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

по дисциплине: Архитектура вычислительных систем

тема: «Арифметические команды центрального процессора»

Выполнил: ст. группы ПВ-223

Дмитриев Андрей Александрович

Проверил:

Осипов Олег Васильевич

Белгород 2024 г.

**Цель работы:** изучение арифметических команд центрального процессора для работы с целыми числами.

**Задания для выполнения к работе**

1. 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). Каждая строка исходного кода программы обязательно должна быть прокомментирована. Программы без подробных комментариев не принимаются!
2. 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».

Задние варианта 2:

1. вычитание 14 байт.

**Выполнение работы**

**Задание 1:**

Код программы:

.DATA

    a DW ?

    b DW ?

    d DB ?

    format db "Result: %d", 0Ah, 0

.CODE

START:                          ; ab+ad+bd-(a+1)/d-1

    ; Вычисляем a \* b

    movsx eax, a                ; Загрузить a в EAX и расширить до 32 бит

    movsx ebx, b                ; Загрузить b в EBX и расширить до 32 бит

    imul  eax, ebx              ; Выполнить a \* b

    push  eax                   ; Сохраняем результат в стеке

    ; Вычисляем a \* d

    movsx eax, a                ; Загрузить a в EAX и расширить до 32 бит

    movsx ebx, d                ; Загрузить d в EBX и расширить до 32 бит

    imul  eax, ebx              ; Выполнить a \* d

    ; Вычисляем a \* b + a \* d

    add   dword ptr [esp], eax  ; Добавить результат сложения в стек

    ; Вычисляем b \* d

    movsx eax, b                ; Загрузить b в EAX и расширить до 32 бит

    movsx ebx, d                ; Загрузить d в EBX и расширить до 32 бит

    imul  eax, ebx              ; Выполнить b \* d

    ; Вычисляем a \* b + a \* d + b \* d

    add   dword ptr [esp], eax  ; Добавить результат сложения в стек

    ; Вычислим a + 1

    movsx eax, a                ; Загрузить a в EAX и расширить до 32 бит

    inc   eax                   ; Прибавляем 1

    ; Вычислим (a + 1) / d

    movsx ebx, d                ; Загрузить d в EBX и расширить до 32 бит

    cdq                         ; Расширение делимого со знаком ax -> dx:ax

    idiv  ebx                   ; Делим предыдущий результат на d

    ; Вычислим a \* b + a \* d + b \* d - (a + 1) / d

    sub   dword ptr [esp], eax  ; Вычитаем из суммы, лежащей в стеке

    ; Вычислим a \* b + a \* d + b \* d - (a + 1) / d - 1

    dec   dword ptr [esp]       ; Вычитаем 1

    ; Восстановление результата из стека в EAX

    pop   eax                   ; Переносим из стека в EAX

    ; Вывод результата в консоль

    push  eax                   ; Помещаем результат в стек для вывода

    push  offset format         ; Помещаем строку формата в стек

    call  crt\_printf            ; Вызов printf для вывода результата

    ; Завершаем программу

    push 0

    call ExitProcess

END START

Тестовые данные:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | b | d | EAX | EDX |
| 5 | 4 | 2 | 00000022h=34 | 00000000h=0 |
| -5 | 4 | 2 | FFFFFEBh=-21 | 00000000h=0 |
| 5 | 4 | -2 | 00000004h=4 | 00000000h=0 |
| 6 | 4 | 3 | 00000033h=51 | 00000001h=1 |

**Задание 2:**

Код программы:

.DATA

    a db 15h, 0DDh, 34h, 4Bh, 57h, 7Fh, 0CDh, 05h, 2Dh, 13h, 44h, 0BBh, 12h, 3Fh

    b db 15h, 0DDh, 34h, 4Bh, 57h, 7Fh, 0CDh, 05h, 2Dh, 13h, 44h, 0BBh, 12h, 3Fh

    r db 14 dup(?) ; Для результата резервируется 14 байт

.CODE

START:

    ; Вычитание младших 4 байт

    mov EAX, dword ptr a[0]

    sub EAX, dword ptr b[0]

    mov dword ptr r[0], EAX

    ; Вычитание следующих 4 байт с учетом переноса

    mov EAX, dword ptr a[4]

    sbb EAX, dword ptr b[4]

    mov dword ptr r[4], EAX

    ; Вычитание следующих 4 байт с учетом переноса

    mov EAX, dword ptr a[8]

    sbb EAX, dword ptr b[8]

    mov dword ptr r[8], EAX

    ; Вычитание оставшихся 2 байт с учетом переноса

    mov AX, word ptr a[12]

    sbb AX, word ptr b[12]

    mov word ptr r[12], AX

    push 0

    call ExitProcess

END START

Тестовые данные:

|  |  |
| --- | --- |
| a | 0001000000000000000000000000h |
| b | 0002000000000000000000000000h |
| r | 00FFFFFFFFFFFFFFFFFFFFFFFh |

|  |  |
| --- | --- |
| a | 0000000000000000010000000000h |
| b | 0000010000000000000000000000h |
| r | 00FFFFFFFFFF000000000000000h |

|  |  |
| --- | --- |
| a | 15DD344B577FCD052D1344BB123Fh |
| b | 15DD344B577FCD052D1344BB123Fh |
| r | 0000000000000000000000000000h |

**Вывод:** в ходе работы изучены арифметические команды центрального процессора для работы с целыми числами.