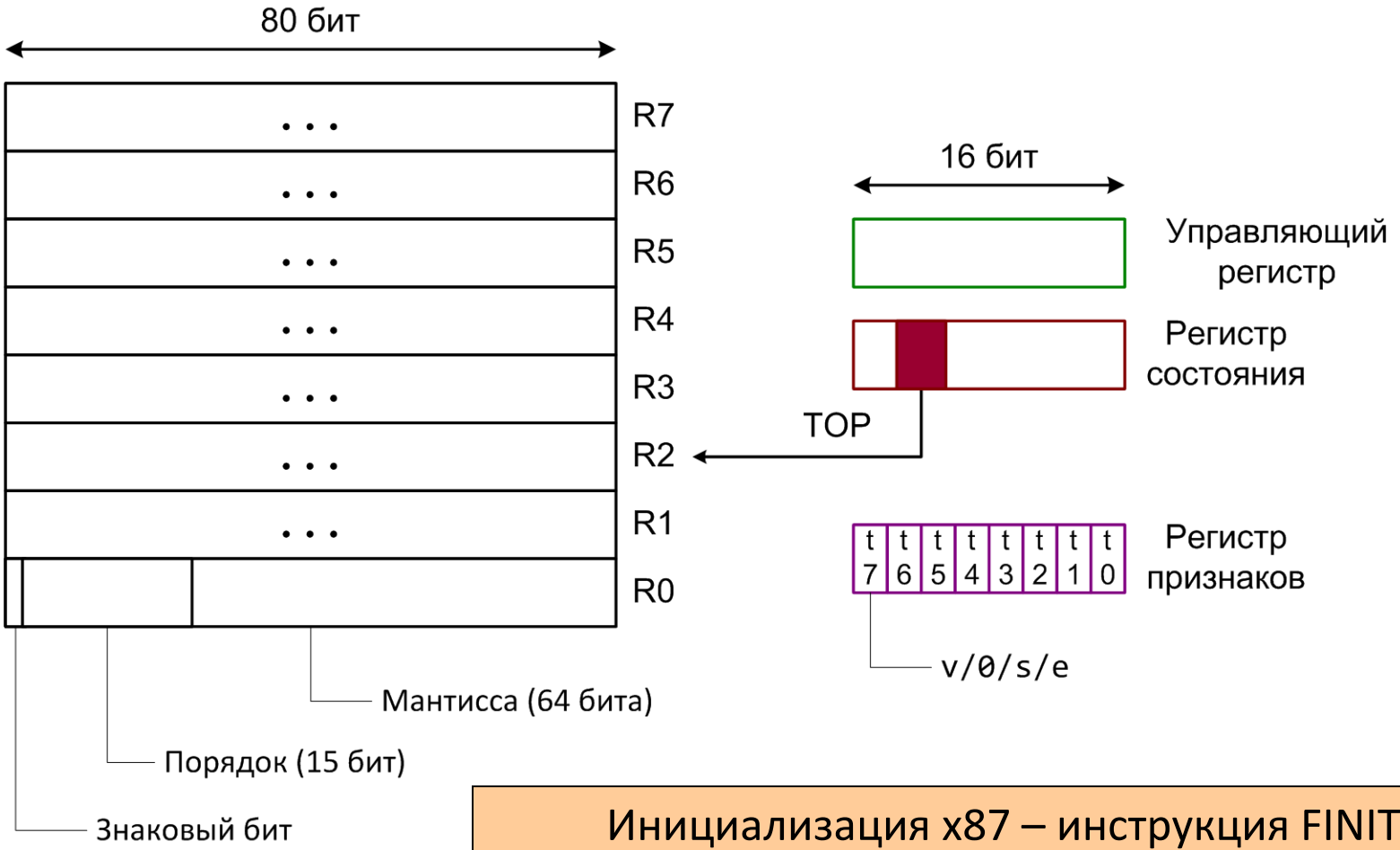
Лекция 18

10 апреля

Числа с плавающей точкой: промежуточные итоги

- IEEE 754 – 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

Упрощенная схема x87

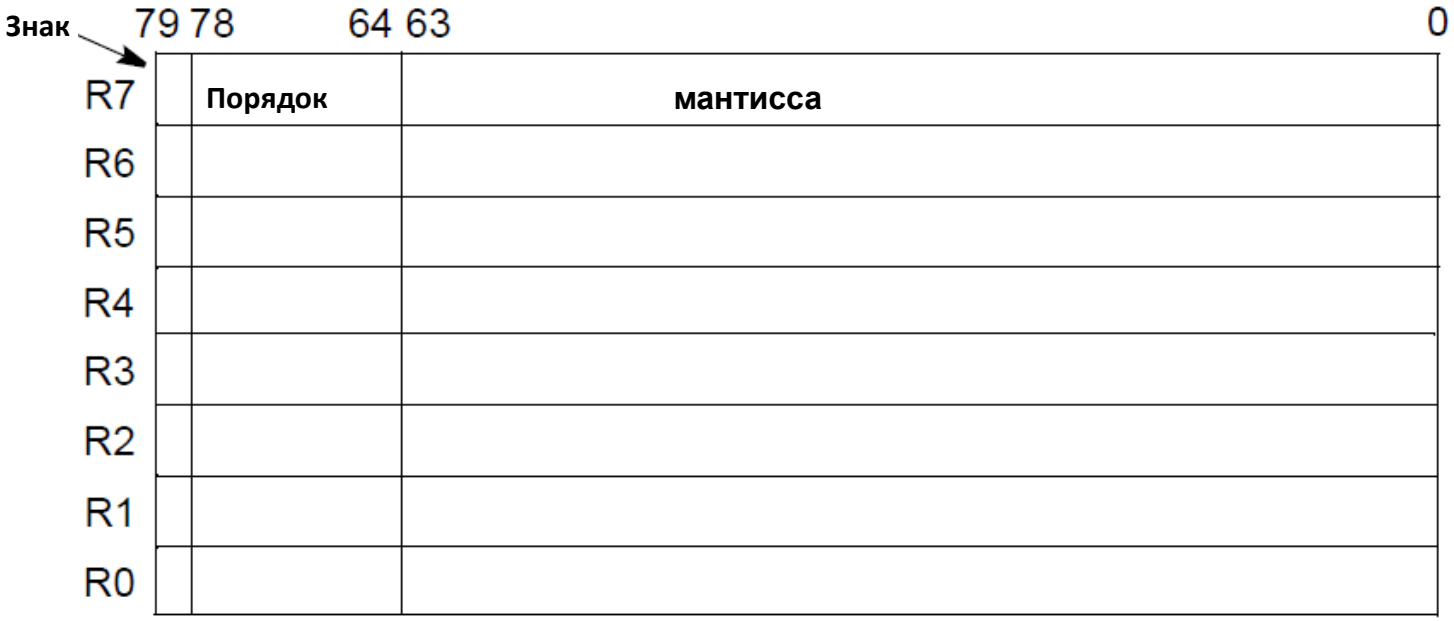


Инициализация x87 – инструкция FINIT

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

Размер чисел с плавающей точкой

Регистры данных

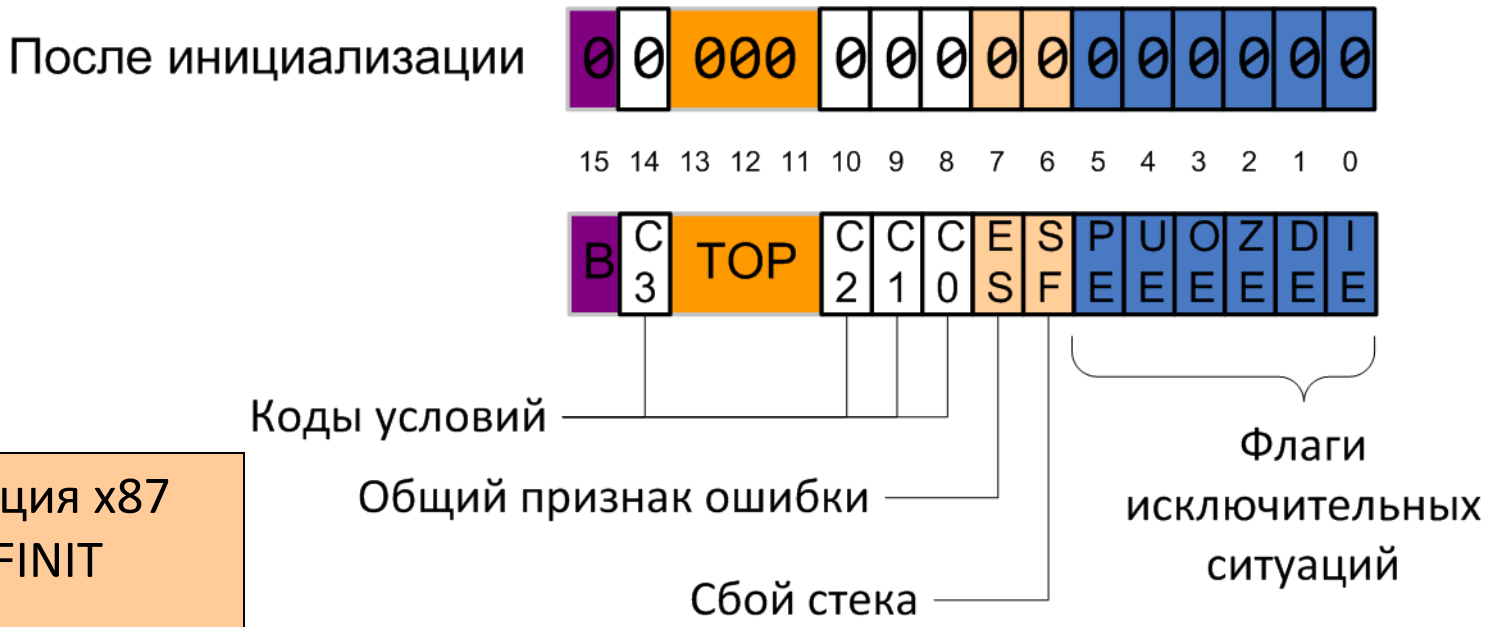


Обмен данными только с памятью.

```
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

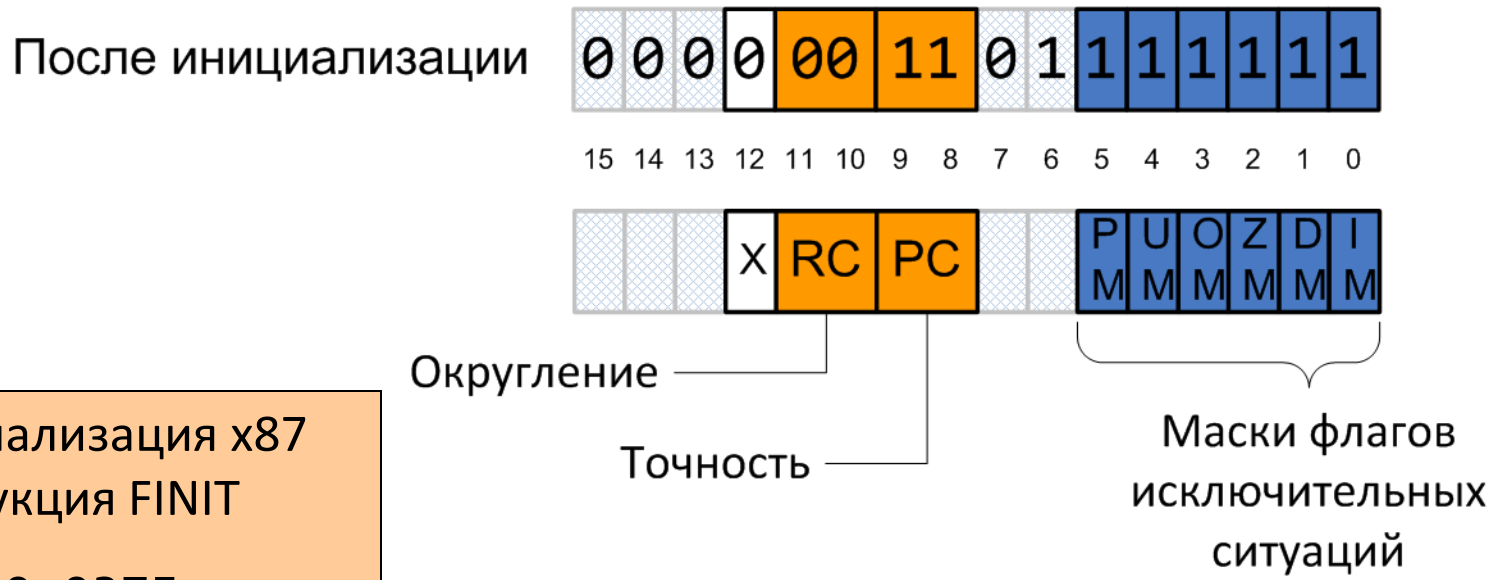


Инициализация x87 инструкция FINIT

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

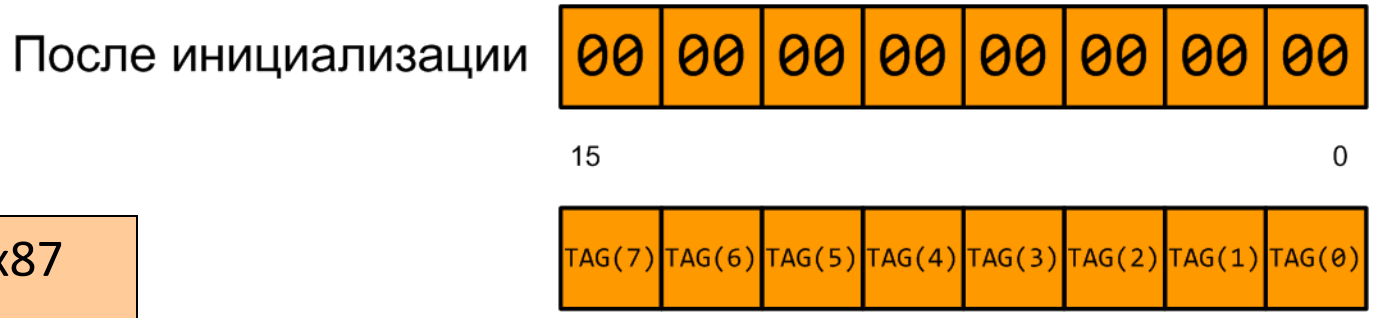


Инициализация x87 инструкция FINIT

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
```

- IEEE 754r – опубликован в 2008 году

NASM и числа с плавающей точкой

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

- __Infinity__
- __NaN__
  - __QNaN__
  - __SNaN__

```
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 ~ 1.11999998092651367875E1

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 = 01000001 00111110 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
    ; …
```

Порядок действий имеет значение

```
%include 'io.inc'

section .data
 x dq  3.14
 y dq  1e50
 z dq -1e50

section .bss
 r resq 1

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

section .text
CEXTERN printf
global CMAIN
```

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

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

    add     esp, 20
    ; … эпилог функции
```

Порядок действий имеет значение

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

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

    add     esp, 20
    ; … эпилог функции
```

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

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

    add     esp, 20
    ; … эпилог функции
```

Распределение слагаемых

```
section .data
 x dq  1e200
 y dq  1e200
 z dq  1e200

section .bss
 r resq 1

section .rodata
 lc db '%lf', 10, 0
```

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

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

    add     esp, 20
    ; … эпилог функции
```

Распределение слагаемых

```
section .data
 x dq  1e200
 y dq  1e200
 z dq  1e200

section .bss
 r resq 1

section .rodata
 lc db '%lf', 10, 0
```

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

   fld     qword [x]
   fld     qword [z]
   fmulp
   fld     qword [y]
   fld     qword [z]
   fmulp
   fsubp
   ; вызов printf

   add     esp, 20
   ; … эпилог функции
```

Распределение слагаемых

```
section .data
 x dq  1e200
 y dq  1e200
 z dq  1e200

section .bss
 r resq 1
 cw resw 1

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

CMAIN:
   ; … пролог функции
   sub     esp, 20
   fstcw   word [cw]
   and  word [cw], 11111111_11000000b
   fldcw   word [cw]

   fld     qword [x]
   fld     qword [z]
   fmulp
   fld     qword [y]
   fld     qword [z]
   fmulp
   fsubp
   ; вызов printf

   add     esp, 20
   ; … эпилог функции
```

Польская обратная запись

- (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
    ;…
```

Предопределенные константы

- На «верхушку» стека регистров (ST0) помещается определенная константа
  - FLD1 +1.0
  - FLDL2T $\log_2 10$
  - L2E $\log_2 e$
  - FLDPI π
  - FLDLG2 $\log_{10} 2$
  - FLDLN2 $\log_e 2$
  - FLDZ +0.0

Сравнение чисел

```
_Bool isLe(double x, float y) {
   return x <= y;
}
```

Результат сравнения	C3	C2	C0
St0 > St1	0	0	0
St0 < St1	0	0	1
St0 == St1	1	0	0
неопределенно	1	1	1

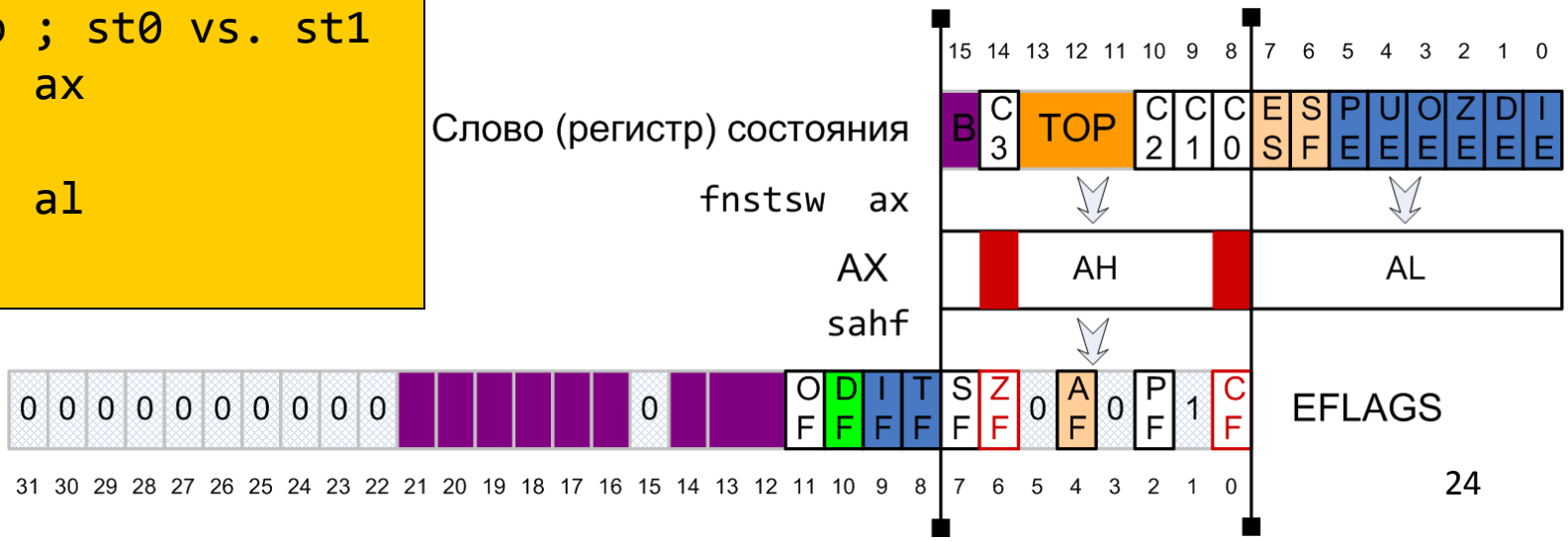
```
isLe:
        push    ebp
        mov     ebp, esp
        fld     dword [ebp+16]
        fld     qword [ebp+8]
        fucompp ; st0 vs. st1
        fnstsw  ax
        sahf
        setbe   al
        pop     ebp
        ret
```

Извлечение результатов сравнения

- C3 → ZF, C0 → CF
- Можно использовать условные коды, применяемые при сравнении беззнаковых чисел

Результат сравнения	C3	C2	C0
ST(0) > ST(i)	0	0	0
ST(0) < ST(i)	0	0	1
ST(0) == ST(i)	1	0	0
неупорядочены	1	1	1

```
isLe:
    ...
    fld     dword [ebp+16]
    fld     qword [ebp+8]
    fucompp ; st0 vs. st1
    fnstsw  ax
    sahf
    setbe   al
    ...
```



Непосредственное обновление EFLAGS

- F[U]COMI[P]
  - появились в P6

Результат сравнения	ZF	PF	CF
ST(0) > ST(i)	0	0	0
ST(0) < ST(i)	0	0	1
ST(0) == ST(i)	1	0	0
неупорядочены	1	1	1

```
isLe(2.0, 3.0)
```

```
isLe:
    ...
    fld     dword [ebp+16]
    fld     qword [ebp+8]
    fucomip ; st0 vs. st1
    setbe   al
    ...
```

BE:
CF = 1 или ZF = 1

Функции: возвращение числа с плавающей точкой

```
void caller(double *p) {
   *p = inverse(*p);
}

caller:
   push    ebp
   mov     ebp, esp
   sub     esp, 8
   mov     eax, dword [ebp+8]
   fld     dword [eax]
   fstp    qword [esp]
   call    inverse
   mov     eax, dword [ebp+8]
   fstp    qword [eax]
   leave
   ret
```

```
float inverse(double x) {
   return 1/x;
}

inverse:
   push    ebp
   mov     ebp, esp
   fld1
   fld     qword [ebp+8]
   fdivp
   pop ebp
   ret
```