

Лекция 3

15 февраля

```

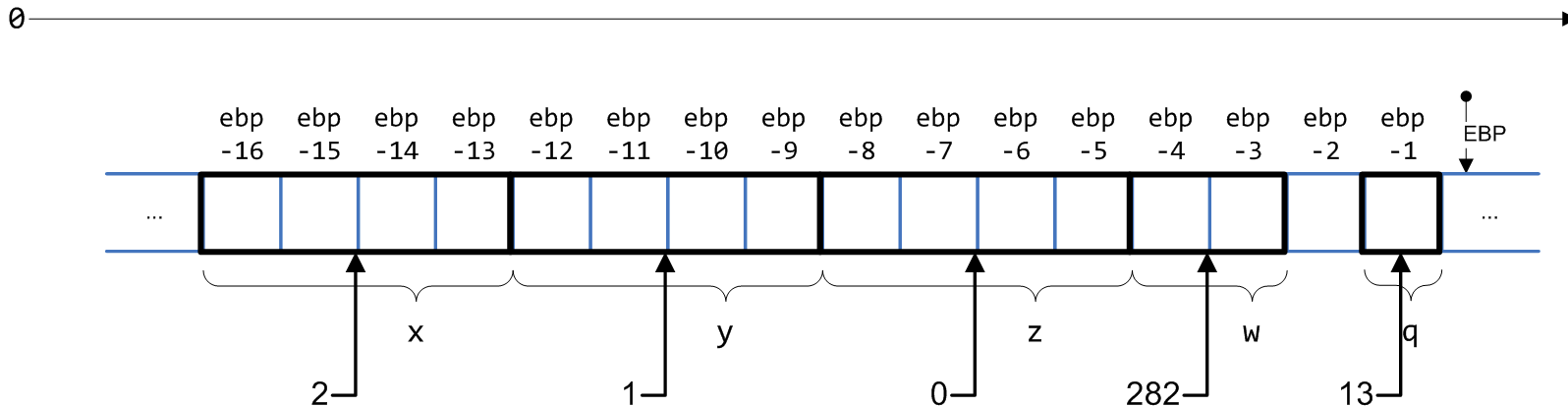
void f() {
    static int cntr = 0;    // 1
    int x = 2, y = 1, z = 0; // 2
    unsigned short w = 282; // 3
    signed char q = 13;    // 4
    ++cntr;                // 5
    z = -x + q * w * y - w; // 6
}

```

```

...
mov    dword [ebp-16], 2    ; (1)
mov    dword [ebp-12], 1    ; (2)
mov    dword [ebp-8], 0     ; (3)
mov    word  [ebp-4], 282   ; (4)
mov    byte  [ebp-1], 13    ; (5)
add    dword [cntr], 1     ; (6)
movsx  eax,  byte [ebp-1]   ; (7)
movzx  edx,  word [ebp-4]   ; (8)
imul   eax,  edx           ; (9)
imul   eax,  dword [ebp-12] ; (10)
sub    eax,  dword [ebp-16] ; (11)
sub    eax,  edx           ; (12)
mov    dword [ebp-8], eax   ; (13)
...

```



```
%include 'io.inc'
```

```
section .data
```

Секция статических инициализированных данных

```
    a    dw 1
```

```
    addr dd $
```

```
    var  dd 0x1234F00D
```

```
section .bss
```

Секция статических неинициализированных данных

```
    cntr resd 1
```

```
section .text
```

Секция кода

```
global CMAIN
```

```
CMAIN:
```

```
    add dword [cntr], 1
```

```
    mov eax, [addr]
```

```
    PRINT_HEX 4, eax
```

```
    NEWLINE
```

```
    PRINT_HEX 4, addr
```

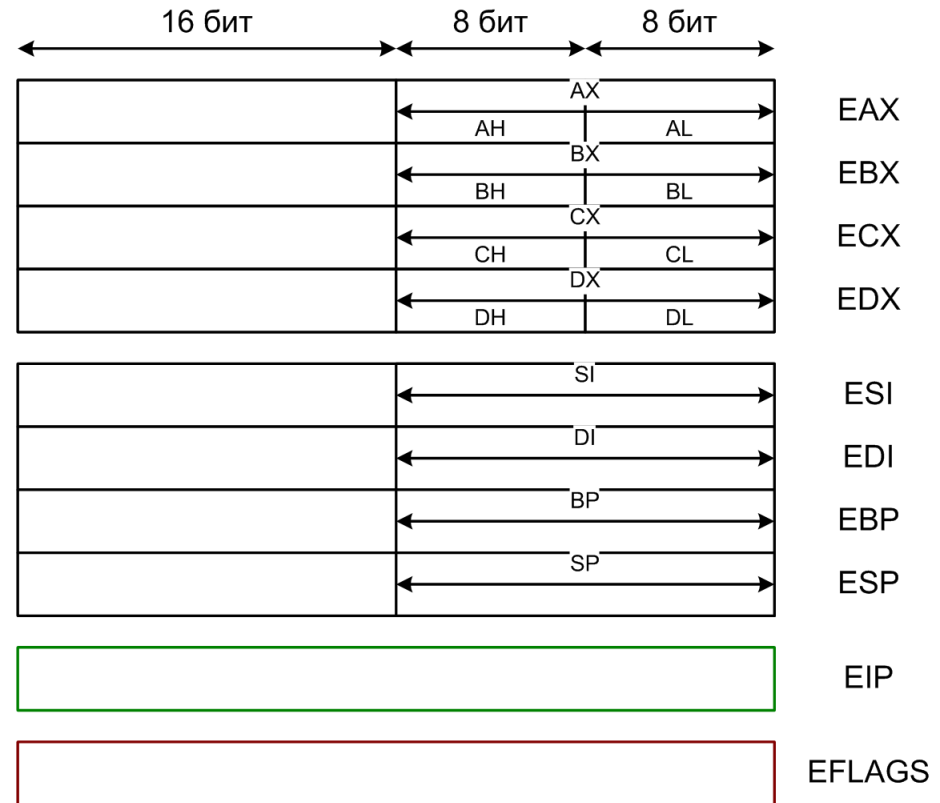
```
    NEWLINE
```

```
    xor eax, eax
```

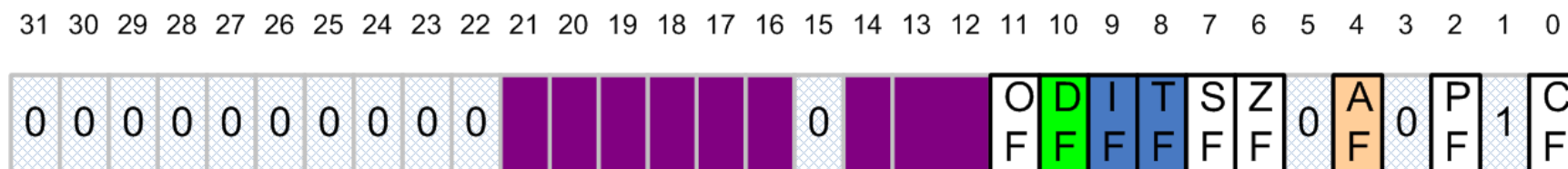
В тех случаях, когда размер других операндов позволяет определить размер операнда-памяти, допускается не указывать его спецификатор размера

Основные арифметические команды

- MOV
- MOVSX, MOVZX
- ADD, SUB
- NEG
 - r/m 8/16/32
- MUL
 - r/m 8/16/32
- IMUL
 - r/m 8/16/32
 - r 16/32, r/m 16/32
 - r 16/32, r/m 16/32, imm 16/32
- DIV, IDIV
 - r/m 8/16/32
- CBW, CWD, CDQ



Регистр EFLAGS



EFLAGS

