**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-31  
Новожен Н.В

Проверил: Осипов.О.В

**Белгород 2019**

**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 11**

**(r / s+55)4-1**

r – dword сложение; s – byte

2 задание : **сложение 20 байт**

**Задание 1**

.386

.MODEL FLAT, STDCALL

OPTION CASEMAP: NONE

include E:\masm32\include\windows.inc

include E:\masm32\include\kernel32.inc

include E:\masm32\include\user32.inc

includelib E:\masm32\lib\user32.lib

includelib E:\masm32\lib\kernel32.lib

.DATA

r dw -6248

s dw 2

.CODE

START:

MOVSX EAX, r

MOV EBX, EAX

XOR EAX, EAX

MOVSX EAX, s

XCHG EAX, EBX

CDQ

IDIV EBX ; **EAX=EAX/EBX=r/s**

XOR EDX, EDX

ADC EAX, 5\*5\*5\*5\*5 ; **EAX=r/s+55**

IMUL EAX ; **EAX=EAX\*EAX= (r/s+55)2**

IMUL EAX ; **EAX=EAX\*EAX= (r/s+55)4**

DEC EAX ; **EAX= (r/s+55)4 -1**

**(доп EAX +2\*\*31-5)**

push NULL

call ExitProcess

END START

Тестовые данные

|  |  |  |  |
| --- | --- | --- | --- |
| *s* | *r* | Частное (EAX) | Остаток (EDX) |
| 2 | -6248 | 00000000h=0 | 00000000h=0 |
| 4 | -6248 | Down reg  8E5563F0h=2387960816 | Up reg  56Dh=1389 |
| 1 | -6248 | Down reg  ABEF4A90h= 2884586128 | Up reg  5683h= 22147 |

**Задание 2(сложение 20 байт)**

*При выполнении второго задания числа требуется хранить в виде последовательности байт следующим образом:*

.386

.MODEL FLAT, STDCALL

OPTION CASEMAP: NONE

include E:\masm32\include\windows.inc

include E:\masm32\include\kernel32.inc

include E:\masm32\include\user32.inc

includelib E:\masm32\lib\user32.lib

includelib E:\masm32\lib\kernel32.lib

.DATA

a db 0FFh, 0FFh, 0FFh, 0FFh,0FFh, 0FFh, 0FFh, 0FFh,0FFh, 0FFh, 0FFh, 0FFh,0FFh, 0FFh, 0FFh, 0FFh,0FFh, 0FFh, 0FFh, 0FFh; 20 байт

b db 01h, 00h, 00h,00h, 00h, 00h,00h, 00h, 00h,00h, 00h, 00h,00h, 00h, 00h,00h, 00h, 00h,00h, 00h ; 20 байт

res db 21 dup(?) ; Для результата резервируется на один байт больше

.CODE

START:

MOV ESI ,offset a

MOV EDI ,offset b

MOV EAX, DWORD PTR[ESI]

MOV EBX, DWORD PTR[EDI]

XOR ECX, ECX

ADD EAX, EBX

ADC ECX, 0

MOV DWORD PTR[res],EAX

MOV EAX, DWORD PTR[ESI+4]

MOV EBX, DWORD PTR[EDI+4]

ADD EAX, ECX

ADD EAX, EBX

MOV DWORD PTR[res+4],EAX

XOR EAX, EAX

ADC ECX, 0

MOV EAX,DWORD PTR[ESI+8]

MOV EBX,DWORD PTR[EDI+8]

ADD EAX, ECX

ADD EAX, EBX

MOV DWORD PTR[res+8],EAX

XOR EAX, EAX

ADC ECX, 0

MOV EAX,DWORD PTR[ESI+12]

MOV EBX,DWORD PTR[EDI+12]

ADD EAX, ECX

ADD EAX, EBX

MOV DWORD PTR[res+12],EAX

XOR EAX, EAX

ADC ECX, 0

MOV EAX,DWORD PTR[ESI+16]

MOV EBX,DWORD PTR[EDI+16]

ADD EAX, ECX

ADD EAX, EBX

MOV DWORD PTR[res+16],EAX

XOR EAX, EAX

ADC ECX, 0

XOR AL,AL

ADD AL,CL

MOV [res+20],AL

push NULL

call ExitProcess

END START

**res = 3F E0 46 29|9A 61 02 20| DA 12 0D AD|0C 7E F1 67|35 44 A7 AC| 00**

**(число ACA7443567F17E0CAD0D12DA2002619A2946E03F)**

**Вывод:** мы изучили арифметические команды центрального процессора для работы с целыми числами, научились вычислять длинные числа заданной длины и изучили команды изменения размерности и знака числа, научились работать с массивами в вссемблере.