

# Лекция 7

28 февраля

# Обратная задача

```
static int a, b, c, d;
```

```
???
```

mov	eax, dword [b] ; (1)
mov	edx, dword [c] ; (2)
or	al, -1 ; (3)
sal	eax, 3 ; (4)
add	edx, eax ; (5)
mov	dword [a], eax ; (6)
mov	eax, edx ; (7)
sar	edx, 31 ; (8)
idiv	dword [d] ; (9)
mov	dword [c], eax ; (10)

# Операции над отдельными битами

- Логические инструкции
  - AND
  - OR
  - XOR
  - NOT
- Сдвиги и вращения
  - SAR
  - SHR
  - SAL, SHL
  - ROR
  - ROL
  - RCR
  - RCL
- Битовые и байтовые
  - BT
  - BTS
  - BTR
  - BTC
  - BSF
  - BSR
  - SET*cc*
  - TEST

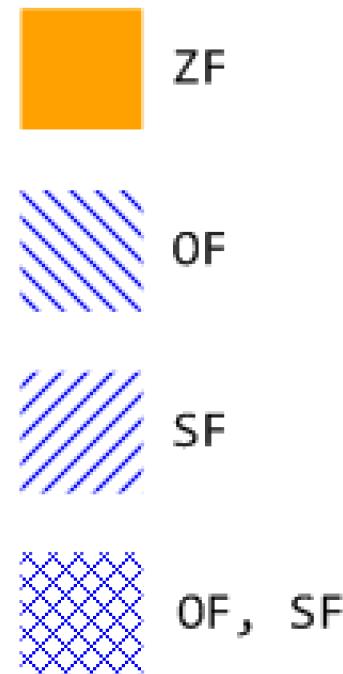
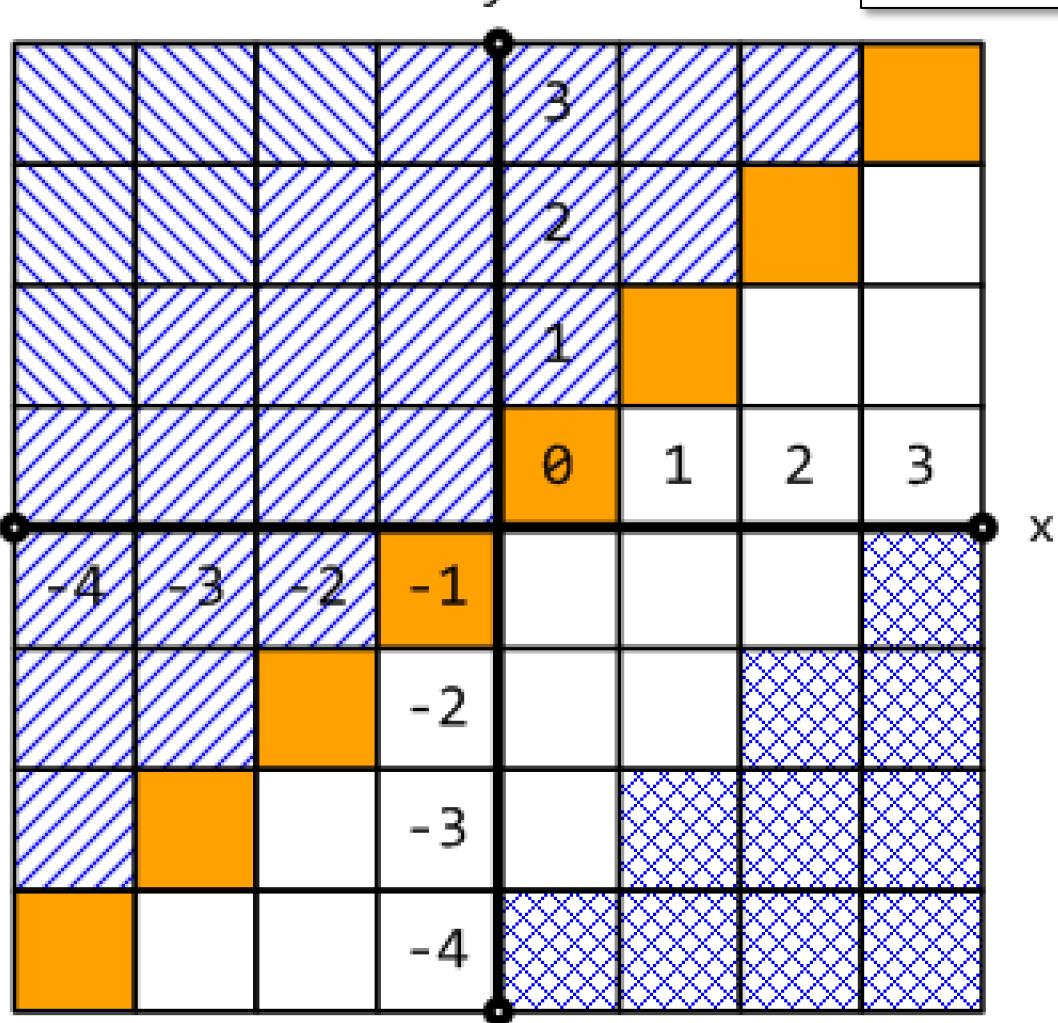
## Далее...

- Арифметика
- Операции над машинными словами
- Реализация управляющих операторов языка Си
  - Условная передача данных
  - Организация циклов
  - Оператор switch

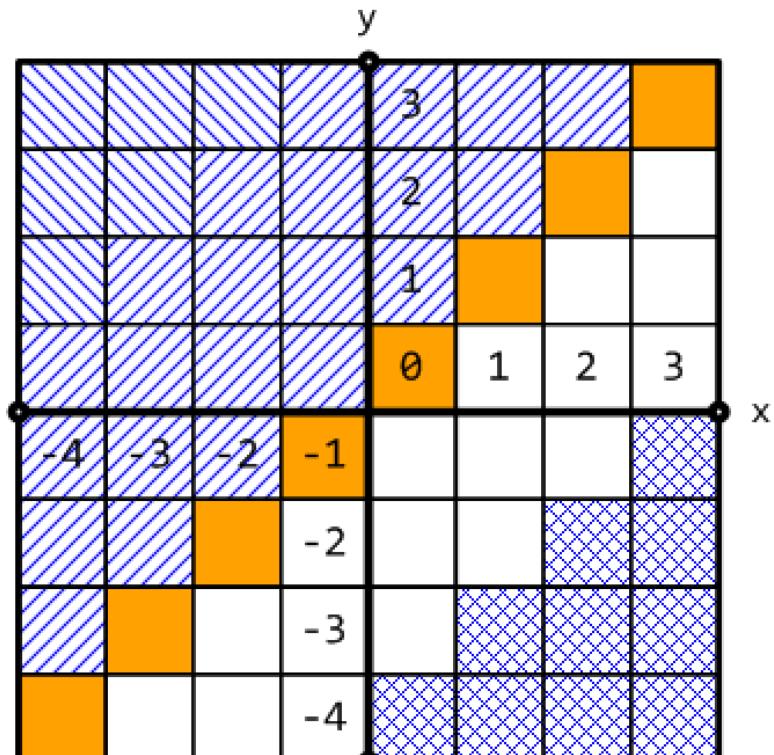
<b>Jcc</b>	<b>Условие</b>	<b>Описание</b>
JE	ZF	Равно / Ноль
JNE	$\sim ZF$	Не равно / Не ноль
JS	SF	Отрицательное число
JNS	$\sim SF$	Неотрицательное число
JG	$\sim(SF \wedge OF) \& \sim ZF$	Больше (знаковые числа)
JGE	$\sim(SF \wedge OF)$	Больше либо равно (знаковые числа)
JL	$(SF \wedge OF)$	Меньше (знаковые числа)
JLE	$(SF \wedge OF) \mid ZF$	Меньше либо равно (знаковые числа)
JA	$\sim CF \& \sim ZF$	Больше (числа без знака)
JB	CF	Меньше (числа без знака)

# Сравнение знаковых чисел

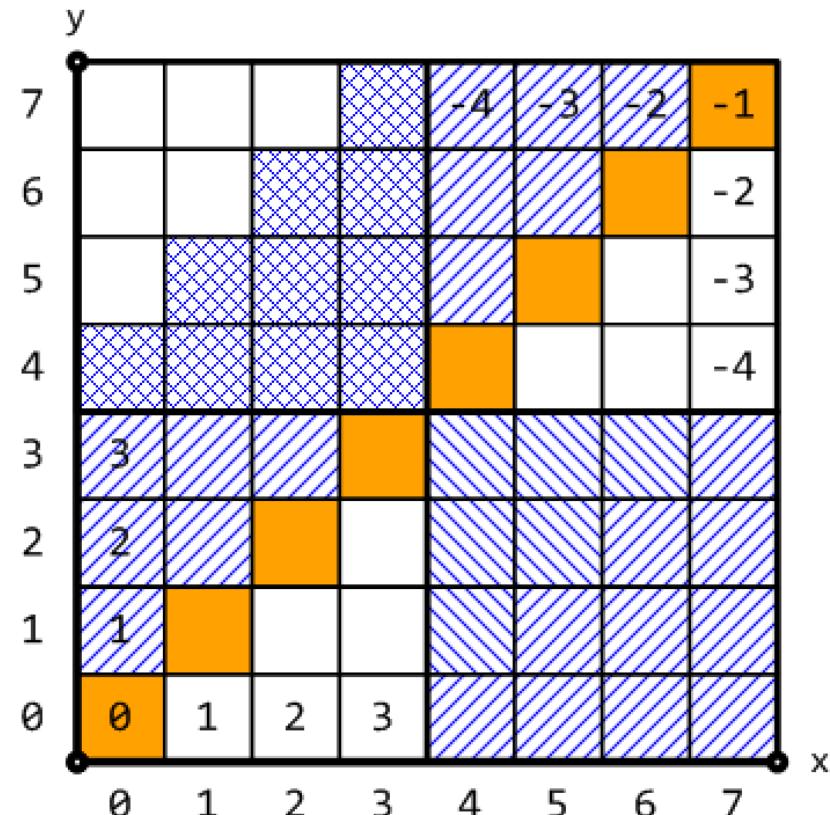
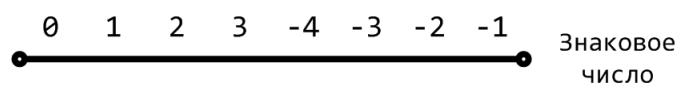
Переходим от  $(x \vee y)$  к  $(x-y)$



# Сравнение: со знаком и без



000	001	010	011	100	101	110	111
-----	-----	-----	-----	-----	-----	-----	-----



```
int absdiff(int x, int y) {  
    int result;  
    if (x > y) {  
        result = x-y;  
    } else {  
        result = y-x;  
    }  
    return result;  
}
```

```
absdiff:  
    push ebp  
    mov  ebp, esp  
    mov  edx, dword [8 + ebp] ; (1)  
    mov  eax, dword [12 + ebp] ; (2)  
    cmp  edx, eax ; (3)  
    jle  .L6 ; (4)  
    sub  edx, eax ; (5)  
    mov  eax, edx ; (6)  
    jmp  .L7 ; (7)  
.L6: ; (8)  
    sub  eax, edx ; (9)  
.L7: ; (10)  
    pop  ebp  
    ret
```

```
int goto_ad(int x, int y) {  
    int result;  
    if (x <= y) goto Else;  
    result = x-y;  
    goto Exit;  
Else:  
    result = y-x;  
Exit:  
    return result;  
}
```

```
absdiff:  
    push ebp  
    mov  ebp, esp  
    mov  edx, dword [8 + ebp] ; (1)  
    mov  eax, dword [12 + ebp] ; (2)  
    cmp  edx, eax ; (3)  
    jle  .L6 ; (4)  
    sub  edx, eax ; (5)  
    mov  eax, edx ; (6)  
    jmp  .L7 ; (7)  
.L6: ; (8)  
    sub  eax, edx ; (9)  
.L7: ; (10)  
    pop  ebp  
    ret
```

# Условная передача данных

```
val = Test ? Then_Expr : Else_Expr;
```

```
val = x>y ? x-y : y-x;
```



```
nt = !(Test);
if (nt) goto Else;
val = Then_Expr;
goto Done;
Else:
    val = Else_Expr;
Done:
    ...
```

```
tmp_val = Then_Expr;
val = Else_Expr;
t = Test;
if (t) val = tmp_val;
```



```
int absdiff(int x, int y) {
    int result;
    if (x > y) {
        result = x-y;
    } else {
        result = y-x;
    }
    return result;
}
```

```
int absdiff(int x, int y) {
    return (x > y)? x-y: y-x;
}
```

Более короткая запись ...

Регистр	Значение
edi	x
esi	y

absdiff:

```
...
mov    edx, edi
sub    edx, esi    ; tmp_val:edx = x-y
mov    eax, esi
sub    eax, edi    ; result:eax = y-x
cmp    edi, esi    ; Compare x:y
cmovg eax, edx    ; If >, result:eax = tmp_val:edx
...
...
```