МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

#### ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ

БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет автоматики и вычислительной техники

Кафедра радиоэлектронных средств

Отчет по дисциплине

«Цифровые устройства и микропроцессоры»

Лабораторная работа №2

«ПРИНЦИПЫ ВЫПОЛНЕНИЯ КОМАНД ВЕТВЛЕНИЯ, ОРГАНИЗАЦИЯ ЦИКЛОВ И ПОДПРОГРАММ»

Вариант №15

|  |  |  |
| --- | --- | --- |
| Выполнил: студент группы ИНБс– 3301 |  | А.С. Кокшарова |
|  |  |  |
| Проверил: доцент кафедры РЭС |  | М.А. Земцов |

#### Киров 2022

**Цель работы:** изучение принципов выполнения команд ветвления, организации циклов и подпрограмм микропроцессоров с архитектурой x86.

**Ход работы:**

**1**.Исходные данные представлены в таблице 1.

##### Таблица 1 – Исходные данные



**2**.Текст программыпредставлен ниже:

.686

.model flat,stdcall

.stack 100h

.data

;arr dw 0ab7ch,0c58eh,0abcdh ;массив исходных данных

arr dw 0ab7ch,0c53eh,0abc1h ;массив для верификации

len equ ($-arr)/2

m dw 1110100111011111b ;маска для обнуления битов

.code

ExitProcess PROTO STDCALL :DWORD

Start:

xor eax,eax

xor ebx,ebx

xor ecx,ecx

xor edx,edx

lea esi,[arr]

lea edi,[arr]

mov cx,len

mov bx,m

@cycle:

lodsw

and ax,bx

stosw

loop @cycle

lea esi,[arr]

lodsw

mov bx,ax

lodsw

and ax,bx

mov bx,ax ;левая часть в bx

lodsw

not ax

sub bx,ax ;М в BX

mov ax,bx

mov bl,al

and bl,11111111b

jp pp1

jnp pp2

pp1:

clc

sar ax,6

stc

jmp ifc

pp2:

clc

mov bx, 0f1f1h

and ax,bx

jmp ifc

ifc:

jc adr1

jnc adr2

adr1:

add ax,1

jmp exit

adr2:

or ax,1021h

exit:

Invoke ExitProcess,1

End Start

**3**. Расчет вручную.

В цикле обнулить в X, Y, Z биты 3, 5, 6, 10 (результат X',Y',Z')

X=AB7C=1010101101111100

Y=С58E=1100010110001110

Z=ABCD=1010101111001101

Тогда

X`=1010100101011100 = A95C

Y`=1100000110001110 = C18E

Z`=1010100111001101 = A9CD

(X`&Y`)=810C



~(Z`)=5632



M=(X`&Y`) - ~(Z`)=810C-5632=2ADA =10101011011010



мл. М =11011010, количество единиц нечетное, значит переходим к п/п2

R = M&F1F1= 20D0



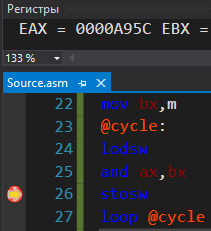
Переноса нет, значит переход к АДР2

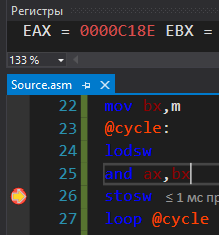
R or 1021 = 30F1

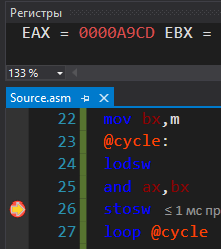


Скриншоты состояния регистров во время выполнения программы представлены ниже:

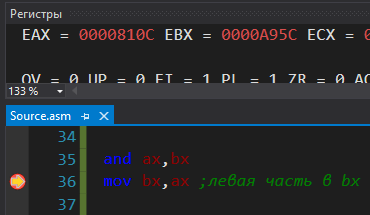
Переменные, измененные в цикле



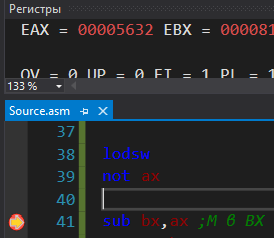




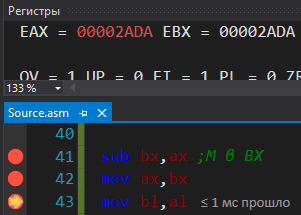
(X`&Y`)=810C



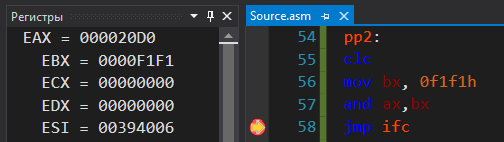
~(Z`)=5632



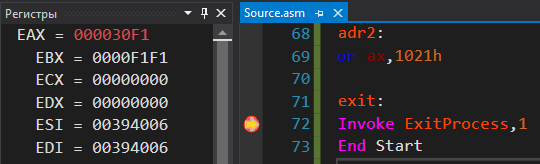
M=(X`&Y`) - ~(Z`)=810C-5632=2ADA =10101011011010



Выполнение п/п2:



Выполнение АДР2:



**Верификация программы:**

Для верификации программы используем другие входные данные:

X=AB7C

Y=C53E

Z=ABC1

X`=A95C

Y`=C11E

Z`=A9C1

(X`&Y`)=811C

~(Z`)=563E

M=(X`&Y`) - ~(Z`)=810C-563E= 2ADE =10101011011110– четное

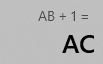
Значит переход к п/п1

R = M(>6>) =AB = 10101011

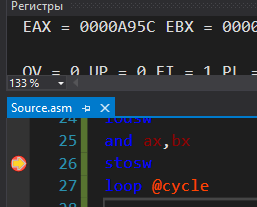


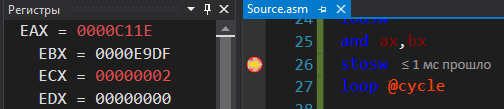
При переносе переход к АДР1:

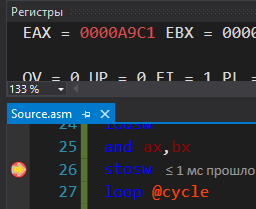
R+1 = AB + 1 = AC



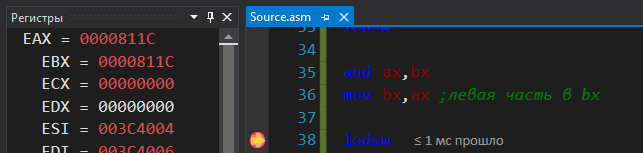
Изменение переменных в цикле:



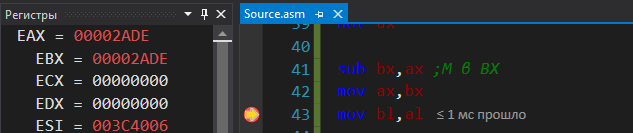




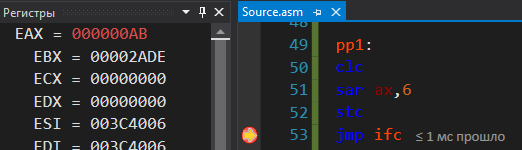
Левая часть выражения:



М в АХ

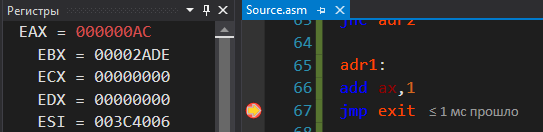


Т.к. количество единиц в младшем байте четное, переход к п/п1



После команды M (>6>) флаг переноса не может быть установлен при любых данных, поэтому для проверки АДР1 установим его вручную командой stc (set carry flag).

Далее, соответственно переход к АДР1



**Вывод:** в ходе лабораторной работы была изучена система команд и способов адресации микропроцессоров с архитектурой х86, получены навыки программирования на языке assembler.