|  |  |
| --- | --- |
| **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**

**«Программирование арифметических операций»**

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

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-11Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Отрошенко Т. В.)  (Подпись) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Амеличева К. А.)  (Подпись) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2021

**Цель работы:** практическое овладение навыками разработки программного кода на языке Ассемблер. Программирование арифметических операций. Практическое освоение основных функций отладчика TD.**Задание 1 (Вариант 5)**

|  |  |  |
| --- | --- | --- |
| .model small  .stack 100h  .data  message db 'Offset message','$'  ;1.1  xi          db 0  a           db 29  b           db 13h  c           db 13  d           db 12  ;x1         db 68  ;x2         db 81h  ;x3         db 252  ;1.2  summ        db 0  diff        db 0  multi       dw 1  all         dw 0  part        db 0  .code  start:      mov ax,@data      mov ds, ax  **;1 часть**      mov ah, 00      mov xi, 68  ;вычитание | mov al, xi      sub al, a      mov diff, al  ;сложение      mov summ, al      mov al, b      add al, summ  ;умножить      mul c      mov multi, ax  ;делить      div d      mov *byte* ptr all, al     mov *byte* ptr part, ah  **;2 часть**      mov ah, 00      mov xi, 81h  ;вычитание      mov al, xi      sub al, a      mov diff, al  ;сложение      mov summ, al      mov al, b      add al, summ  ;умножить      mul c      mov multi, ax | ;делить      div d      mov *byte* ptr all, al     mov *byte* ptr part, ah  **;3 часть**      mov ah, 00      mov xi, 252  ;вычитание      mov al, xi      sub al, a      mov diff, al  ;сложение      mov summ, al      mov al, b      add al, summ  ;умножить      mul c      mov multi, ax  ;делить      mov bl, d      div bx      mov all, ax      mov part, dl        mov ax, 4c00h      int 21  end start  END |

**Результат выполнения программы**

|  |  |
| --- | --- |
| Конец 1 части |  |
| Конец 2 части |  |
| Конец 3 части |  |

**Вычисления вручную:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | x1 | x2 | x3 |
| xi | 68 | 129 | 252 |
| -29 | 39 | 100 | 223 |
| +19 | 58 | 119 | 242 |
| \*13 | 754 | 1547 | 3146 |
| /12 | 62 | 128 | 262 |
| %12 | 10 | 11 | 2 |

**Задание 2 (Вариант 15)**

Вычислить

Распишем формулу по отдельным операциям:

1. АL ← a; значение a в АL
2. АX ← (AL)\*a; a^2 в АХ
3. AX ← (AX)\*x; a^2\*x в AX
4. AX ← (AX)\*x; a^2\*x^2 в AX
5. ВL ← b; значение b в BL
6. ВХ ← (ВL)-2; b-2 в ВХ
7. AX ← (AХ) / (BX); a^2\*x^2/(b-2) в AX
8. CX ← AX; сохранение значения в BX
9. АХ ← a; значение А в регистре АХ
10. АХ ← (AX)\*(АХ); a^2 в АХ
11. AX ← (AX)+x; a^2+x в AX
12. AX ← (AX)/5; (a^2+x)/5 в AX
13. AX ← (CX)-(AX); в AX итоговое значение

|  |  |
| --- | --- |
| Диапазон от 0 до 10 |  |
| .model small  .stack 100h  .data  x           db 1  a           db 2  b           db 3  y           dw 0  .code  start:      mov ax,@data      mov ds, ax      mov ah, 00      mov al, a      mul al      mul x      mul x      mov bl, b      sub bl, 2      idiv bl      mov cx, ax        mov al, a      mul al      add al, x      mov bl, 5      div bl      sub cx, ax      mov y, cx      mov ax, 4c00h      int 21  end start  END |  |

|  |  |
| --- | --- |
|  | mov al,[bp]  mov al,ds:[bp]  mov al,es:[bx]  mov ax,cs:[bx]  ;базовая  mov ax,[bx]+2  mov ax,[bx]+4  mov ax,[bx+2]  mov ax,[4+bx]  mov ax,2+[bx]  mov ax,4+[bx]  mov al,[bx]+2  mov bp,bx  mov ax,[bp+2]  mov ax,ds:[bp]+2  mov ax,ss:[bx+2]  ;индексная  mov si,2  mov ah,B\_TAB[si]  mov al,[B\_TAB+si]  mov bh,[si+B\_TAB]  mov bl,[si]+B\_TAB  mov bx,es:W\_TAB2[si]  mov di,4  mov bl,byte ptr es:W\_TAB2[di]  mov bl,B\_TAB[si]  ;базовая индексная  mov bx,offset B\_TAB  mov al,3[bx][si]  mov ah,[bx+3][si]  mov al,[bx][si+2]  mov ah,[bx+si+2]  mov bp,bx  mov ah,3[bp][si]  mov ax,ds:3[bp][si]  mov ax,word ptr ds:2[bp][si]  mov AX, 4C00h  int 21  end Start  END |

**Вывод:** в ходе выполнения лабораторной работы были приобретены практическое овладение навыками разработки программного кода на языке Ассемблер. Программирование арифметических операций. Практическое освоение основных функций отладчика TD.

Задание 1

.model small

.stack 100h

.data

message     db 'Offset message','$'

;1.1

xi          db 0

a           db 29

b           db 13h

c           db 13

d           db 12

;x1         db 68

;x2         db 81h

;x3         db 252

;1.2

summ            db 0

diff            db 0

multi           dw 1

all             dw 0

part            db 0

.code

start:

    mov ax,@data

    mov ds, ax

;1 часть

    mov ah, 00

    mov xi, 68

;вычитание

    mov al, xi

    sub al, a

    mov diff, al

;сложение

    mov summ, al

    mov al, b

    add al, summ

;умножить

    mul c

    mov multi, ax

;делить

    div d

    mov *byte* ptr all, al

    mov *byte* ptr part, ah

;2 часть

    mov ah, 00

    mov xi, 81h

;вычитание

    mov al, xi

    sub al, a

    mov diff, al

;сложение

    mov summ, al

    mov al, b

    add al, summ

;умножить

    mul c

    mov multi, ax

;делить

    div d

    mov *byte* ptr all, al

    mov *byte* ptr part, ah

;3 часть

    mov ah, 00

    mov xi, 252

;вычитание

    mov al, xi

    sub al, a

    mov diff, al

;сложение

    mov summ, al

    mov al, b

    add al, summ

;умножить

    mul c

    mov multi, ax

;делить

    mov bl, d

    div bx

    mov all, ax

    mov part, dl

;учет переполнения

    ;jnc dop\_sum

    ;adc word ptr l+2, 0

    ;dop\_sum:

    mov ax, 4c00h

    int 21

end start

END