МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ

БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНВЕРСИТЕТ

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 К.А. Юрлов

Проверил: доцент кафедры РЭС М.А. Земцов

Киров 2024

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

**Ход работы:**

Задание по варианту представлено на рисунке 1.

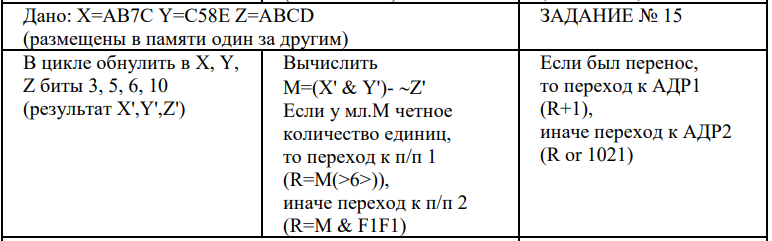


Рисунок 1 – задание по варианту

**Ручной расчет:**

Переводим из шестнадцатеричной в двоичную:

X = AB7C16 = 10101011011111002

Y = C58E16 = 11000101100011102

Z = ABCD16 = 10101011110011012

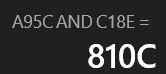
Обнуляем 3, 5, 6, 10 бит:

X’ = 10101001010111002 = A95C16

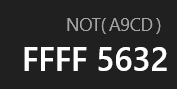
Y’ = 11000001100011102 = C18E16

Z’ = 10101001110011012 = A9CD16­

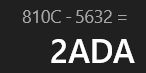
(X’ & Y’) = (A95C & C18E) = 810C



~Z’ = ~(A9CD) = 5632

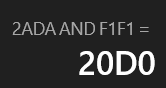


M = 810C – 5632 = 2ADA16 = 101010110110102



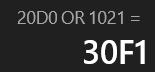
мл. М = 11011010, количество единиц нечетное, переходим к п/п 2

R = M & F1F1 = 2ADA & F1F1 = 20D0



т.к. нет переноса, переходим к АДР2

R or 1021 = 30F1



**Код программы:**

.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

**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

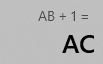
Значит переход к п/п1

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



При переносе переход к АДР1:

R+1 = AB + 1 = AC



GitHub: https://github.com/LLIkoJIbHuk/mpp