**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО**

**ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №3

Дисциплина: Архитектура вычислительных систем

по теме Арифметические команды центрального процессора

Выполнил: ст. группы ВТ-32  
 Воскобойников И. С.

Проверил: Осипов.О.В

**Белгород 2020**

**Цель работы**: изучение арифметических команд центрального процессора для работы

с целыми числами.

**Задания для выполнения к работе**

* Написать программу для вычисления значения арифметического выражения

согласно варианту задания. Все переменные, используемые в программе,

требуется использовать как знаковые и расширять до размерности двойного

слова. Результат должен быть записан в регистр EAX. Если результат содержит

остаток от деления, оставить его в регистре EDX. Подобрать набор тестовых

данных (не менее 3).

* Написать программу для сложения или вычитания целых беззнаковых чисел

большой размерности (размерность и операция зависят от варианта задания).

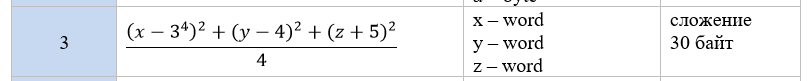
Младшие байты при этом хранить по младшему адресу. Подобрать наборы

тестовых данных (не менее 3). Для выполнения этого задания изучить

теоретический материал главы «Вычитание и сложение операндов большой

размерности», начиная со страницы 176 учебника Юрова «Assembler».

**Вариант 3**

****

**Задание 1**

.386 ; Тип процессора

.model flat, stdcall ; Модель памяти и стиль вызова подпрограмм

option casemap: none ; Чувствительность к регистру

; --- Подключение файлов с кодом, макросами, константами, прототипами функций и т.д.

include c:\masm32\include\windows.inc

include c:\masm32\include\kernel32.inc

include c:\masm32\include\user32.inc

include c:\masm32\include\msvcrt.inc

; --- Подключаемые библиотеки ---

includelib c:\masm32\lib\user32.lib

includelib c:\masm32\lib\kernel32.lib

includelib c:\masm32\lib\msvcrt.lib

; --- Сегмент данных ---

.data

x dw ?

y dw ?

z dw ?

fin db "%d %d %d", 0

format db "EAX = %d", 10, 13, 0

fh db "EAX(h) = %x; EAX = %d", 0

output dd 0

; --- Сегмент кода ---

.code

start:

push offset z

push offset y

push offset x

push offset fin

call crt\_scanf

add esp, 4\*4 ;очистка стека

push 0

MOVSX AX, x ; AX = x

SUB EAX, 3\*3\*3\*3 ;EAX = EAX - 3^4 = (x-3^4)

IMUL EAX ; EAX = EAX \* EAX = (x-3^4)^2

MOVSX CX, y ; CX = y

SUB ECX,4 ; ECX = ECX - 4= (y-4)

IMUL ECX ; ECX = ECX \* ECX = (y-4)^2

ADD EAX, ECX ; EAX = EAX + ECX = (x-3^4)^2 + (y-4)^2

MOVSX BX, z ; BX = z

ADD BX, 5 ; EBX = EBX + 5 = (z+5)

IMUL EBX ; EBX = EBX \* EBX = (z+5)^2

ADD EAX, EBX ; EAX = EAX + EBX = (x-3^4)^2 + (y-4)^2 + (z+5)^2

MOVSX DX, 4 ; DX = 4

XCHG EAX, EDX ; Обмен местами содержимого EAX и EDX

CDQ ; Расширение перед делением до двойного слова

IDIV EDX ; EAX = EDX:EAX = (x-3^4)^2 + (y-4)^2 + (z+5)^2 / 4

push output

push output

push offset fh

call crt\_printf

add esp, 4\*3 ;очистка стека

push 0

push NULL

call ExitProcess

end start

Тестовые данные

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *x* | *y* | *z* | Частное (EAX) | Остаток (EDX) |
| 81 | 4 | -5 | 00000000h=0 | 00000000h=0 |
| 10 | 14 | 10 | 000053Dh=1341 | 00000005h=5 |
| 151 | 454 | 45 | 000CCFBh=52475 | 00000000h=0 |

**Задание 2**

.386

.model flat, stdcall

option casemap: none

; --- Подключение файлов с кодом, макросами, константами, прототипами функций и т.д.

include C:\masm32\include\windows.inc

include C:\masm32\include\kernel32.inc

include C:\masm32\include\user32.inc

include C:\masm32\include\msvcrt.inc

; --- Подключаемые библиотеки ---

includelib C:\masm32\lib\user32.lib

includelib C:\masm32\lib\kernel32.lib

includelib C:\masm32\lib\msvcrt.lib

.data

op1 db 0h,0h,0h,0h, 0h,0h,0h,0h, 0h,0h,0h,0h, 0h,0h,0h,0h, 0FFh,0FFh

op2 db 0h,0h,0h,0h, 0h,0h,0h,0h, 0h,0h,0h,0h, 0h,0h,0h,0h, 0FFh,0FEh

res dd 18 dup(?)

format db "Res = %x%x%x%x%x", 0

b1 dd 0

b2 dd 0

b3 dd 0

b4 dd 0

b5 dd 0

.code

start:

; Обрабатываем младшие 4 байта.

mov eax, dword ptr op1[0]

add eax, dword ptr op2[0]

mov b1, eax

mov dword ptr res[0], eax

; Обрабатываем следующие 4 байта (всего - 8 байт).

mov eax, dword ptr op1[4]

adc eax, dword ptr op2[4]

mov b2,eax

mov dword ptr res[4], eax

; Обрабатываем следующие 4 байта (всего - 12 байт).

mov eax, dword ptr op1[8]

adc eax, dword ptr op2[8]

mov b3, eax

mov dword ptr res[8], eax

; Обрабатываем следующие 4 байта (всего - 16 байт).

mov eax, dword ptr op1[12]

adc eax, dword ptr op2[12]

mov b4, eax

mov dword ptr res[12], eax

; Обрабатываем последние 2 байта (всего - 18 байт)

mov ax, word ptr op1[16]

adc ax, word ptr op2[16]

mov b5, eax

mov word ptr res[16], ax

push b5

push b4

push b3

push b2

push b1

push offset format

call crt\_printf

add esp, 4\*6 ;очистка стека

call crt\_\_getch

push 0

call ExitProcess

end start

