МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

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

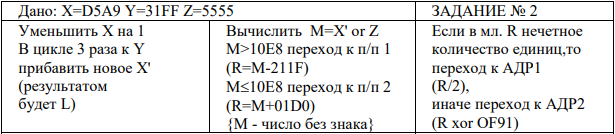
«ПРИНЦИПЫ ВЫПОЛНЕНИЯ КОМАНД ВЕТВЛЕНИЯ,   
ОРГАНИЗАЦИЯ ЦИКЛОВ И ПОДПРОГРАММ»

Вариант 2

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

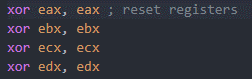
Киров 2024

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



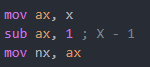
**Ход работы:**

Обнуляем регистры регистры EAX, EBX, ECX, EDX:



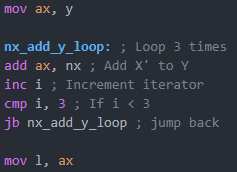


Уменьшаем X на 1:





В цикле 3 раза прибавляем X’ к Y:



Итоговое значение B2F7 записываем в переменную L.

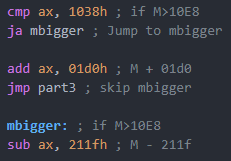
Далее вычисляем M = X’ or Z:





M = D5FD.

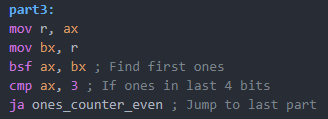
Значение M больше, чем 10E8, значит вычитаем 211F из M:





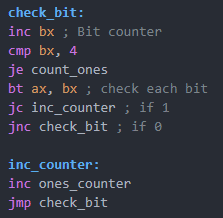
R = B4DE, в двоичном виде: 1011 0100 1101 1110‬. Количество единиц в 4 младших разрядах 3 – нечетное.

Сначала командой BSF находим наименьший разряд с единицей (1011 0100 1101 1110):





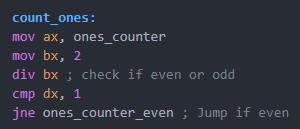
Начиная с этого разряда, командой BT проверяем каждый следующий на наличие единицы, если есть – увеличиваем счетчик.



Соответственно количество единиц:

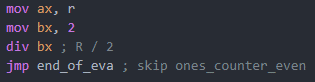


Далее это значение делим на 2 и проверяем на четность:





EDX = 1 – остаток. Число нечётное. Делим R на 2, получаем:





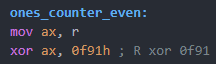
Новое R = DA6F без остатка.



Чтобы получить четное количество единиц, можно заменить переменную X на число 1, тогда X’ = 0. Соответственно M = Z = 5555, R = M – 211F = 3436 = 0011 0100 0011 0110:



Получается четное количество единиц в младших разрядах и выполняется другое условие: R xor 0F91.





Ответ равен 3BA7.

**Код программы:**

.686

.model flat, stdcall

.stack 100h

.data

x dw 1

nx dw ?

y dw 31ffh

z dw 5555h

i dw 0

ones\_counter dw ?

l dw ?

m dw ?

r dw ?

.code

ExitProcess PROTO STDCALL :DWORD

Start:

xor eax, eax ; reset register

xor ebx, ebx

xor ecx, ecx

xor edx, edx

mov ax, x

sub ax, 1 ; X - 1

mov nx, ax

mov ax, y

nx\_add\_y\_loop: ; Loop 3 times

add ax, nx ; Add X' to Y

inc i ; Increment iterator

cmp i, 3 ; If i < 3

jb nx\_add\_y\_loop ; jump back

mov l, ax

mov ax, nx

or ax, z ; M=X' or Z

mov m, ax

cmp ax, 1038h ; if M>10E8

ja mbigger ; Jump to mbigger

add ax, 01d0h ; M + 01d0

jmp part3 ; skip mbigger

mbigger: ; if M>10E8

sub ax, 211fh ; M - 211f

part3:

mov r, ax

mov bx, r

bsf ax, bx ; Find first ones

cmp ax, 3 ; If ones in last 4 bits

ja ones\_counter\_even ; Jump to last part

dec ax

mov bx, ax

mov ax, r

check\_bit:

inc bx ; Bit counter

cmp bx, 4

je count\_ones

bt ax, bx ; check each bit

jc inc\_counter ; if 1

jnc check\_bit ; if 0

inc\_counter:

inc ones\_counter

jmp check\_bit

count\_ones:

mov ax, ones\_counter

mov bx, 2

div bx ; check if even or odd

cmp dx, 1

jne ones\_counter\_even ; Jump if even

mov ax, r

mov bx, 2

div bx ; R / 2

jmp end\_of\_eva ; skip ones\_counter\_even

ones\_counter\_even:

mov ax, r

xor ax, 0f91h ; R xor 0f91

end\_of\_eva:

mov r, ax

exit:

Invoke ExitProcess, r

End Start

GitHub репозиторий: <https://github.com/khivus/DDMLabs>

**Вывод:** были изучены принципы выполнения команд ветвления, организации циклов и подпрограмм микропроцессоров с архитектурой x86.