

# **Лабораторная работа №2**

**"Примеры форматов чисел представленных в цифровых процессорах"**

Программа на языке ассемблера для этой лабораторной работы содержит пустое "тело" вычисления, т.е. только загрузку регистра DS и выход в DosDox.

сегмент данных

сегмент кода

```
lek1 - Блокнот
Файл  Правка  Формат  Вид  Справка
data segment
;=====
;
;      A dw 19h
;
data ends
;=====
code segment
assume cs:code, ds:data, ss:nothing
start:
    mov ax, data ;load address
    mov ds, ax ; data segment
;=====
; "тело" вычисления ПУСТО
;=====
quit:
    mov ax, 4c00h ; cod to finish 0
    int 21h ; exit to dos
code ends
end start
```

**=19<sub>(16)</sub>**

**ПУСТО**

# Целые числа

(на ассемблере для 16-разрядного процессора)

$$A = 25_{(10)} = 19_{(16)}$$

h=hex=(16)

сегмент данных

```
data segment
;
A dw 19h
data ends
;
code segment
assume cs:code, ds:data, ss:nothing
start:
    mov ax, data ;load adress
    mov ds, ax ; data segment
;
; "тело" вычисления пусто
;
quit:
    mov ax, 4c00h ; cod to finish 0
    int 21h ; exit to dos
code ends
end start
```

сегмент кода

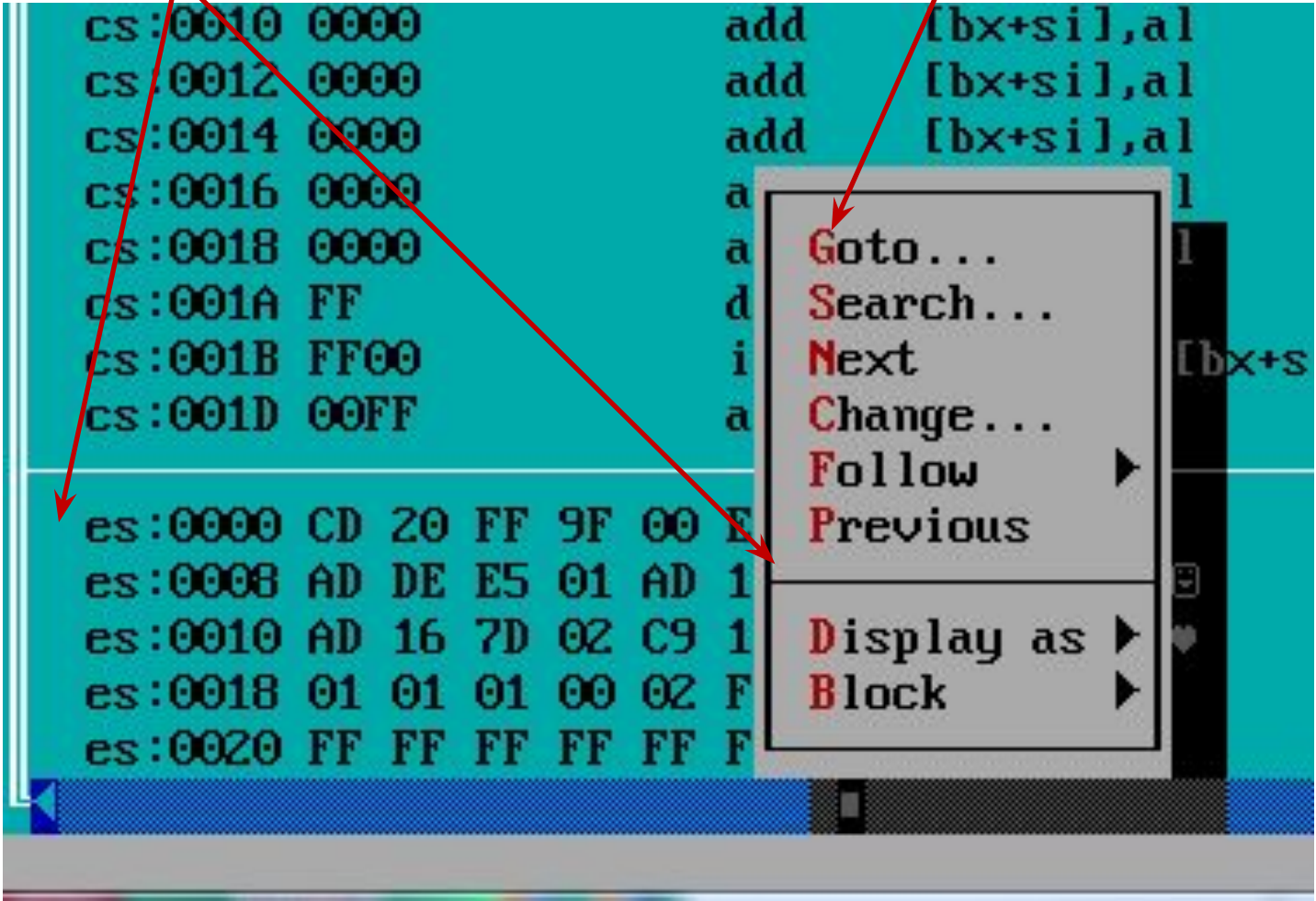
=19<sub>(16)</sub>

Чтобы отобразить сегмент данных нужно по клавише F8 выполнить первые две команды программы, чтобы загрузить регистр DS:

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:
File View Run Breakpoints Data Option
[ ] CPU 80486
cs:0000 B85A46 mov ax,465A
cs:0003 8ED8 mov ds,ax
cs:0005 B8004C mov ax,4C00
cs:0008 CD21 int 21
cs:000A 0000 add [bx+si],al
cs:000C 0000 add [bx+si],al
cs:000E 0000 add [bx+si],al
cs:0010 0000 add [bx+si],al
cs:0012 0000 add [bx+si],al
cs:0014 0000 add [bx+si],al
cs:0016 0000 add [bx+si],al
cs:0018 0000 add [bx+si],al
cs:001A FF db FF
cs:001B FF00 inc word ptr [bx+si]
cs:001D 00FF add bh,bh
ds:0000 CD 20 FF 9F 00 EA FF FF = f Ω
ds:0008 AD DE E5 01 AD 16 AF 01 ; |r i - >
ds:0010 AD 16 7D 02 C9 10 3F 03 i - } 0 p ? ♡
ds:0018 01 01 01 00 02 FF FF FF 000 0
ds:0020 FF FF FF FF FF FF FF FF
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next
```

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:
File View Run Breakpoints Data Option
[ ] CPU 80486
cs:0000 B85A46 mov ax,465A
cs:0003 8ED8 mov ds,ax
cs:0005 B8004C mov ax,4C00
cs:0008 CD21 int 21
cs:000A 0000 add [bx+si],al
cs:000C 0000 add [bx+si],al
cs:000E 0000 add [bx+si],al
cs:0010 0000 add [bx+si],al
cs:0012 0000 add [bx+si],al
cs:0014 0000 add [bx+si],al
cs:0016 0000 add [bx+si],al
cs:0018 0000 add [bx+si],al
cs:001A FF db FF
cs:001B FF00 inc word ptr [bx+si]
cs:001D 00FF add bh,bh
es:0000 CD 20 FF 9F 00 EA FF FF = f Ω
es:0008 AD DE E5 01 AD 16 AF 01 ; |r i - >
es:0010 AD 16 7D 02 C9 10 3F 03 i - } 0 p ? ♡
es:0018 01 01 01 00 02 FF FF FF 000 0
es:0020 FF FF FF FF FF FF FF FF
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next
```

Переключиться на окно данных (окно ОП), щелкнув на нем правой кнопкой.левой кнопкой, вызвать контекстное меню. Выбрать "Goto" нажать Enter:



Задать отображение памяти начиная с адреса 0000, нажать Enter:

The screenshot shows a debugger window with a teal background. The main area displays memory addresses and assembly instructions. A dialog box is overlaid on the screen, prompting for an address. A red arrow points from the text above to the dialog box. The dialog box title is "[ ]=Enter address to position to" and the input field contains "ds:0\_". There are "OK", "Cancel", and "Help" buttons. A status bar at the bottom reads "Enter item prompted for in dialog title".

Address	Hex	Assembly	Segment	Value
cs:0014	0000	add [bx+si],al	es	4041
cs:0016	0000	add [bx+si],al	ss	4659
cs:0018	0000	add [bx+si],al	cs	4651
cs:001A	FF	db		0000
cs:001B	FF00	inc		0000
cs:001D	00FF	add		0000
es:0000	CD 20 FF 9F 00 EA			0000
es:0008	AD DE E5 01 AD 16			0000
es:0010	AD 16 7D 02 C9 10			0000
es:0018	01 01 01 00 02 FF FF		ss:0000	0000
es:0020	FF FF FF FF FF FF		ss:0000	0000

Должно быть в 16-разрядном процессоре по адресу ds:0000 целое число =0019<sub>(16)</sub>

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TD

File View Run Breakpoints Data Options Window Help

[CPU 80486]

cs:0000 B85A46	mov	ax,465A	ax	465A	c=0
cs:0003 BED8	mov	ds,ax	bx	0000	z=0
cs:0005 B8004C	mov	ax,4C00	cx	0000	s=0
cs:0008 CD21	int	21	dx	0000	o=0
cs:000A 0000	add	[bx+sil],al	si	0000	p=0
cs:000C 0000	add	[bx+sil],al	di	0000	a=0
cs:000E 0000	add	[bx+sil],al	bp	0000	i=1
cs:0010 0000	add	[bx+sil],al	sp	0000	d=0
cs:0012 0000	add	[bx+sil],al	ds	465A	
cs:0014 0000	add	[bx+sil],al	es	464A	
cs:0016 0000	add	[bx+sil],al	ss	4659	
cs:0018 0000	add	[bx+sil],al	cs	465B	
cs:001A FF	db	FF	ip	0005	
cs:001B FF00	inc	word ptr [bx+sil]			
cs:001D 00FF	add	bh,bh			

ds:0000 19 00 00 00 00 00 00 00 ↓

ds:0008 00 00 00 00 00 00 00 00

ds:0010 B8 5A 46 8E D8 B8 00 4C ↵ ZF A L

ds:0018 CD 21 00 00 00 00 00 00 =!

ds:0020 00 00 00 00 00 00 00 00

ss:0008 6864

ss:0006 742E

ss:0004 706C

ss:0002 6568

ss:0000 6474

ОП  
Сегмент данных

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

# Побайтовое отображение памяти с ds:0

CS:001D	00FF	add								Bh, Bh
ds:0000	19	00	99	02	52	34	29	00	↓ 00R4)	
ds:0008	00	00	00	00	00	00	00	00		
ds:0010	BB	5A	46	8E	D8	BB	00	4C	↑ ZFÄ† L	
ds:0018	CD	21	00	00	00	00	00	00	=!	
ds:0020	00	00	00	00	00	00	00	00		

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next

адреса  
памяти

содержимое  
по адресам памяти

СИМВОЛЬНЫЙ  
код  
байтов



# 16-разрядный процессор

Целые числа =  $19_{(16)}$  и  $299_{(16)}$

```
LEK1 - Блокнот
Файл  Правка  Формат  Вид  Справка
data segment
=====
        A dw 19h
        B dw 299h
data ends
=====
code segment
assume cs:code, ds:data, ss:nothing
start:  mov ax, data ;load adress
        mov ds, ax ; data segment
        =====
        ; "тело" ВЫЧИСЛЕНИЯ
        =====
quit:   mov ax, 4c00h ; cod to finish 0
        int 21h ; exit to dos
code ends
end start
```

сегмент данных

сегмент кода

# 16-разрядный процессор

Целые числа  $0019_{(16)}$  и  $0229_{(16)}$  в ОП

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip: 0, Program: TD
File View Run Breakpoints Data Options
[ ]=CPU 80486
cs:0000 B85A46 mov ax,465A
cs:0003 8ED8 mov ds,ax
cs:0005 B8004C mov ax,4C00
cs:0008 CD21 int 21
cs:000A 0000 add [bx+si],al
cs:000C 0000 add [bx+si],al
cs:000E 0000 add [bx+si],al
cs:0010 0000 add [bx+si],al
cs:0012 0000 add [bx+si],al
cs:0014 0000 add [bx+si],al
cs:0016 0000 add [bx+si],al
cs:0018 0000 add [bx+si],al
cs:001A FF db FF
cs:001B FF00 inc word ptr [bx+si]
cs:001D 00FF add bh,bh

ds:0000 19 00 99 02 00 00 00 00 ↓ 00
ds:0008 00 00 00 00 00 00 00 00
ds:0010 B8 5A 46 8E D8 B8 00 4C ↵ZFÄ† L
ds:0018 CD 21 00 00 00 00 00 00 =!
ds:0020 00 00 00 00 00 00 00 00
```

# Формат представления правильных дробей

Дробное число  $M_1$ , заданное в восьмеричной системе счисления, представить в формате с ФТ в 8-разрядном процессоре (1 байт).

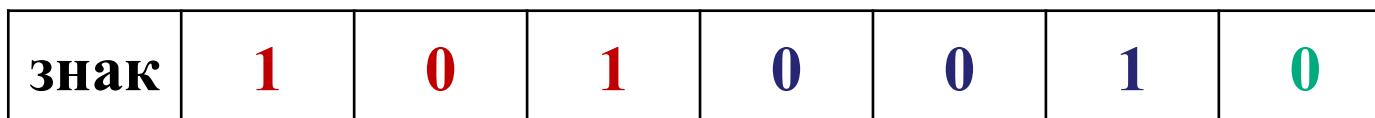
$$M = 0,51_{(8)} = 0,101001_{(2)}$$

без знака:



заполнение разрядов

со знаком:



$$\approx 0, A4_{(16)}$$



# Целые числа $19_{(16)}$ и $229_{(16)}$ и дробное со знаком $+0,A4_{(16)}$

```
LEK1 - Блокнот
Файл  Правка  Формат  Вид  Справка
|data segment
;=====
      A dw 19h          ;целое
      B dw 299h        ;целое
;-----
      C db 52h         ;дробное = +0,A4h
data ends
;=====
code segment
assume cs:code, ds:data, ss:nothing
start:      mov ax, data ;load adress
            mov ds, ax ; data segment
            ;=====
            ; "тело" ВЫЧИСЛЕНИЯ
            ;=====
quit:      mov ax, 4c00h ; cod to finish 0
            int 21h ; exit to dos
code ends
end start
```

сегмент данных

сегмент кода

Целые числа  $19_{(16)}$  и  $229_{(16)}$  и в 8-разрядном процессоре дробное  $0,44_{(16)}$

```
DOSBox 0.74, Cpu speed: 3000 cycles Frameskip: 0 Program: TT
File View Run Breakpoints Data Options
[ ]-CPU 80486
cs:0000 B85A46 mov ax,465A
cs:0003 8ED8 mov ds,ax
cs:0005 B8004C mov ax,4C00
cs:0008 CD21 int 21
cs:000A 0000 add [bx+si],al
cs:000C 0000 add [bx+si],al
cs:000E 0000 add [bx+si],al
cs:0010 0000 add [bx+si],al
cs:0012 0000 add [bx+si],al
cs:0014 0000 add [bx+si],al
cs:0016 0000 add [bx+si],al
cs:0018 0000 add [bx+si],al
cs:001A FF db FF
cs:001B FF00 inc word ptr [bx+si]
cs:001D 00FF add bh,bh

ds:0000 19 00 99 02 52 00 00 00 ↓ 00R
ds:0008 00 00 00 00 00 00 00 00
ds:0010 BB 5A 46 8E D8 B8 00 4C ↑ZFÄ† L
ds:0018 CD 21 00 00 00 00 00 00 =!
ds:0020 00 00 00 00 00 00 00 00

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-
```

16-разрядный: целые числа  $19_{(16)}$  и  $229_{(16)}$ , а также смешанное  $29,34_{(16)}$   
8-разрядный: дробное  $0,A4_{(16)}$ ;

```
LEK1 - Блокнот
Файл  Правка  Формат  Вид  Справка
|data segment
;=====
      A dw 19h          ;целое
      B dw 299h        ;целое
;-----
      C db 52h          ;дробное = +0,A4h
data ends D dw 2934h
;=====
code segment
assume cs:code, ds:data, ss:nothing
start:      mov ax, data ;load adress
            mov ds, ax ; data segment
            ;=====
            ; "тело" ВЫЧИСЛЕНИЯ
            ;=====
quit:      mov ax, 4c00h ; cod to finish 0
            int 21h ; exit to dos
code ends
end start
```

сегмент данных

сегмент  
кода

Целые числа  $19_{(16)}$  и  $229_{(16)}$ ; дробное  $0,А4_{(16)}$ ; смешанное  $29,34_{(16)}$

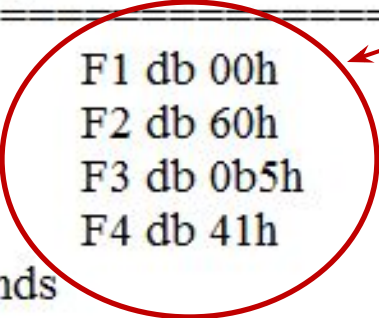
```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TD
File View Run Breakpoints Data Options Wind
[ ]=CPU 80486
cs:0000 B85A46 mov ax,465A
cs:0003 8ED8 mov ds,ax
cs:0005 B8004C mov ax,4C00
cs:0008 CD21 int 21
cs:000A 0000 add [bx+si],al
cs:000C 0000 add [bx+si],al
cs:000E 0000 add [bx+si],al
cs:0010 0000 add [bx+si],al
cs:0012 0000 add [bx+si],al
cs:0014 0000 add [bx+si],al
cs:0016 0000 add [bx+si],al
cs:0018 0000 add [bx+si],al
cs:001A FF db FF
cs:001B FF00 inc word ptr [bx+si]
cs:001D 00FF add bh,bh

ds:0000 19 00 99 02 52 34 29 00 L ÜBR4)
ds:0008 00 00 00 00 00 00 00 00
ds:0010 BB 5A 46 8E DB BB 00 4C 7ZFÄ†† L
ds:0018 CD 21 00 00 00 00 00 00 =!
ds:0020 00 00 00 00 00 00 00 00
```

смешанное  $29,34_{(16)}$

# Вещественное число 16,АС<sub>(16)</sub> в формате КВ

```
LEK11 - Блокнот
Файл  Правка  Формат  Вид  Справка
|data segment
;=====
F1 db 00h           ;младший байт
F2 db 60h
F3 db 0b5h         ;старший байт
F4 db 41h
data ends
;=====
code segment
assume cs:code, ds:data, ss:nothing
start:             mov ax, data ;load adress
                  mov ds, ax ; data segment
                  ;=====
                  ; "тело" ВЫЧИСЛЕНИЯ
                  ;=====
quit:              mov ax, 4c00h ; cod to finish 0
                  int 21h ; exit to dos
code ends
end start
```



сегмент данных

сегмент кода



# Вещественное число 16,АС<sup>(16)</sup> в формате КВ

```
DOS BOX  DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TL
File View Run Breakpoints Data Options
[ ]=CPU 80486
cs:0000 B85A46      mov     ax,465A
cs:0003 8ED8        mov     ds,ax
cs:0005 B8004C        mov     ax,4C00
cs:0008 CD21        int     21
cs:000A 0000        add     [bx+si],al
cs:000C 0000        add     [bx+si],al
cs:000E 0000        add     [bx+si],al
cs:0010 0000        add     [bx+si],al
cs:0012 0000        add     [bx+si],al
cs:0014 0000        add     [bx+si],al
cs:0016 0000        add     [bx+si],al
cs:0018 0000        add     [bx+si],al
cs:001A FF          db     FF
cs:001B FF00       inc     word ptr [bx+si]
cs:001D 00FF       add     bh,bh

ds:0000 00 60 B5 41 00 00 00 00  'A
ds:0008 00 00 00 00 00 00 00 00
ds:0010 B8 5A 46 8E D8 B8 00 4C  ZFA L
ds:0018 CD 21 00 00 00 00 00 00  =!
ds:0020 00 00 00 00 00 00 00 00
```