

Ассемблер Intel 8086

Вычисление выражений.

Выражения в ассемблере рассчитываются при трансляции, и в них могут использоваться только имена, значения которых известны транслятору.

В ассемблерных выражениях можно использовать четыре целочисленные арифметические операции - +, -, *, /, и четыре логические — and, or, not, xor.

Выражения, в которых участвуют значения переменных, реализуются с помощью соответствующей последовательности команд.

Ассемблер Intel 8086

Вычисление выражений. ($Y = ((a+b*c)^2+10)/3$)

```
;vav_p1
;Y = ((a+b*c)^2+10)/3

assume    cs:cod, ds:d, ss:s

s        segment    stack
        dw    128 dup (?)
s        ends

d        segment
;резервируем области под переменные a, b, c
        a            dw    ?
        b            dw    ?
        c            dw    ?
;размещаем строки - подсказки для ввода переменных
        msga         db    'Vvedite a:$'
        msgb         db    'Vvedite b:$'
        msgc         db    'Vvedite c:$'
        mmm          db    10,13,'$'
;описываем данные для процедур ввода и вывода целых чисел
        ...
d        ends
```

Ассемблер Intel 8086

Вычисление выражений. ($Y = ((a+b*c)^2+10)/3$)

```
cod      segment
;описываем процедуры ввода и вывода целых чисел
...
;устанавливаем ds на сегмент данных
start_s:  mov  ax,d
          mov  ds,ax
          ;вводим значение переменной a
          Mov  ah,09      ;выводим строку - подсказку для a
          Lea  dx,msga
          Int  21h
          Call IntegerIn ;вводим число
          Mov  a,ax      ;помещаем его в область a
          Mov  ah,09      ;переводим курсор на следующую строку
          Lea  dx,mmm
          Int  21h
          ;вводим значение переменной b
          Mov  ah,09      ;выводим строку - подсказку для b
          Lea  dx,msgb
          Int  21h
          Call IntegerIn ;вводим число
          Mov  b,ax      ;помещаем его в область b
          Mov  ah,09      ;переводим курсор на следующую строку
          Lea  dx,mmm
          Int  21h
```

Ассемблер Intel 8086

Вычисление выражений. ($Y = ((a+b*c)^2+10)/3$)

; вводим значение переменной с

```
Mov ah,09 ;выводим строку - подсказку для с
Lea dx,msgc
Int 21h
Call IntegerIn ;вводим число
Mov c,ax ;помещаем его в область с
Mov ah,09 ;переводим курсор на следующую строку
Lea dx,mmm
Int 21h
```

; рассчитываем значение выражения

```
mov ax,b ;умножаем b на c
mov bx,c
imul bx
add ax,a ;добавляем a
imul ax ;возводим в квадрат
add ax,10 ;увеличиваем на 10
mov bx,3 ;делим на 3
cwd
idiv bx
call IntegerOut ;выводим результат
;завершаем работу программы
mov ax,4c00h
int 21h
```

cod ends

end start s

```
C:\>tasm uav_p1
Turbo Assembler Version 1.0 Copyright (c) 1988

Assembling file: UAU_P1.ASM
*Warning* UAU_P1.ASM(119) Reserved word used as s
Error messages: None
Warning messages: 1
Remaining memory: 502k
```

```
C:\>tlink uav_p1
Turbo Link Version 5.1 Copyright (c) 1992 Borlan
```

```
C:\>uav_p1
Uedite a:1
Uedite b:1
Uedite c:1
4
C:\>_
```

Ассемблер Intel 8086

Вычисление выражений. ($Y = (a+b)/c - 3a^2/(b+c)$)

```
;vav_pl
;Y = ((a+b*c)^2+10)/3

assume  cs:cod, ds:d, ss:s

s      segment  stack
      dw  128 dup (?)
s      ends

d      segment
;резервируем области под переменные a, b, c
      a      dw  ?
      b      dw  ?
      c      dw  ?

;размещаем строки - подсказки для ввода переменных
      msga   db  'Vvedite a:$'
      msgb   db  'Vvedite b:$'
      msgc   db  'Vvedite c:$'
      mmm    db  10,13,'$'

;размещаем строки - сообщения об ошибках
      err1   db  'Деление на ноль (первая дробь)$'
      err2   db  'Деление на ноль (вторая дробь)$'

;описываем данные для процедур ввода и вывода целых чисел
      ...
d      ends
cod    segment
;описываем процедуры ввода и вывода целых чисел
      ...
```

Ассемблер Intel 8086

Вычисление выражений. ($Y = (a+b)/c - 3a^2/(b+c)$)

;устанавливаем ds на сегмент данных

```
start_s:  mov  ax,d
          mov  ds,ax
          ;вводим значение переменной a
Mov  ah,09  ;выводим строку - подсказку для a
Lea  dx,msga
Int  21h
Call  IntegerIn  ;вводим число
Mov  a,ax   ;помещаем его в область a
Mov  ah,09  ;переводим курсор на следующую строку
Lea  dx,mmm
Int  21h
;вводим значение переменной b
Mov  ah,09  ;выводим строку - подсказку для b
Lea  dx,msgb
Int  21h
Call  IntegerIn  ;вводим число
Mov  b,ax   ;помещаем его в область b
Mov  ah,09  ;переводим курсор на следующую строку
Lea  dx,mmm
Int  21h
;вводим значение переменной c
Mov  ah,09  ;выводим строку - подсказку для c
Lea  dx,msgc
Int  21h
Call  IntegerIn  ;вводим число
Mov  c,ax   ;помещаем его в область c
Mov  ah,09  ;переводим курсор на следующую строку
Lea  dx,mmm
Int  21h
```

Ассемблер Intel 8086

Вычисление выражений. ($Y = (a+b)/c - 3a^2/(b+c)$)

```
; проверяем знаменатели на равенство 0  
    Стр c, 0 ; проверяем первый знаменатель  
    Jnz m1 ; если не 0 - на m1  
    Mov ah, 09 ; выводим сообщение об ошибке  
    Lea dx, err1  
    Int 21h  
    Jmp err_p ; выходим  
m1: Mov ax, b ; считаем второй знаменатель  
    Add ax, c  
    Jnz m11 ; если не 0 - на m11  
    Mov ah, 09 ; выводим сообщение об ошибке  
    Lea dx, err2  
    Int 21h  
    Jmp err_p ; выходим  
; рассчитываем значение выражения
```

Ассемблер Intel 8086

Вычисление выражений. ($Y = (a+b)/c - 3a^2/(b+c)$)

```
mm1:    Mov    bx,ax    ; помещаем второй знаменатель в bx
        Mov    ax,a     ; считаем второй числитель
        Imul   ax
        Mov    cx,3
        Imul   cx
        Idiv   bx      ; считаем значение второй дроби
        Push  ax      ; сохраняем его в стеке
        Mov    ax,a     ; считаем первый числитель
        Add   ax,b
        Mov    bx,c     ; считаем значение первой дроби
        Cwd
        Idiv   bx
        Pop   bx      ; извлекаем из стека значение второй
; дроби
        Sub   ax,bx    ; вычитаем
        call  IntegerOut ; выводим результат
; завершаем работу программы
        mov   ax,4c00h ; с кодом завершения 0 - без ошибок
        int  21h
err_p:   mov   ax,4cfffh ; с кодом завершения 0fffh (-1) - с ошибкой
        int  21h
cod      ends
end      start_s
```

```
C:\>tasm uav_p2
Turbo Assembler Version 1.0 Copyright (c) 1988-1991
Assembling file:   UAV_P2.ASM
*Warning* UAV_P2.ASM(123) Reserved
Error messages:    None
Warning messages:  1
Remaining memory:  501k

C:\>tlink uav_p2
Turbo Link Version 5.1 Copyright (c) 1988-1991

C:\>uav_p2
Wedite a:1
Wedite b:1
Wedite c:1
1
C:\>
```

Ассемблер Intel 8086

Реализация многоразрядной арифметики. Сложение.

```
assume    cs:cod, ds:d, ss:s
s        segment    stack
    dw    128 dup (?)
s        ends
d        segment
;резервируем области под переменные
    X     dd    65538
    Y     dd    76543
d        ends
cod      segment
;устанавливаем ds на сегмент данных
start_s:    mov    ax,d
            mov    ds,ax

;выполняем сложение X+Y
    Mov    ax, word ptr X ; помещаем в ax младшее слово X
    Mov    dx, word ptr X+2 ; помещаем в dx старшее слово X
    Add    ax, word ptr Y ; добавляем к ax младшее слово Y
    Adc    dx, word ptr Y+2 ; добавляем к dx старшее слово Y

    mov    ax,4c00h ;с кодом завершения 0 - без ошибок
    int    21h
cod      ends
end      start_s
```

Ассемблер Intel 8086

Реализация многоразрядной арифметики. Сложение.

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TD
File Edit View Run Breakpoints Data Options Window Help
[ ]-CPU 80486
cs:0000 B8FD48 mov ax,48FD ax 0000 c=0
cs:0003 8ED8 mov ds,ax bx 0000 z=0
cs:0005 A10000 mov ax,[0000] cx 0000 s=0
cs:0008 8B160200 mov dx,[0002] dx 0000 o=0
cs:000C 03060400 add ax,[0004] si 0000 p=0
cs:0010 13160600 adc dx,[0006] di 0000 a=0
cs:0014 B8004C mov ax,4C00 bp 0000 i=1
cs:0017 CD21 int 21 sp 0100 d=0
cs:0019 0000 add [bx+si],al ds 48DD
cs:001B 0000 add [bx+si],al es 48DD
cs:001D 0000 add [bx+si],al ss 48FE
cs:001F 0000 add [bx+si],al cs 48FD
cs:0021 0000 add [bx+si],al ip 0021

ds:0000 CD 20 FF 9F 00 EA FF FF = f Ω
ds:0008 AD DE E4 01 C9 15 AE 01 i |P|S<@
ds:0010 C9 15 80 02 24 10 92 01 r|S<@|@
ds:0018 01 01 01 00 02 FF FF FF @@@ @

Alt: F2-Bkpt at F3-Close F4-Back F5-User F6-Undo F7-Ins
```

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TD
File Edit View Run Breakpoints Data Options Window Help
[ ]-CPU 80486
cs:0000 B8FD48 mov ax,48FD ax 2B01 c=0
cs:0003 8ED8 mov ds,ax bx F7CE z=0
cs:0005 A10000 mov ax,[0000] cx 7282 s=0
cs:0008 8B160200 mov dx,[0002] dx 0002 o=0
cs:000C 03060400 add ax,[0004] si 03B0 p=0
cs:0010 13160600 adc dx,[0006] di 15C9 a=0
cs:0014 B8004C mov ax,4C00 bp 0100 i=1
cs:0017 CD21 int 21 sp 0100 d=0
cs:0019 0000 add [bx+si],al ds 48FD
cs:001B 0000 add [bx+si],al es 48DD
cs:001D 0000 add [bx+si],al ss 48FE
cs:001F 0000 add [bx+si],al cs 48FE
cs:0021 0000 add [bx+si],al ip 0014

es:0000 CD 20 FF 9F 00 EA FF FF = f Ω
es:0008 AD DE E4 01 C9 15 AE 01 i |P|S<@
es:0010 C9 15 80 02 24 10 92 01 r|S<@|@
es:0018 01 01 01 00 02 FF FF FF @@@ @

ss:0102 0001
ss:0100 0002

Alt: F2-Bkpt at F3-Close F4-Back F5-User F6-Undo F7-Instr F8-Rtn
```

$$65538 + 76543 = 142081 = 22B01_{16}$$

Ассемблер Intel 8086

Реализация многоразрядной арифметики. Вычитание.

```
assume    cs:cod, ds:d, ss:s
s        segment    stack
    dw    128 dup (?)
s        ends
d        segment
;резервируем области под переменные
    X     dd    65538
    Y     dd    76543
d        ends
cod      segment
;устанавливаем ds на сегмент данных
start_s:    mov    ax,d
            mov    ds,ax

;выполняем вычитание X-Y
    Mov    ax, word ptr X ; помещаем в ax младшее слово X
    Mov    dx, word ptr X+2 ; помещаем в dx старшее слово X
    Sub    ax, word ptr Y ; вычитаем из ax младшее слово Y
    Sbb    dx, word ptr Y+2 ; вычитаем из dx старшее слово Y

    mov    ax,4c00h ;с кодом завершения 0 - без ошибок
    int    21h
cod      ends
end    start_s
```

Ассемблер Intel 8086

Реализация многоразрядной арифметики. Вычитание.

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TD
File Edit View Run Breakpoints Data Options Window Help READY
[ ]-CPU 80486-
cs:0000 B8FD48 mov ax,48FD ax 0000 c=0
cs:0003 8ED8 mov ds,ax bx 0000 z=0
cs:0005 A10000 mov ax,[0000] cx 0000 s=0
cs:0008 8B160200 mov dx,[0002] dx 0000 o=0
cs:000C 2B060400 sub ax,[0004] si 0000 p=0
cs:0010 1B160600 sbb dx,[0006] di 0000 a=0
cs:0014 B8004C mov ax,4C00 bp 0000 i=1
cs:0017 CD21 int 21 sp 0100 d=0
cs:0019 0000 add [bx+sil,al] ds 48DD
cs:001B 0000 add [bx+sil,al] es 48D
cs:001D 0000 add [bx+sil,al] ss 48E
cs:001F 0000 add [bx+sil,al] cs 48F
cs:0021 0000 add [bx+sil,al] ip 000

ds:0000 CD 20 FF 9F 00 EA FF FF = f 0
ds:0008 AD DE E4 01 C9 15 AE 01 i | 0 0 0 0
ds:0010 C9 15 80 02 24 10 92 01 | 0 0 0 0
ds:0018 01 01 01 00 02 FF FF FF 000 0

ss:010
ss:010
```

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TD
File Edit View Run Breakpoints Data Options Window Help
[ ]-CPU 80486-
cs:0000 B8FD48 mov ax,48FD ax D503 c=1
cs:0003 8ED8 mov ds,ax bx 0000 z=0
cs:0005 A10000 mov ax,[0000] cx 0000 s=1
cs:0008 8B160200 mov dx,[0002] dx FFFF o=0
cs:000C 2B060400 sub ax,[0004] si 0000 p=1
cs:0010 1B160600 sbb dx,[0006] di 0000 a=1
cs:0014 B8004C mov ax,4C00 bp 0000 i=1
cs:0017 CD21 int 21 sp 0100 d=0
cs:0019 0000 add [bx+sil,al] ds 48FD
cs:001B 0000 add [bx+sil,al] es 48DD
cs:001D 0000 add [bx+sil,al] ss 48ED
cs:001F 0000 add [bx+sil,al] cs 48FE
cs:0021 0000 add [bx+sil,al] ip 0014

es:0000 CD 20 FF 9F 00 EA FF FF = f 0
es:0008 AD DE E4 01 C9 15 AE 01 i | 0 0 0 0
es:0010 C9 15 80 02 24 10 92 01 | 0 0 0 0
es:0018 01 01 01 00 02 FF FF FF 000 0

ss:0102 0001
ss:0100 0002
```

$$65538 - 76543 = -11005 = \text{FFFFD503}_{16}$$

Ассемблер Intel 8086

Реализация многоразрядной арифметики. Умножение.

```
;ma_p3
;X*Y

assume    cs:cod, ds:d, ss:s

s        segment    stack
        dw    128 dup (?)
s        ends

d        segment
;резервируем области под переменные
        X        dd    65538
        Y        dd    76543
        tmp1     dd    0
        tmp2     dd    0
        tmp3     dd    0
d        ends

cod      segment

;устанавливаем ds на сегмент данных
start_s:    mov    ax,d
            mov    ds,ax
```

Ассемблер Intel 8086

Реализация многоразрядной арифметики. Умножение.

```
; выполняем умножение X*Y
Mov  ax, word ptr X+2
Mov  bx, word ptr Y+2
Mul  bx
Mov  word ptr tmp1, ax
Mov  word ptr tmp1+2, dx
Mov  ax, word ptr X+2
Mov  bx, word ptr Y
Mul  bx
Mov  word ptr tmp2, ax
Mov  word ptr tmp2+2, dx
Mov  ax, word ptr X
Mov  bx, word ptr Y+2
Mul  bx
Mov  word ptr tmp3, ax
Mov  word ptr tmp3+2, dx
Mov  ax, word ptr X
Mov  bx, word ptr Y
Mul  bx
Xor  bx, bx
Xor  cx, cx
Add  dx, word ptr tmp3
Adc  cx, word ptr tmp3+2
Add  dx, word ptr tmp2
```

Ассемблер Intel 8086

Реализация многоразрядной арифметики. Умножение.

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TD
File Edit View Run Breakpoints Data Options Window Help
[ ]=CPU 80486
cs:0000 B8FD48 mov ax,48FD ax 0000 c=0
cs:0003 8ED8 mov ds,ax bx 0000 z=0
cs:0005 A10200 mov ax,[0002] cx 0000 s=0
cs:0008 8B1E0600 mov bx,[0006] dx 0000 o=0
cs:000C F7E3 mul bx si 0000 p=0
cs:000E A30800 mov [0008],ax di 0000 a=0
cs:0011 89160A00 mov [000A],dx bp 0000 i=1
cs:0015 A10200 mov ax,[0002] sp 0100 d=0
cs:0018 8B1E0400 mov bx,[0004] ds 48DD
cs:001C F7E3 mul bx es 48D
cs:001E A30C00 mov [000C],ax ss 48E
cs:0021 89161200 mov [0012],dx cs 48F
cs:0025 A10000 mov ax,[0000] ip 000
ds:0000 CD 20 FF 9F 00 EA FF FF = f Ω
ds:0008 AD DE E4 01 C9 15 AE 01 i |z|s|<|
ds:0010 C9 15 80 02 24 10 92 01 |s|c|s|>|f|
ds:0018 01 01 01 00 02 FF FF FF |e|e|e|e|
ss:010
ss:010
Alt: F2-Bkpt at F3-Close F4-Back F5-User F6-Undo F7-Ins
```

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TD
File Edit View Run Breakpoints Data Options Window Help
[ ]=CPU 80486
cs:004A 03160C00 add dx,[000C] ax 55FE c=0
cs:004E 130E0E00 adc cx,[000E] bx 0000 z=1
cs:0052 83D300 adc bx,0000 cx 0001 s=0
cs:0055 030E0800 add cx,[0008] dx 2B01 o=0
cs:0059 131E0A00 adc bx,[000A] si 0000 p=1
cs:005D B8004C mov ax,4C00 di 0000 a=0
cs:0060 CD21 int 21 bp 0000 i=1
cs:0062 0000 add [bx+si],al sp 0100 d=0
cs:0064 0000 add [bx+si],al ds 48FD
cs:0066 0000 add [bx+si],al es 48DD
cs:0068 0000 add [bx+si],al ss 48ED
cs:006A 0000 add [bx+si],al cs 48FF
cs:006C 0000 add [bx+si],al ip 005D
es:0000 CD 20 FF 9F 00 EA FF FF = f Ω
es:0008 AD DE E4 01 C9 15 AE 01 i |z|s|<|
es:0010 C9 15 80 02 24 10 92 01 |s|c|s|>|f|
es:0018 01 01 01 00 02 FF FF FF |e|e|e|e|
ss:0102 0001
ss:0100 0002
Alt: F2-Bkpt at F3-Close F4-Back F5-User F6-Undo F7-Instr F8-Rtn
```

$$65538 * 76543 = 5016475134 = 12B0155FE_{16}$$