|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** ***ИУК «Информатика и управление»***

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ЛАБОРАТОРНАЯ РАБОТА №3**

**«Программирование арифметических операций»**

**ДИСЦИПЛИНА: «Машинно-зависимые языки программирования»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-32Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Карельский М.К. )  (Подпись) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Амеличева К.А. )  (Подпись) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2021

**Цель:** научиться выполнять операции сложения, вычитания, умножения и деления с целыми числами без знака и со знаком.

**Вариант 15**

**Задание:**

Разработать программу для каждого случая согласно индивидуальному варианту заданию при решении следующей задачи.

1. В сегменте данных определить:
   * байтовые значения Хi, a, b, c, d в десятичной системе счисления;
2. В сегменте данных зарезервировать байтовые ячейки для хранения:
   * суммы и разности с нулевыми первоначальными значениями,
   * двухбайтовую ячейку для хранения произведения с единичным первоначальным значением,
   * две байтовые ячейки для хранения остатка от деления и частного с произвольными первоначальными значениями.
3. Написать программу на языке Ассемблер, которая выполняет следующие операции над байтовыми значениями.
   * из каждого целого числа Xi, где (i = 1, 2, 3), требуется вычесть число a,
   * к результату добавить число b,
   * полученный результат умножить на число c,
   * полученный результат разделить на число d.
   * Определить сумму инкремента неполного частного e и декремента остатка от деления f.
4. Результаты работы программ вместе с их листингами внести в отчет и сравнить со значениями, полученными при выполнении тех же вычислений вручную.

**Решение:**

*Листинг:*

.model small

.stack 100h

.data

X1 db 99

X2 db 0B6h

X3 db 201

a db 21

b db 36h

c db 7

d db 2

sum db 3 dup(0)

dif db 3 dup(0)

pro dw 3 dup(1)

rem dw 3 dup(?)

quo dw 3 dup(?)

res dw 3 dup(0)

.code

Start:

mov ax, @data

mov ds, ax

mov ax, 0

mov al, X1

sub al, a

mov dif, al

add al, b

mov sum, al

mov al, c

mul sum

mov pro, ax

mov dx, 0

mov bx, 0

mov bl, d

div bx

mov rem, dx

mov quo, ax

inc ax

dec dx

mov bx, ax

add bx, dx

mov res, bx

mov ax, 0

mov al, X2

sub al, a

mov dif+1, al

add al, b

mov sum+1, al

mov al, c

mul sum+1

mov pro+2, ax

mov dx, 0

mov bx, 0

mov bl, d

div bx

mov rem+2, dx

mov quo+2, ax

inc ax

dec dx

mov bx, ax

add bx, dx

mov res+2, bx

mov ax, 0

mov al, X3

sub al, a

mov dif+2, al

add al, b

mov sum+2, al

mov al, c

mul sum+2

mov pro+4, ax

mov dx, 0

mov bx, 0

mov bl, d

div bx

mov rem+4, dx

mov quo+4, ax

inc ax

dec dx

mov bx, ax

add bx, dx

mov res+4, bx

mov ax, 4C00h

int 21h

end start

END

*Вычисления:*

x1 = 99

x2 = 182

x3 = 201

99 – 21 = 78

182 – 21 = 161

201 – 21 = 180

78 + 54 = 132

161 + 54 = 215

180 + 54 = 234

132 · 7 = 924

215 · 7 = 1505

234 · 7 = 1638

924 / 2 = 462

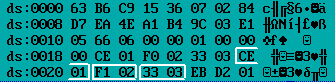
1505 / 2 = 752 (1)

1638 / 2 = 819

462 + 1 + 0 – 1 = 462 = 1CEh

752 + 1 + 1 – 1 = 753 = 2F1h

819 + 1 + 0 – 1 = 819 = 333h



**Рисунок 1.** Dump с результатами

|  |  |  |
| --- | --- | --- |
| Оператор | Операнд-приемник | |
| До изменения | После изменения |
| mov al, X1 | al: 00 | al: 63 |
| sub al, a | al: 63 | al: 4E |
| mov dif, al | dif: 00 | dif: 4E |
| add al, b | al: 4E | al: 84 |
| mov sum, al | sum: 00 | sum: 84 |
| mov al, c | al: 84 | al: 07 |
| mul sum | ax: 00 07 | ax: 03 9C |
| mov pro, ax | pro: 01 00 | pro: 9C 03 |
| mov dx, 0 | dx: 00 00 | dx: 00 00 |
| mov bx, 0 | bx: 00 00 | bx: 00 00 |
| mov bl, d | bl: 00 | bl: 02 |
| div bx | ax: 03 9C; dx: 00 00 | ax: 01 CE; dx: 00 00 |
| mov rem, dx | rem: 00 00 | rem: 00 00 |
| mov quo, ax | quo: 00 00 | quo: CE 01 |
| inc ax | ax: 01 CE | ax: 01 CF |
| dec dx | dx: 00 00 | dx: FF FF |
| mov bx, ax | bx: 00 02 | bx: 01 CF |
| add bx, dx | bx: 01 CF | bx: 01 CE |
| mov res, bx | res: 00 00 | res: CE 01 |
| mov ax, 0 | ax: 01 CF | ax: 00 00 |
| mov al, X2 | al: 00 | al: B6 |
| sub al, a | al: B6 | al: A1 |
| mov dif+1, al | dif+1: 00 | dif+1: A1 |
| add al, b | al: A1 | al: D7 |
| mov sum+1, al | sum+1: 00 | sum+1: D7 |
| mov al, c | al: D7 | al: 07 |
| mul sum+1 | ax: 00 07 | ax: 05 E1 |
| mov pro+2, ax | pro+2: 01 00 | pro+2: E1 05 |
| mov dx, 0 | dx: FF FF | dx: 00 00 |
| mov bx, 0 | bx: 01 CE | bx: 00 00 |
| mov bl, d | bl: 00 | bl: 02 |
| div bx | ax: 05 E1; dx: 00 00 | ax: 02 F0; dx: 00 01 |
| mov rem+2, dx | rem+2: 00 00 | rem+2: 01 00 |
| mov quo+2, ax | quo+2: 00 00 | quo+2: F0 02 |
| inc ax | ax: 02 F0 | ax: 02 F1 |
| dec dx | dx: 00 01 | dx: 00 00 |
| mov bx, ax | bx: 00 02 | bx: 02 F1 |
| add bx, dx | bx: 02 F1 | bx: 02 F1 |
| mov res+2, bx | res+2: 00 00 | res+2: F1 02 |
| mov ax, 0 | ax: 02 F1 | ax: 00 00 |
| mov al, X3 | al: 00 | al: C9 |
| sub al, a | al: C9 | al: B4 |
| mov dif+2, al | dif+2: 00 | dif+2: B4 |
| add al, b | al: B4 | al: EA |
| mov sum+2, al | sum+2: 00 | sum+2: EA |
| mov al, c | al: EA | al: 07 |
| mul sum+2 | ax: 00 07 | ax: 06 66 |
| mov pro+4, ax | pro+4: 01 00 | pro+4: 66 06 |
| mov dx, 0 | dx: 00 00 | dx: 00 00 |
| mov bx, 0 | bx: 02 F1 | bx: 00 00 |
| mov bl, d | bl: 00 | bl: 02 |
| div bx | ax: 06 66; dx: 00 00 | ax: 03 33; dx: 00 00 |
| mov rem+4, dx | rem+4: 00 00 | rem+4: 00 00 |
| mov quo+4, ax | quo+4: 00 00 | quo+4: 33 03 |
| inc ax | ax: 03 33 | ax: 03 34 |
| dec dx | dx: 00 00 | dx: FF FF |
| mov bx, ax | bx: 00 02 | bx: 03 34 |
| add bx, dx | bx: 03 34 | bx: 03 33 |
| mov res+4, bx | res+4: 00 00 | res+4: 33 03 |

**Таблица 1.** Состояние программы в ходе ее выполнения

**Вариант 5**

**Задание:**

1. Используя текстовый редактор, создать и отредактировать исходный модуль программы Prog\_3\_1.asm, которая вычисляет значение Y в соответствии с вариантом задания.
2. Распишем формулу по отдельным операциям, аналогично образцу.
3. Написать три варианта программы, позволяющие работать с исходных данных:
   * диапазон от 0 до 10
   * диапазон от -100 до 100
   * диапазон от -1000 до 1000
4. Скомпилируйте и выполните отладку полученной программы, убедившись в правильности проведения вычислений, заполнив таблицу пошаговой отладки с указанием на отслеживаемые данных.

**Решение:**

*Детализация формулы:*

AL ← b; b в AL

DX ← 3; 3 в DX

AX ← (AL) \* (DL); 3b в AX

BX ← (AX); 3b в BX

AL ← a; a в AL

AX ← (AL) \* a; a2 в AX

AL ← (AL) + x; x + a2 в AL

DX ← 2; 2 в DX

AX ← (AL) \* (DL); 2(x + a2) в AX

CX ← (AX); 2(x + a2) в CX

AL ← x; x в AL

AX ← (AL) \* x; x2 в AX

AX ← (AL) \* (BL); 3bx2 в AX

CX ← (CX) – (AX); 2(x + a2) – 3bx2 в CX

AX ← (BX); 3b в AX

AX ← (AL) \* (DL); 6b в AX

BX ← (AX); 6b в BX

AX ← (CX); 2(x + a2) – 3bx2 в AX

AX ← (AL) / (BL); (2(x + a2) – 3bx2) / 6b в AX

y ← (AX); (2(x + a2) – 3bx2) / 6b в y

*Листинг первой программы:*

.model small

.stack 100h

.data

a db 0

b db 10

x db 10

y dw 0

.code

Start:

mov ax, @data

mov ds, ax

mov ax, 0

mov al, b

mov dx, 3

mul dl

mov bx, ax

mov ax, 0

mov al, a

mul a

add al, x

mov dx, 2

mul dl

mov cx, ax

mov ax, 0

mov al, x

mul x

mul bl

sub cx, ax

mov ax, bx

mul dl

mov bx, ax

mov ax, cx

idiv bl

mov y, ax

mov ax, 4C00h

int 21h

end start

END

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Оператор | Операнд-приемник | | Арифметическое вычисление | Результат |
| До | После |
| mov al, b | al: 00 | al: 0A | 3b | 1E |
| mov dx, 3 | dx: 00 00 | dx: 00 03 |
| mul dl | ax: 00 0A | ax: 00 1E |
| mov bx, ax | bx: 00 00 | bx: 00 1E |
| mov ax, 0 | ax: 00 1E | ax: 00 00 |
| mov al, a | al: 00 | al: 00 | a2 | 0 |
| mul a | ax: 00 00 | ax: 00 00 |
| add al, x | al: 00 | al: 00 0A | x + a2 | A |
| mov dx, 2 | dx: 00 03 | dx: 00 02 | 2(x + a2) | 14 |
| mul dl | ax: 00 0A | ax: 00 14 |
| mov cx, ax | cx: 00 00 | cx: 00 14 |
| mov ax, 0 | ax: 00 14 | ax: 00 00 |
| mov al, x | al: 00 | al: 0A | x2 | 64 |
| mul x | ax: 00 0A | ax: 00 64 |
| mul bl | ax: 00 64 | ax: 0B B8 | 3bx2 | B B8 |
| sub cx, ax | cx: 00 14 | cx: F4 5C | 2(x + a2) – 3bx2 | F4 5C |
| mov ax, bx | ax: 0B B8 | ax: 00 1E | 6b | 3C |
| mul dl | ax: 00 1E | ax: 00 3C |
| mov bx, ax | bx: 00 1E | bx: 00 3C |
| mov ax, cx | ax: 00 3C | ax: F4 5C | (2(x + a2) – 3bx2) / 6b | D8 (CF) |
| idiv bl | ax: F4 5C | ax: D8 CF |
| mov y, ax | y: 00 00 | y: CF D8 |

**Таблица 2.** Состояние первой программы в ходе ее выполнения

*Листинг второй программы:*

.model small

.486

.stack 100h

.data

a db 100

b db -1

x db 100

yQuo dw 0

yRem dw 0

temp dd 0

.code

Start:

mov ax, @data

mov ds, ax

mov ax, 0

mov al, b

mov dx, 3

imul dl

mov bx, ax

mov ax, 0

mov al, a

imul a

movsx cx, x

add ax, cx

mov dx, 2

imul dx

mov cx, ax

mov ax, 0

mov al, x

imul x

imul bx

mov si, offset temp

mov [si], ax

mov [si] + 2, dx

mov ax, cx

cwde

sub eax, temp

mov temp, eax

mov eax, 0

mov ax, bx

mov bx, 2

imul bx

mov bx, ax

mov ax, 0

mov ax, [si]

mov dx, [si] + 2

idiv bx

mov yQuo, ax

mov yRem, dx

mov ax, 4C00h

int 21h

end start

END

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Оператор | Операнд-приемник | | Арифметическое вычисление | Результат |
| До | После |
| mov al, b | al: 00 | al: FF | 3b | FF FD |
| mov dx, 3 | dx: 00 00 | dx: 00 03 |
| imul dl | ax: 00 FF | ax: FF FD |
| mov bx, ax | bx: 00 00 | bx: FF FD |
| mov ax, 0 | ax: FF FD | ax: 00 00 |
| mov al, a | al: 00 | al: 64 | a2 | 27 10 |
| imul a | ax: 00 64 | ax: 27 10 |
| movsx cx, x | cx: 00 00 | cx: 00 64 | x + a2 | 27 74 |
| add ax, cx | ax: 27 10 | ax: 27 74 |
| mov dx, 2 | dx: 00 03 | dx: 00 02 | 2(x + a2) | 4E E8 |
| imul dx | ax: 27 74; dx: 00 02 | ax: 4E E8; dx: 00 00 |
| mov cx, ax | cx: 00 64 | cx: 4E E8 |
| mov ax, 0 | ax: 4E E8 | ax: 00 00 |
| mov al, x | al: 00 | al: 64 | x2 | 27 10 |
| imul x | ax: 00 64 | ax: 27 10 |
| imul bx | ax: 27 10; dx: 00 00 | ax: 8A D0; dx: FF FF | 3bx2 | FF FF D0 8A |
| mov si, offset temp | si: 00 00 | si: 00 07 |
| mov [si], ax | [si]: 00 00 | [si]: D0 8A |
| mov [si]+2, dx | [si]+2: 00 00 | [si]+2: FF FF |
| mov ax, cx | ax: 8A D0 | ax: 4E E8 | 2(x + a2) – 3bx2 | 00 00 C4 18 |
| cwde | eax: 00 00 4E E8 | eax: 00 00 4E E8 |
| sub eax, temp | eax: 00 00 4E E8 | eax: 00 00 C4 18 |
| mov temp, eax | temp: D0 8A FF FF | temp: 18 C4 00 00 |
| mov eax, 0 | eax: 00 00 C4 18 | eax: 00 00 00 00 |
| mov ax, bx | ax: 00 00 | ax: FF FD | 6b | FF FA |
| mov bx, 2 | bx: FF FD | bx: 00 02 |
| imul bx | ax: FF FD; dx: FF FF | ax: FF FA; dx: FF FF |
| mov bx, ax | bx: 00 02 | bx: FF FA |
| mov ax, 0 | ax: FF FA | ax: 00 00 |
| mov ax, [si] | ax: 00 00 | ax: C4 18 | (2(x + a2) – 3bx2) / 6b | 52 DF (04 00) |
| mov dx, [si]+2 | dx: FF FF | dx: 00 00 |
| idiv bx | ax: C4 18; dx: 00 00 | ax: DF 52; dx: 00 04 |
| mov yQuo, ax | yQuo: 00 00 | yQuo: 52 DF |
| mov yRem, dx | yRem: 00 00 | yRem: 04 00 |

**Таблица 3.** Состояние второй программы в ходе ее выполнения

*Листинг третьей программы:*

.model small

.486

.stack 100h

.data

a dw 1000

b dw -1

x dw 1000

yQuo dd 0

yRem dw 0

temp dd 0

.code

Start:

mov ax, @data

mov ds, ax

mov ax, 0

mov ax, b

mov dx, 3

imul dx

cwde

mov ebx, eax

mov eax, 0

mov ax, a

imul a

mov si, offset temp

mov [si], ax

mov [si] + 2, dx

movsx eax, x

add eax, temp

mov edx, 2

imul edx

mov ecx, eax

mov eax, 0

mov edx, 0

mov ax, x

imul x

mov [si], ax

mov [si] + 2, dx

mov eax, temp

imul ebx

mov temp, eax

mov eax, 0

mov edx, 0

mov eax, ecx

sub eax, temp

mov ecx, eax

mov eax, 0

mov ax, bx

mov ebx, 0

mov bx, 2

imul bx

cwde

mov ebx, eax

mov eax, 0

mov eax, ecx

cdq

idiv ebx

mov yQuo, eax

mov yRem, dx

mov ax, 4C00h

int 21h

end start

END

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Оператор | Операнд-приемник | | Арифметическое вычисление | Результат |
| До | После |
| mov ax, b | ax: 00 00 | ax: FF FF | 3b | FF FF FF FD |
| mov dx, 3 | dx: 00 00 | dx: 00 03 |
| imul dx | ax: FF FF; dx: 00 03 | ax: FF FD; dx: FF FF |
| cwde | eax: 00 00 FF FD | eax: FF FF FF FD |
| mov ebx, eax | ebx: 00 00 00 00 | ebx: FF FF FF FD |
| mov eax, 0 | eax: FF FF FF FD | eax: 00 00 00 00 |
| mov ax, a | ax: 00 00 | ax: 03 E8 | a2 | 00 0F 42 40 |
| imul a | ax: 03 E8; dx: FF FF | ax: 42 40; dx: 00 0F |
| mov si, offset temp | si: 00 00 | si: 00 0C |
| mov [si], ax | [si]: 00 00 | [si]: 40 42 |
| mov [si]+2, dx | [si]+2: 00 00 | [si]+2: 0F 00 |
| movsx eax, x | eax: 00 00 42 40 | eax: 00 00 03 E8 | x + a2 | 00 0F 46 28 |
| add eax, temp | eax: 00 00 03 E8 | eax: 00 0F 46 28 |
| mov edx, 2 | edx: 00 00 00 0F | edx: 00 00 00 02 | 2(x + a2) | 00 1E 8C 50 |
| imul edx | eax: 00 0F 46 28; edx: 00 00 00 02 | eax: 00 1E 8C 50; edx: 00 00 00 00 |
| mov ecx, eax | ecx: 00 00 00 00 | ecx: 00 1E 8C 50 |
| mov eax, 0 | eax: 00 1E 8C 50; | eax: 00 00 00 00 |
| mov edx, 0 | edx: 00 00 00 00 | edx: 00 00 00 00 |
| mov ax, x | ax: 00 00 | ax: 03 E8 | x2 | 00 0F 42 40 |
| imul x | ax: 03 E8; dx: 00 00 | ax: 42 40; dx: 00 0F |
| mov [si], ax | [si]: 40 42 | [si]: 40 42 |
| mov [si]+2, dx | [si]+2: 0F 00 | [si]+2: 0F 00 |
| mov eax, temp | eax: 00 00 42 40 | eax: 00 0F 42 40 |
| imul ebx | eax: 00 0F 42 40; edx: 00 00 00 0F | eax: FF D2 39 40; edx: FF FF FF FF | 3bx2 | FF D2 39 40; |
| mov temp, eax | temp: 40 42 0F 00 | temp: 40 39 D2 FF |
| mov eax, 0 | eax: FF D2 39 40 | eax: 00 00 00 00 |
| mov edx, 0 | edx: FF FF FF FF | edx: 00 00 00 00 |
| mov eax, ecx | eax: 00 00 00 00 | eax: 00 1E 8C 50 | 2(x + a2) – 3bx2 | 00 4C 53 10 |
| sub eax, temp | eax: 00 1E 8C 50 | eax: 00 4C 53 10 |
| mov ecx, eax | ecx: 00 1E 8C 50 | ecx: 00 4C 53 10 |
| mov eax, 0 | eax: 00 4C 53 10 | eax: 00 00 00 00 |
| mov ax, bx | ax: 00 00 | ax: FF FD | 6b | FF FF FF FA |
| mov ebx, 0 | ebx: FF FF FF FD | ebx: 00 00 00 00 |
| mov bx, 2 | bx: 00 00 | bx: 00 02 |
| imul bx | ax: FF FD; dx: 00 00 | ax: FF FA; dx: FF FF |
| cwde | eax: 00 00 FF FA | eax: FF FF FF FA |
| mov ebx, eax | ebx: 00 00 00 02 | ebx: FF FF FF FA |
| mov eax, 0 | eax: FF FF FF FA | eax: 00 00 00 00 |
| mov eax, ecx | eax: 00 00 00 00 | eax: 00 4C 53 10 | (2(x + a2) – 3bx2) / 6b | FF F3 47 7E (00 04) |
| cdq | eax: 00 4C 53 10; edx: 00 00 FF FF | eax: 00 4C 53 10; edx: 00 00 00 00 |
| idiv ebx | eax: 00 4C 53 10; edx: 00 00 00 00 | eax: FF F3 47 7E; edx: 00 00 00 04 |
| mov yQuo, eax | yQuo: 00 00 00 00 | yQuo: 7E 47 F3 FF |
| mov yRem, dx | yRem: 00 00 | yRem: 04 00 |

**Таблица 4.** Состояние третьей программы в ходе ее выполнения

**Вывод:** в ходе выполнения лабораторной работы были получены практические навыки реализации операций сложения, вычитания, умножения и деления с целыми числами без знака и со знаком.