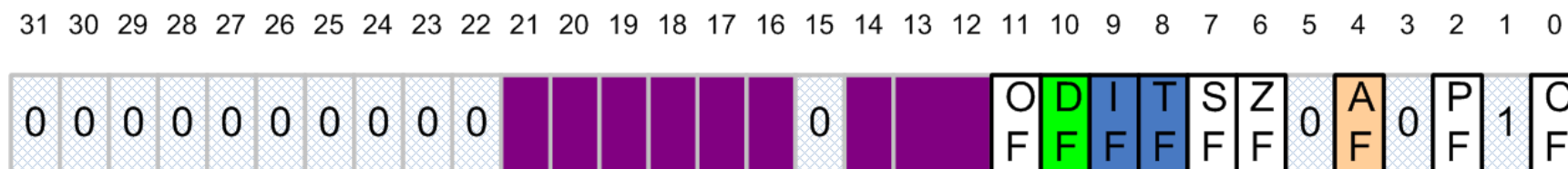


Лекция 4

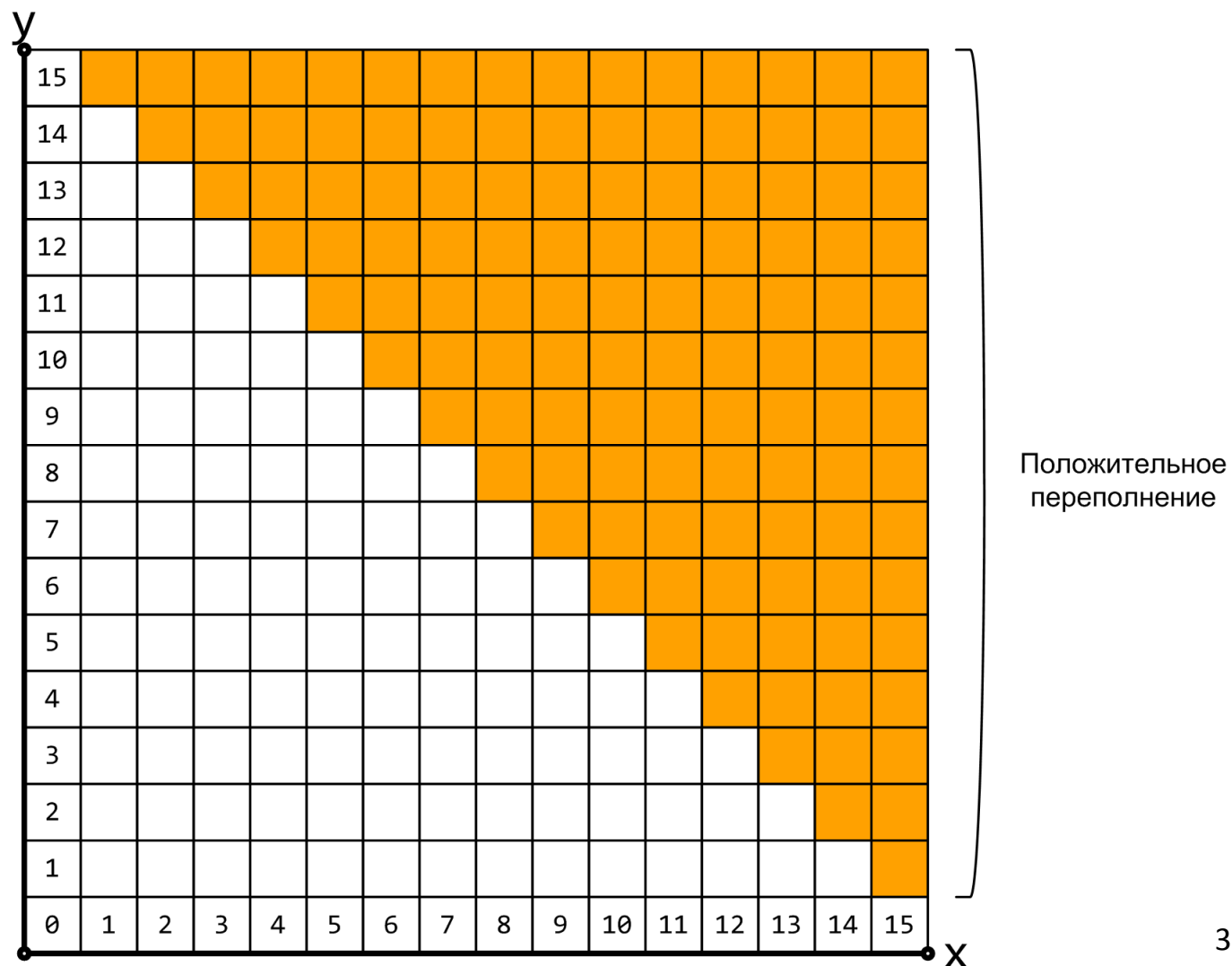
19 февраля

Регистр EFLAGS

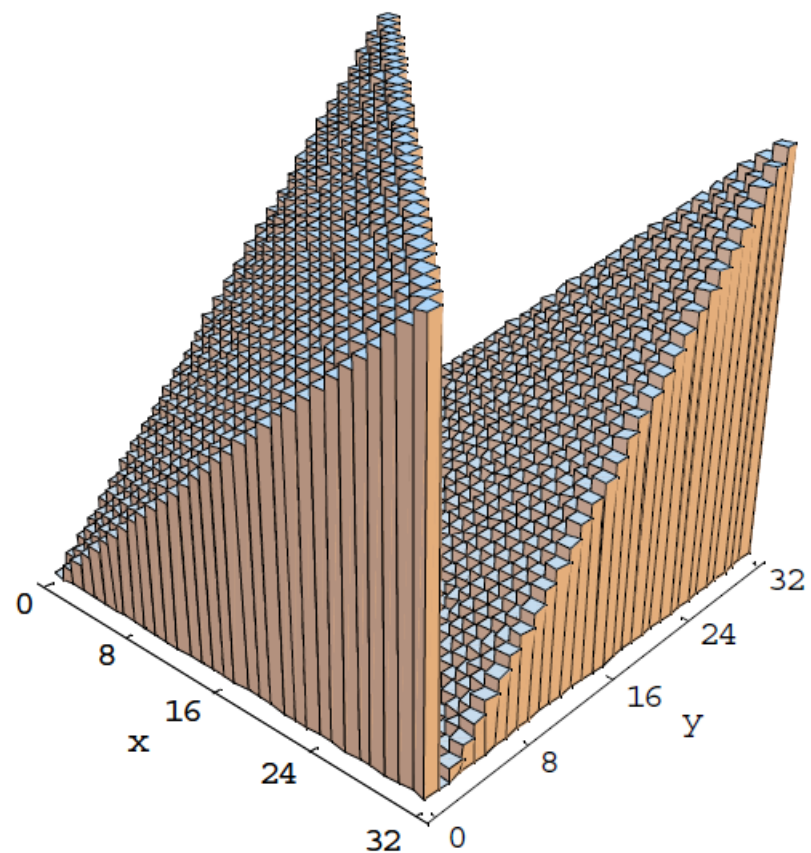
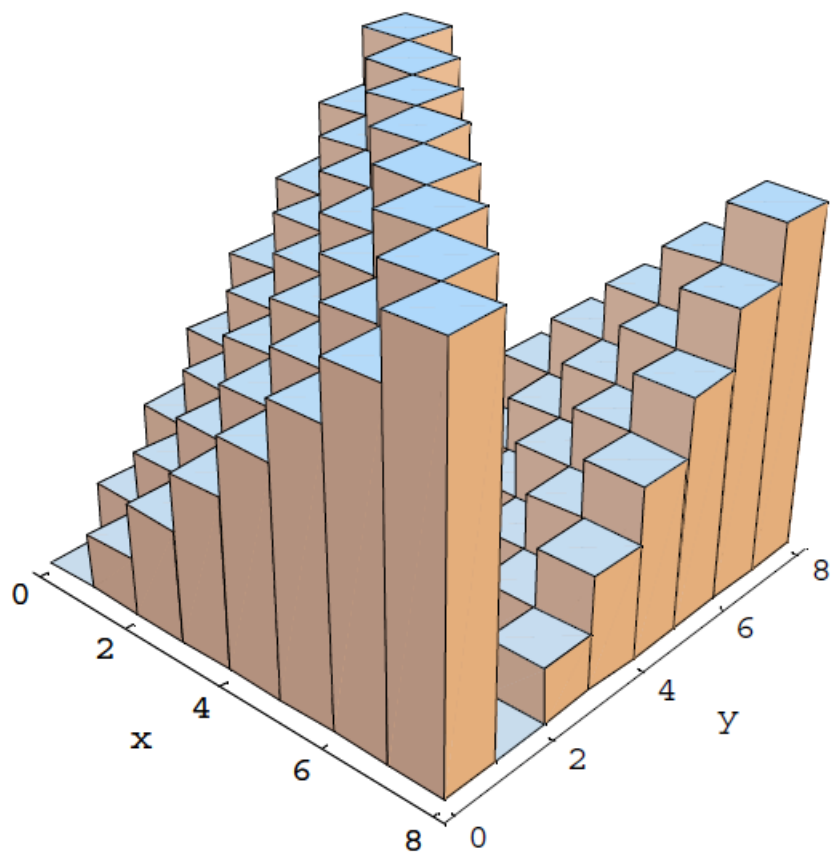


EFLAGS

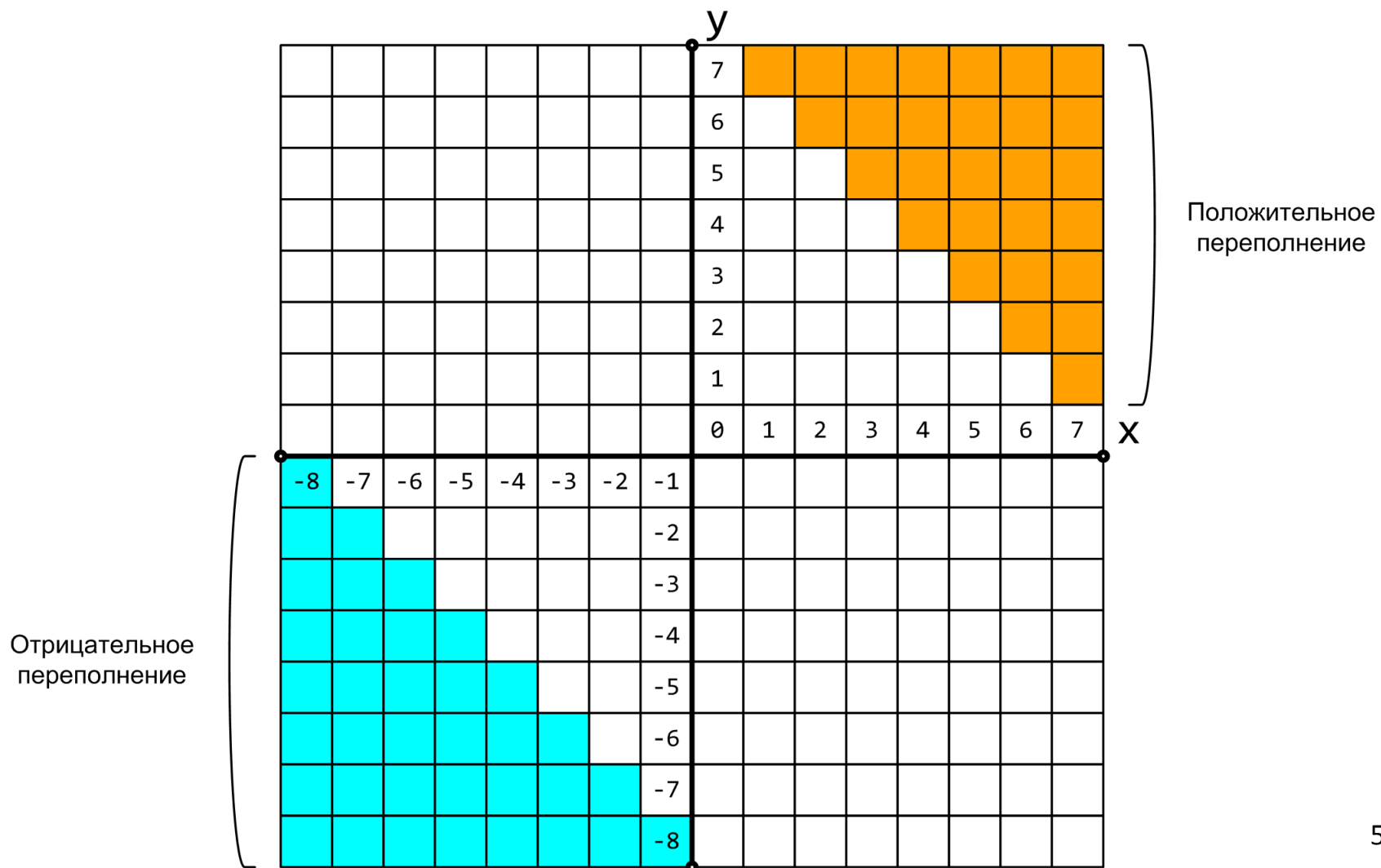
Переполнение при сложении двух беззнаковых чисел



Как «выглядит» переполнение в 3D?



Переполнение при сложении двух знаковых чисел



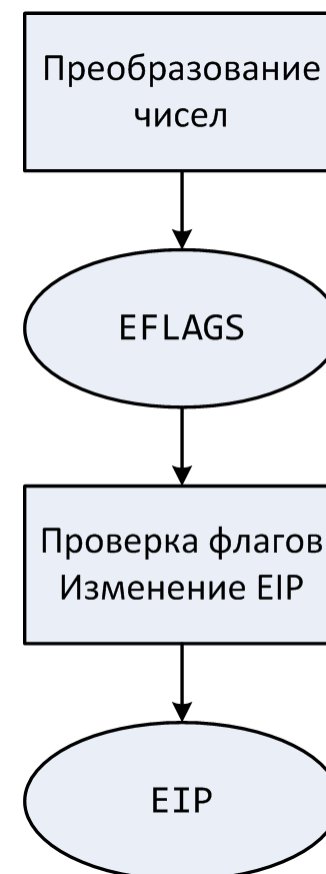
Естественный порядок выполнения

EIP	Машинный код	Длина	Ассемблерная инструкция
8048345	89 e5	2	mov ebp, esp
8048347	83 ec 10	3	sub esp, 0x10
804834a	c7 45 f0 02 00 00 00	7	mov dword [ebp-16],0x2

Прибавляем к значению регистра EIP длину в байтах декодированной команды

Изменение естественного порядка выполнения программы

- Арифметические операции
- CMP
 - r/m 8/16/32, i 8/16/32
 - r/m 8/16/32, r 8/16/32
 - r 8/16/32, r/m 8/16/32
- TEST
 - r/m 8/16/32, i 8/16/32
 - r/m 8/16/32, r 8/16/32
- Переходы
 - Способ адресации: задается абсолютный или относительный адрес передачи управления
 - Условия выполнения: безусловная или условная передача управления
- Безусловная передача управления
JMP
 - r/m/i 32
- Условная передача управления
Jcc
 - i 32



Абсолютная адресация (r/m 32)	Относительная адресация (i 32)
EIP ← ОП	EIP ← EIP + ОП

Регистр EFLAGS и инструкции

	OF	SF	ZF	PF	CF
ADD, SUB, NEG	M	M	M	M	M
INC, DEC	M	M	M	M	
IMUL, MUL	M	–	–	–	M
IDIV, DIV	–	–	–	–	–
CBW, CWD, CDQ					
MOV, MOVSX, MOVZX					
CMP	M	M	M	M	M
TEST	∅	M	M	M	∅

«M» инструкция обновляет флаг (сбрасывает или устанавливает)

«–» влияние инструкции на флаг не определено

« » инструкция на флаг не влияет

«∅» инструкция сбрасывает флаг


```
...  
SECTION .text  
GLOBAL CMAIN  
CMAIN:  
    MOV    EAX, DWORD [a]      ; (1)  
    TEST  EAX, EAX             ; (2)  
    JE    .1                   ; (3)  
    MOV    ECX, DWORD [b]      ; (4)  
    TEST  ECX, ECX            ; (5)  
    JE    .1                   ; (6)  
    CDQ                               ; (7)  
    IDIV  ECX                  ; (8)  
    SUB   DWORD [a], EDX       ; (9)  
.1:  
    XOR   EAX, EAX             ; (10)  
    RET                               ; (11)
```

Просмотр содержимого исполняемого файла

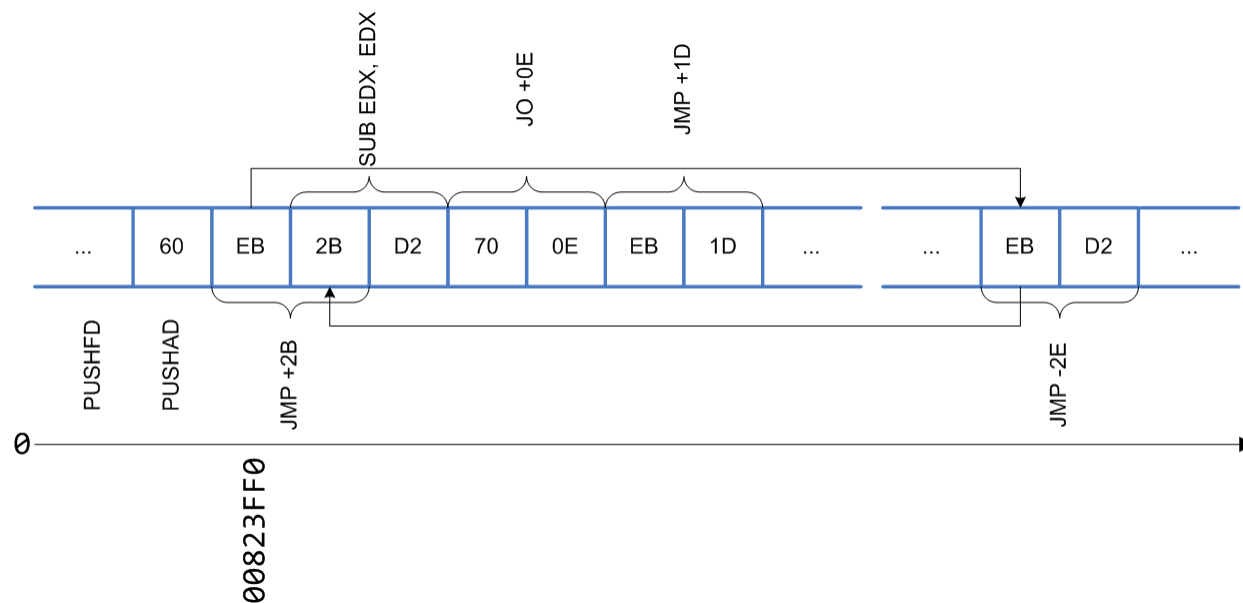
```
-bash-2.05b$ ./build_asm.sh backward.asm  
-bash-2.05b$ objdump -d -M intel backward
```

```
080483e0 <main>:  
80483e0:    a1 fc 94 04 08    mov     eax,ds:0x80494fc  
80483e5:    85 c0             test    eax,eax  
80483e7:    74 13            je     80483fc <main.1>  
80483e9:    8b 0d 00 95 04 08 mov     ecx,ds:0x8049500  
80483ef:    85 c9            test    ecx,ecx  
80483f1:    74 09            je     80483fc <main.1>  
80483f3:    99              cdq  
80483f4:    f7 f9            idiv   ecx  
80483f6:    29 15 fc 94 04 08 sub     ds:0x80494fc,edx  
  
080483fc <main.1>:  
80483fc:    31 c0            xor     eax,eax  
80483fe:    c3              ret  
80483ff:    90              nop
```

Противодействие дизассемблеру

Трасса выполнения

Адрес	Машинная команда	Ассемблерная инструкция
00823FEF	60	PUSHAD
00823FF0	EB2B	JMP 0082401Dh
0082401D	EBD2	JMP 00823FF1h
00823FF1	2BD2	SUB EDX, EDX
00823FF3	700E	JO 00824003h
00823FF5	EB1D	JMP 00824014h



Далее...

- Архитектура фон Неймана
- Архитектура IA-32
- Вычисление арифметических выражений
- Передача управления
- **Организация вызова функций**