**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

ФАИС

Кафедра «Информатика»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

по дисциплине **«Архитектура вычислительных систем»**

на тему: **«Машинные команды и режимы адресации»**

Выполнил: студент гр. ИП-21

Есис М.С.

Принял: преподаватель В.В. Брель

Дата сдачи отчета: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата допуска к защите: ­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Гомель 2024

**Цель**: Закрепление знаний по режимам адресации и формату машинных команд.

**Задание**

1. Изучить приведенный теоретический материал к лабораторной работе.
2. Разработать программу в EMU86, использующую все режимы адресации.
3. Из объектного файла выписать соответствующие машинные коды и представить разбор по полям: КОП, КОПД, mod, W, r/m и т.д.

4. Выполнить задание по обработке одномерных массивов

**Решение.**

**Программа, использующая все режимы адресации.**

data segment

source dw 13,27,23,2,32,37,1,3,5,90

a dw 5

ends

stack segment

dw 128 dup(0)

ends

code segment

start:

mov cx, 10; непосредственная

mov ax, offset data

mov ds, ax

mov es, ax

MOV AX, SOURCE[si];

MOV AX, SOURCE[bx]

MOV AX, SOURCE[bx+SI];- прямая с индексированием и базированием

MOV AX, [BX] ;косв обычная

MOV aX, [BX+SI] ;косвенная с индексированием

MOV AX, offset a;- непосредственная обычная

MOV AX, [SI+a] ; непосредственная с индексированием

MOV AX, [BX+a];- непосредственная с базированием

mul bx; неявная

mov ax, 4c00h

int 21h

ends

end start

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вид адресации** | **Команда** | **HEX** | **Бинарный код** | **КОП** | **W** | **MOD** | **REG** | **r/m** |
| прямая с индексированием | MOV ax, SOURCE[si] | 8B 04 | 1000101100000100 | 1000101 | 1 | 00 | 000 | 100 |
| прямая с базированием | MOV ax, SOURCE[bx] | 8B 07 | 1000101100000111 | 1000101 | 1 | 00 | 000 | 111 |
| прямая с индексированием и базированием | MOV ax, SOURCE[bx+SI] | 8B 00 | 1000101100000000 | 1000101 | 1 | 00 | 000 | 000 |
| косв обычная | MOV ax, [bx] | 8B 07 | 1000101100000111 | 1000101 | 1 | 00 | 000 | 111 |
| косвенная с индексированием | MOV ax, [bs+si] | 8B 00 | 1000101100000000 | 1000101 | 1 | 00 | 000 | 000 |
| непосредственная обычная | MOV ax, offset a | B8 14 00 | 101110000001010000000000 | 1011100 | 0 | 00 | 010 | 100 |
| непосредственная с индексированием | MOV ax, [si+a] | 8B 44 14 | 100010110100010000010100 | 1000101 | 1 | 01 | 000 | 100 |
| непосредственная с базированием | MOV ax, [si+a] | 8B 47 14 | 100010110100011100010100 | 1000101 | 1 | 01 | 000 | 111 |
| неявная | mul bx | F7 E3 | 1111011111100011 | 1111011 | 1 | 11 | 100 | 011 |
| Регистровая | Mov ds, ax | 8E D8 | 1000111011011000 | 1000111 | 0 | 11 | 011 | 000 |

**Программа согласно варианту:**

Найти среднее арифметическое положительных чисел, стоящих на нечетных местах, и количество чисел, меньших заданного B.

**Листинг программы**

**.model small**

**.stack 100h**

**.data**

**array db 10, 20, 30, 40, 50, 60, 70, 80, 90, 100**

**array\_length equ ($ - array)**

**b db 70**

**.code**

**main proc**

**mov ax, @data**

**mov ds, ax**

**mov cl, 0 ; счетчик чисел меньше B**

**mov si, 0 ; индекс массива**

**mov bl, b ; значение B**

**loop\_check\_b:**

**mov al, array[si]**

**cmp al, bl**

**jnl loop\_increment ; переход к концу цикла, если число >= B**

**inc cl ; инкремент переменной count**

**loop\_increment:**

**inc si ; инкремент индекса массива**

**cmp si, array\_length ; сравнение индекса с размером массива**

**jae loop\_end ; переход к концу цикла, если индекс >= размеру массива**

**jmp loop\_check\_b ; переход к началу цикла**

**loop\_end:**

**; вывод результата**

**mov dl, cl ; сохраняем результат в регистре dl**

**add dl, '0' ; преобразуем в символ**

**mov ah, 02h ; функция вывода символа в консоль**

**int 21h ; вызов прерывания для вывода символа**

**mov ah, 4Ch ; функция завершения программы**

**int 21h ; вызов прерывания для завершения программы**

**main endp**

**end main**

**Вывод:** в результате выполнения работы, были закреплены знания по режимам адресации и формату машинных команд*.*