

Пензенский государственный университет
Кафедра «Вычислительная техника»

ОТЧЕТ

по лабораторной работе № 5

по дисциплине: "Арифметические и логические основы вычислительной
техники"

на тему: "Сложение/вычитание чисел в формате с ФТ большей размерности,
чем разрядность процессора (многократная арифметика)"

Выполнил:

xxxxxxxxxxxxxxxxxxxxxx

Принял:

xxxxxxxxxxxxxxxxxxxxxx

Пенза, 2021

Ход работы

1. Перевела числа $a = 65$ и $b = 35018$ в двоичную систему счисления.

$$\begin{array}{r|l} 65 & 16 \\ -64 & 4 \\ \hline 1 & \end{array}$$

$$a = 65_{10} = 41_{16} = 1000001_2$$

$$\begin{array}{r|l} 35018 & 16 \\ -35008 & \\ \hline 10 & \end{array} \quad \begin{array}{r|l} 2188 & 16 \\ -2176 & \\ \hline 12 & \end{array} \quad \begin{array}{r|l} 136 & 16 \\ -128 & \\ \hline 8 & \end{array}$$

$$b = 35018_{10} = 88CA_{16} = 1000\ 1000\ 1100\ 1010_2$$

2. Представила числа a и b как 32-разрядные операнды в дополнительном коде.

$$[a]_2 = 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0100\ 0001$$

$$[b]_2 = 1111\ 1111\ 1111\ 1111\ 0111\ 0111\ 0011\ 0110$$

3. Выполнила операцию $y = a + b$ для 16-разрядного процессора.

Сложение младших 16 битов операндов:

$[a]_2$	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
+																	
$[b]_2$	0	1	1	1	0	1	1	1	0	0	1	1	0	1	1	1	0
$[y]_2$	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1
CF = 0; SF = 0; OF = 0; ZF = 0																	

Сложение старших 16 битов операндов:

$[a]_2$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
+																	
$[b]_2$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
CF																	0
$[y]_2$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
CF = 0; SF = 1; OF = 0; ZF = 0																	

$$[y]_2 = 1111\ 1111\ 1111\ 1111\ 0111\ 0111\ 0111\ 0111$$

$$y = -0000\ 0000\ 0000\ 0000\ 1000\ 1000\ 1000\ 1001_2 = -8889_{16} = -34953_{10}$$

Результат верный.

```

File View Run Breakpoints Data Options Window Help
[1]-CPU 80486-1=[↑][↓]
cs:0020>B8004C      mov     ax,4C00      ax 7777      c=0
cs:002E CD15        int     15          bx FFFF      z=0
cs:0030 0000        add     [bx+si],al    cx 7736      s=1
cs:0032 0000        add     [bx+si],al    dx FFFF      o=0
cs:0034 0000        add     [bx+si],al    si 0000      p=1
cs:0036 0000        add     [bx+si],al    di 0000      a=0
cs:0038 0000        add     [bx+si],al    bp 0000      i=1
cs:003A 0000        add     [bx+si],al    sp 0000      d=0
cs:003C 0000        add     [bx+si],al    ds 4ACF
cs:003E 0000        add     [bx+si],al    es 4ABF
cs:0040 0000        add     [bx+si],al    ss 4ACE
cs:0042 0000        add     [bx+si],al    cs 4AD0
cs:0044 0000        add     [bx+si],al    ip 002B
cs:0046 0000        add     [bx+si],al
cs:0048 0000        add     [bx+si],al

ds:0000 41 00 00 00 36 77 FF FF A 6w
ds:0008 77 77 FF FF 00 00 00 00 ww
ds:0010 8B 1A 8E D8 33 C0 A1 7<J0+3 L6
ds:0018 00 y 3B 1E 02 00 8B 0E 1A0 11
ds:0020 04 00 8B 16 06 00 03 C1 + 1-+ 1

ss:0008 6864
ss:0006 742E
ss:0004 706C
ss:0002 6568
ss:0000>6474

```

$$[-b]_2 = 0000\ 0000\ 0000\ 0000\ 1000\ 1000\ 1100\ 1010$$

$[a]_2$	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
$[b]_2$	1	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0
$[y]_2$	1	0	0	0	1	0	0	1	0	0	0	0	1	0	1	1
CF = 0; SF = 1; OF = 0; ZF = 0																

[illegible]

CF = 0; SF = 0; OF = 0; ZF = 0

$[y]_2 = 0000\ 0000\ 0000\ 0000\ 1000\ 1001\ 0000\ 1011$

$y = 1000\ 1001\ 0000\ 1011_2 = 890B_{16} = 35083_{10}$

Проверка:

$y = a - b = 41_{16} + 88CA_{16} = 890B_{16}$

$y = a - b = 65_{10} + 35018_{10} = 35083_{10}$

Результат верный.

6. Выполнила проверку.

The screenshot shows the CPU 80486 debugger window. The assembly code is as follows:

Address	Disassembly	Comment
cs:0000	B8CF4A	mov ax,4ACF
cs:0003	8ED8	mov ds,ax
cs:0005	33C0	xor ax,ax
cs:0007	A10000	mov ax,[0000]
cs:000A	8B1E0200	mov bx,[0002]
cs:000E	8B0E0400	mov cx,[0004]
cs:0012	8B160600	mov dx,[0006]
cs:0016	03C1	add ax,cx
cs:0018	13DA	adc bx,dx
cs:001A	700A	jo 0026
cs:001C	A30800	mov [0008],ax
cs:001F	891E0A00	mov [000A],bx
cs:0023	EB06	jmp 002B
cs:0025	90	nop
cs:0026	C6060C0001	mov byte ptr [000C],01

The registers are shown on the right:

Register	Value	Comment
ax	890B	c=0
bx	0000	z=1
cx	88CA	s=0
dx	0000	o=0
si	0000	p=1
di	0000	a=0
bp	0000	i=1
sp	0000	d=0
ds	4ACF	
es	4ABF	
ss	4ACE	
cs	4AD0	
ip	001F	

The data segment (ds) is shown at the bottom:

Address	Value	Comment
ds:0000	41 00 00 00	CA 88 00 00 A
ds:0008	0B 09 00 00	00 00 00 00 8H
ds:0010	00 00 00 00	00 00 00 00 8H
ds:0018	00 00 00 00	00 00 00 00 8H
ds:0020	04 00 0B 16	00 03 C1 + 00 00 01

The status bar at the bottom shows: F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

Вывод: научилась выполнять сложение и вычитание чисел в формате с фиксированной точкой большей размерности, чем разрядность процессора.