Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ярославский государственный технический университет»

Кафедра «Информационные системы и технологии»

Отчет по лаб/работе защищен

с оценкой \_\_\_\_\_\_\_\_\_

Преподаватель к.т.н Ю.В. Царёв

«23» марта 2022

**Режимы адресации**

Отчет о лабораторной работе по курсу «Машино-зависимые языки программирования»

ЯГТУ 09.03.04-004 ЛР

|  |  |
| --- | --- |
|  | Отчет выполнил  студент группы ЦПИ-11    Аристов Д.В.  «23» марта 2022 |

2022

Цель работы: разработка простой программы преобразования данных для приобретения практических навыков программирования на языке ассемблера и закрепления знаний по режимам адресации.

Задание:

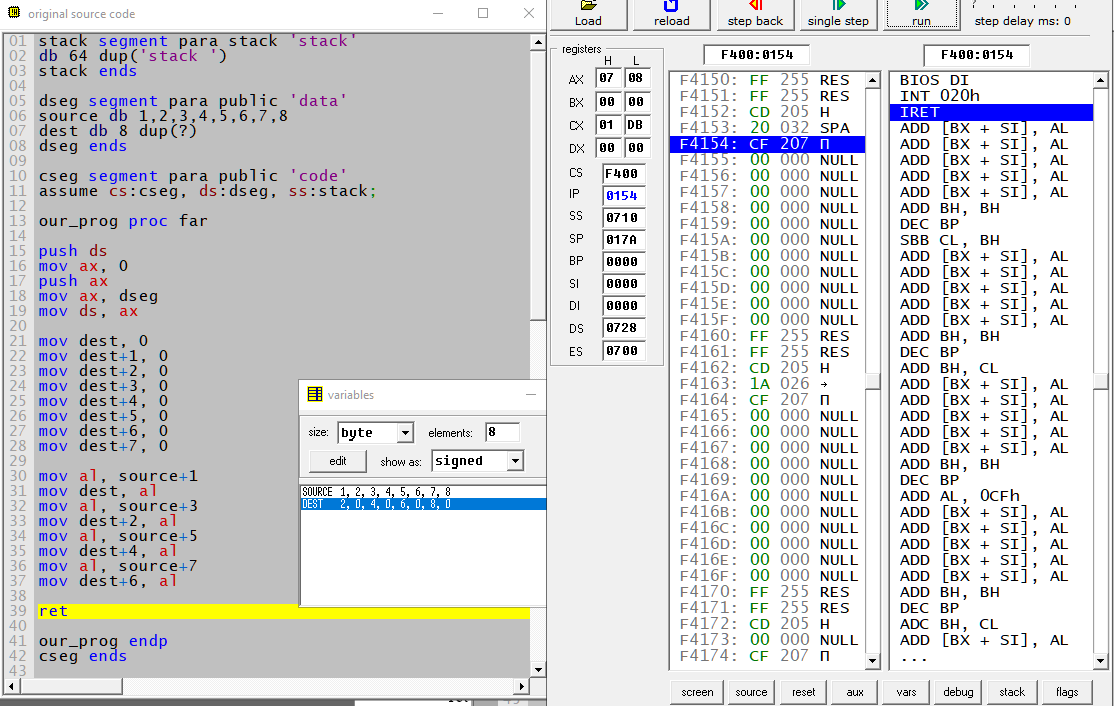
4. Задать одномерный массив, состоящий из Х элементов (Х задается преподавателем из диапазона [7...10]). Заполнить массив константами. Создать новый одномерный массив, поместив в него на место элементов с четными номерами элементы заданного массива с нечетными номерами и обнулив элементы нового массива с нечетными номерами

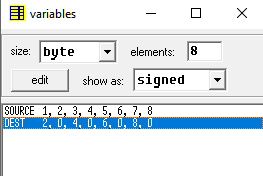
а) элементы массива – однобайтовые;

б) элементы массива – двухбайтовые;

X = 8

Схема выполнения: создаем два одномерных массива с заданным количеством элементов. Для пункта ‘а’ для однобайтовых элементов используем прямую адресацию и перемещаем из массива source элементы под номерами 1,3,5,7 (нечетными) в ячейки с четными номера одномерного массива dest, то есть 0,2,4,6. В пункте ‘б’ используем такой же алгоритм, однако используем косвенную адресацию данных для двухбайтовых элементов: адреса начала массивов записываем в базовый и индексный регистры.

а) 



Текст программы:

stack segment para stack 'stack'

db 64 dup (0)

stack ends

dseg segment para public 'data'

source db 1,2,3,4,5,6,7,8

dest db 8 dup(?)

dseg ends

cseg segment para public 'code'

assume cs: cseg, ds: dseg, ss: stack;

our\_prog proc far

push ds

mov ax, 0

push ax

mov ax, dseg

mov ds, ax

mov dest, 0

mov dest+1, 0

mov dest+2, 0

mov dest+3, 0

mov dest+4, 0

mov dest+5, 0

mov dest+6, 0

mov dest+7, 0

mov al, source+1

mov dest, al

mov al, source+3

mov dest+2, al

mov al, source+5

mov dest+4, al

mov al, source+7

mov dest+6, al

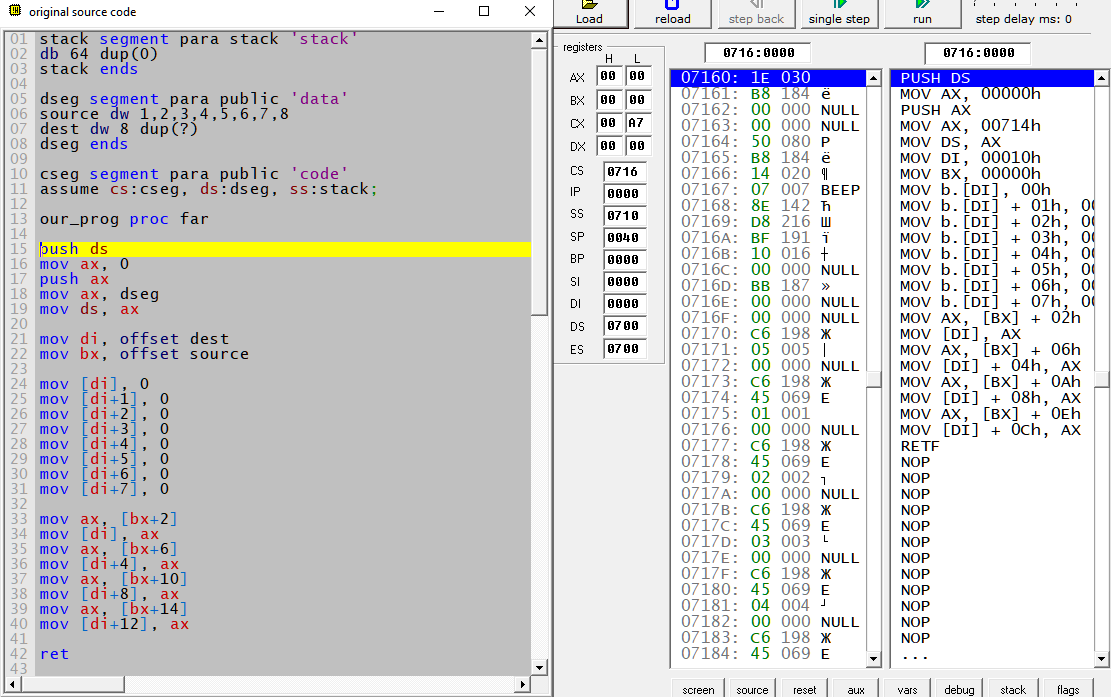
ret

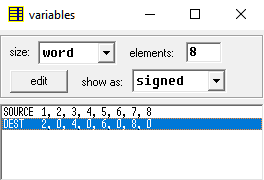
our\_prog endp

cseg ends

end our\_prog

б)





Текст программы:

stack segment para stack 'stack'

db 64 dup (0)

stack ends

dseg segment para public 'data'

source dw 1,2,3,4,5,6,7,8

dest dw 8 dup(?)

dseg ends

cseg segment para public 'code'

assume cs: cseg, ds: dseg, ss: stack;

our\_prog proc far

push ds

mov ax, 0

push ax

mov ax, dseg

mov ds, ax

mov di, offset dest

mov bx, offset source

mov [di], 0

mov [di+1], 0

mov [di+2], 0

mov [di+3], 0

mov [di+4], 0

mov [di+5], 0

mov [di+6], 0

mov [di+7], 0

mov ax, [bx+2]

mov [di], ax

mov ax, [bx+6]

mov [di+4], ax

mov ax, [bx+10]

mov [di+8], ax

mov ax, [bx+14]

mov [di+12], ax

ret

our\_prog endp

cseg ends

end our\_prog

Вывод: с помощью данной лабораторной работы я научился разрабатывать простые программы преобразования данных для того, чтобы приобрести практические навыки программирования на языке ассемблера и закрепить знания по режимам адресации.