© 2018 МГУ/ВМК/СП

Лекция 7

3 марта

© 2018 МГУ/ВМК/СП

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)
```

© 2018 МГУ/ВМК/СП

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**
  - **SET*cc***
  - **TEST**

© 2018 МГУ/ВМК/СП

Далее…

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

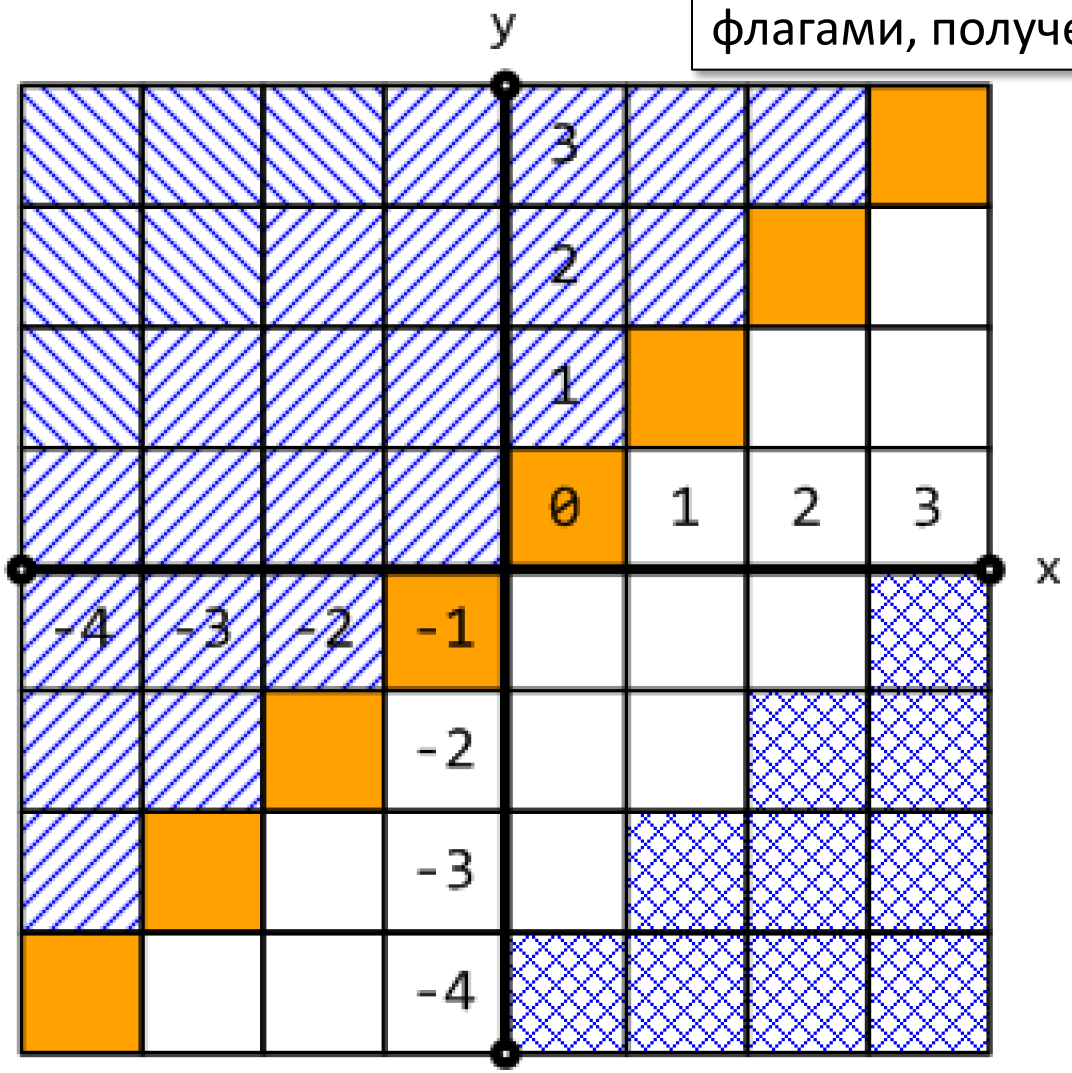
© 2018 МГУ/ВМК/СП

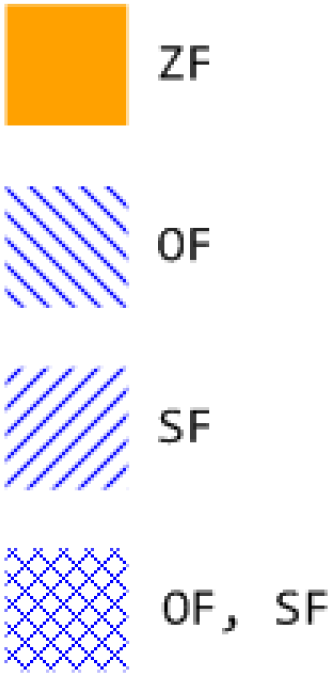
J*cc*	Условие	Описание
JE	ZF	Равно / Ноль
JNE	~ZF	Не равно / Не ноль
JS	SF	Отрицательное число
JNS	~SF	Неотрицательное число
JG	~(SF^OF)&~ZF	Больше (знаковые числа)
JGE	~(SF^OF)	Больше либо равно (знаковые числа)
JL	(SF^OF)	Меньше (знаковые числа)
JLE	(SF^OF)|ZF	Меньше либо равно (знаковые числа)
JA	~CF&~ZF	Больше (числа без знака)
JB	CF	Меньше (числа без знака)

© 2018 МГУ/ВМК/СП

Сравнение знаковых чисел

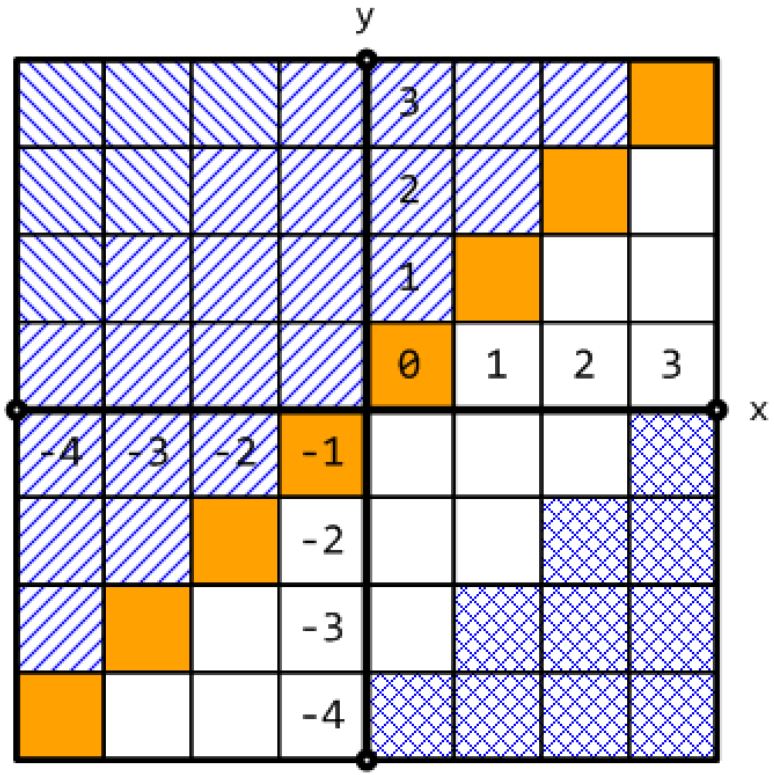
Сравнение x и y реализуется как формула над флагами, полученными после вычисления (x-y)

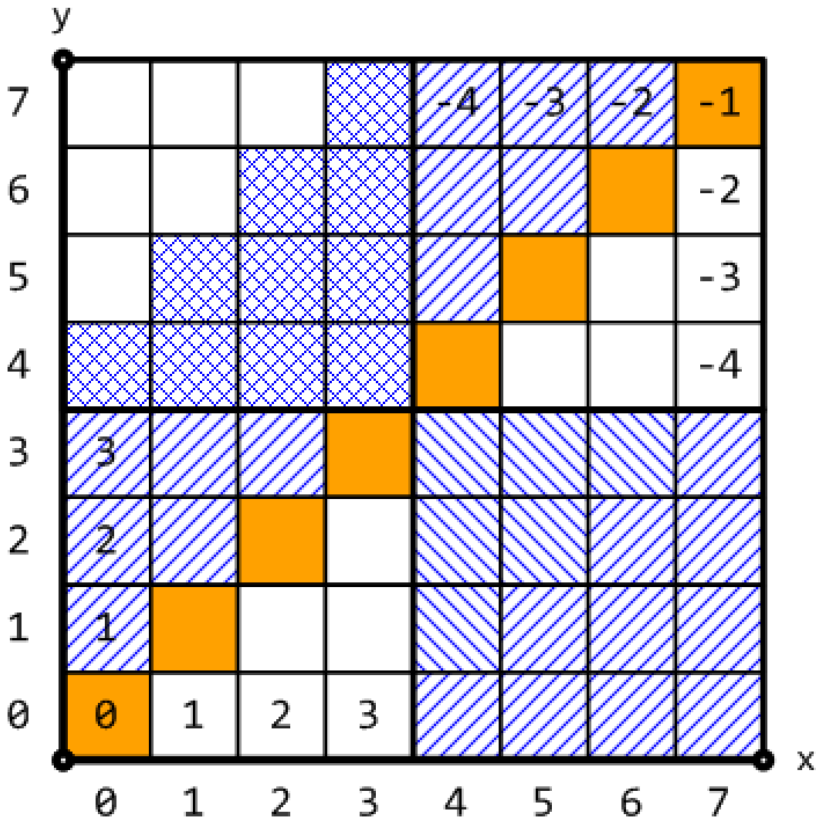




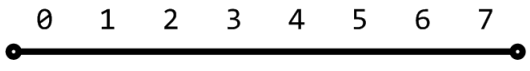
© 2018 МГУ/ВМК/СП

Сравнение: со знаком и без

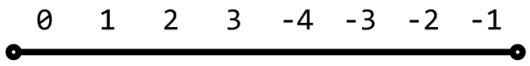




000	001	010	011	100	101	110	111



Беззнаковое число



Знаковое число

© 2018 МГУ/ВМК/СП

```
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
```

© 2018 МГУ/ВМК/СП

```
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
```

© 2018 МГУ/ВМК/СП

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;
```

© 2018 МГУ/ВМК/СП

```
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
    ...
```

© 2018 МГУ/ВМК/СП

Оператор do-while

```
int pcount_do(unsigned x) {
  int result = 0;
  do {
    result += x & 0x1;
    x >>= 1;
  } while (x);
  return result;
}
```

```
int pcount_do(unsigned x) {
  int result = 0;
loop:
  result += x & 0x1;
  x >>= 1;
  if (x)
    goto loop;
  return result;
}
```

© 2018 МГУ/ВМК/СП

Оператор do-while

Регистр	Значение
edx	**x**
ecx	**result**

```
int pcount_do(unsigned x) {
  int result = 0;
loop:
  result += x & 0x1;
  x >>= 1;
  if (x)
    goto loop;
  return result;
}
```

```
   mov ecx, 0     ;   result = 0
.L2:              ; loop:
   mov eax, edx
   and eax, 1     ;   t = x & 1
   add ecx, eax   ;   result += t
   shr edx, 1     ;   x >>= 1
   jne .L2        ;   If !0, goto loop
```

© 2018 МГУ/ВМК/СП

Оператор while

```
int pcount_while(unsigned x) {
  int result = 0;
  while (x) {
    result += x & 0x1;
    x >>= 1;
  }
  return result;
}
```

```
int pcount_do(unsigned x) {
  int result = 0;
  if (!x) goto done;
loop:
  result += x & 0x1;
  x >>= 1;
  if (x)
    goto loop;
done:
  return result;
}
```

```
int pcount_do(unsigned x) {
  int result = 0;
loop:
  if (!x) goto done;
  result += x & 0x1;
  x >>= 1;
  goto loop;
done:
  return result;
}
```