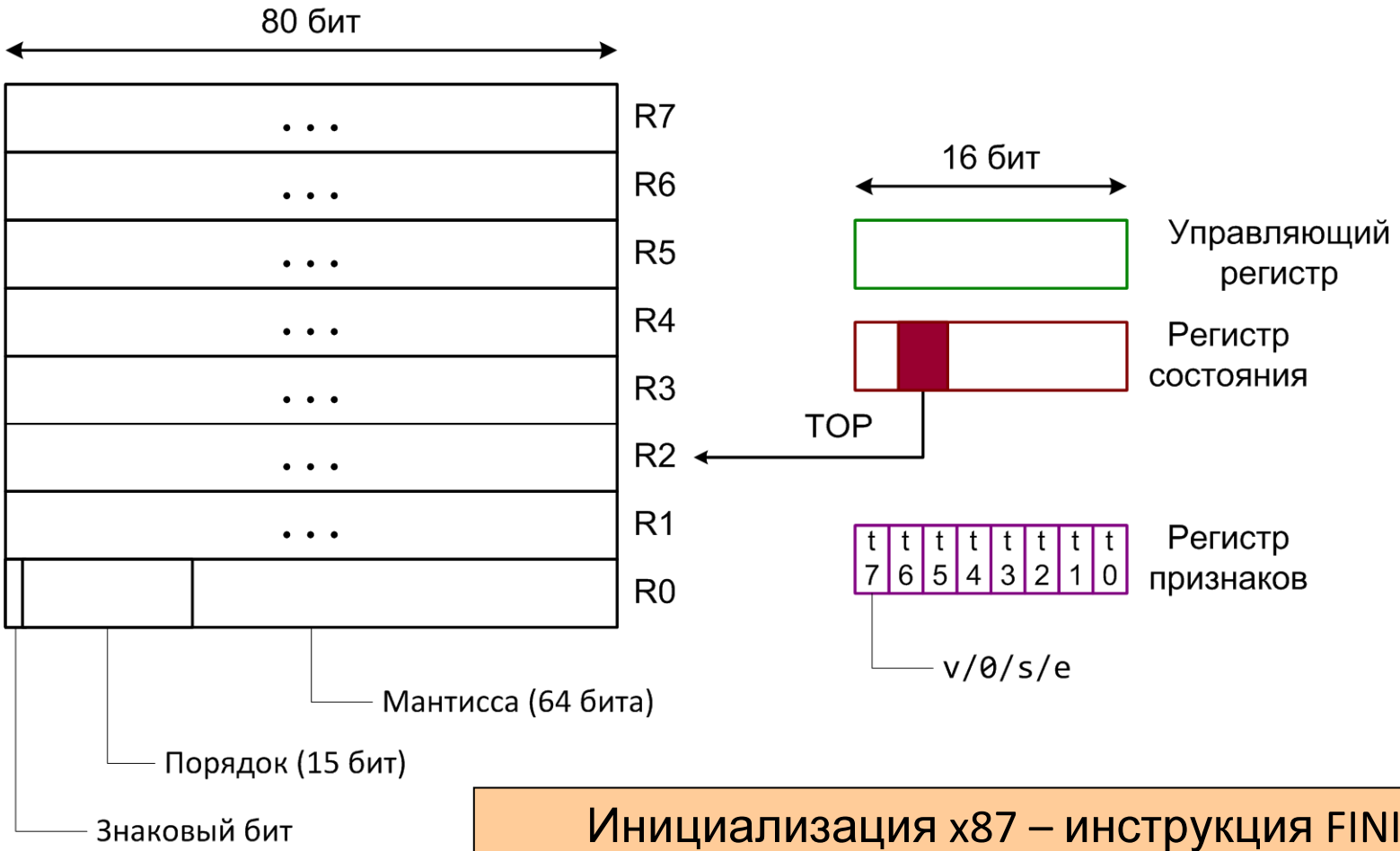
Лекция 15

2 апреля

Упрощенная схема x87

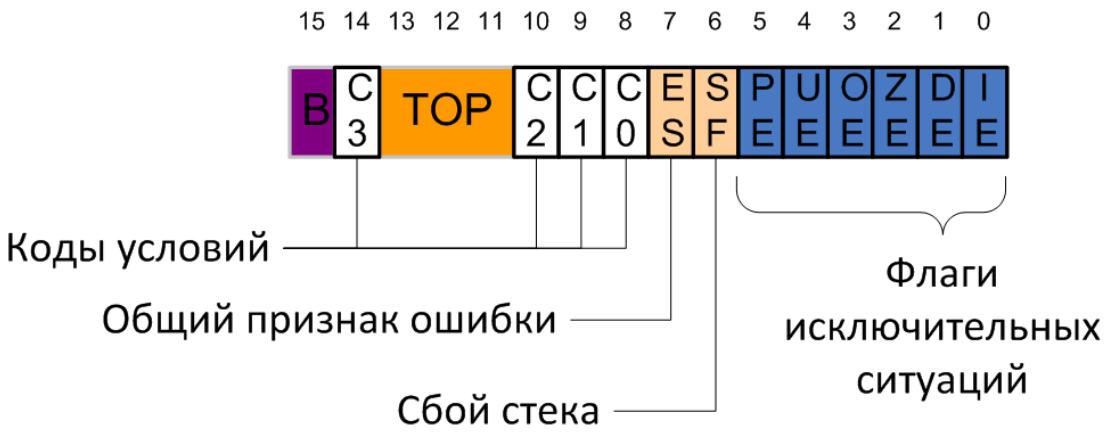


Инициализация x87 – инструкция FINIT

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

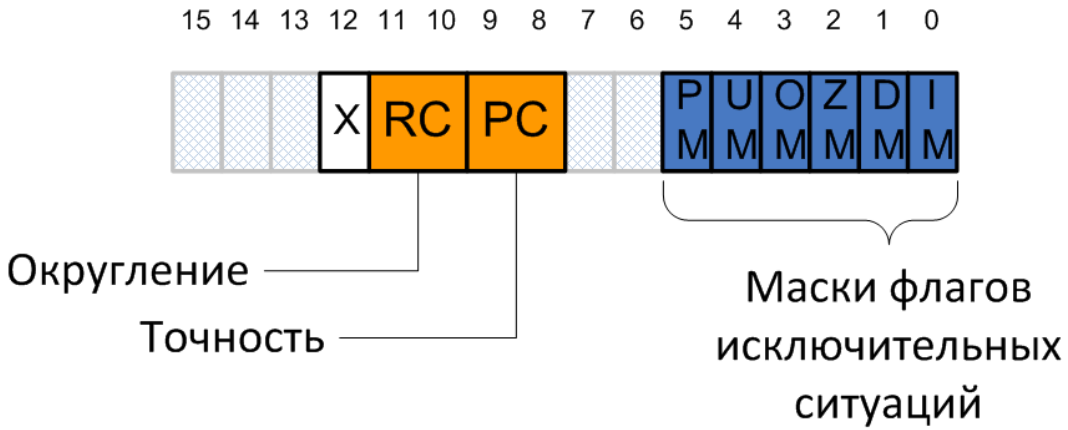
Слово (регистр) состояния

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



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



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

```
x

11.2 ~ 1.11999998092651367187 5E1

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:
    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:
    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

1.18999996185302734375E1 ~ 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
```

```
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
```

```
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
    ;…
```

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

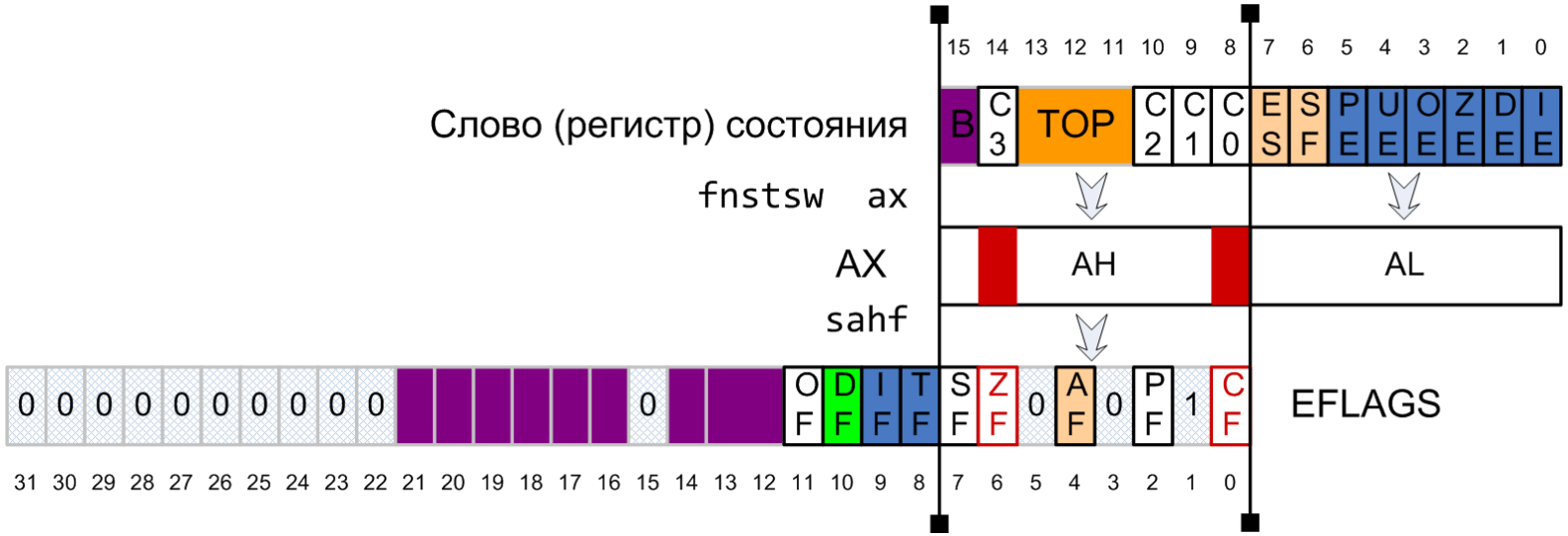
```
int isLe(double x, float y) {
   return x <= y;
}
```

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
        movzx   eax, al
        pop     ebp
        ret
```

Извлечение результатов сравнения

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



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
```