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

Отчёт по дисциплине

«Цифровые устройства и микропроцессоры»

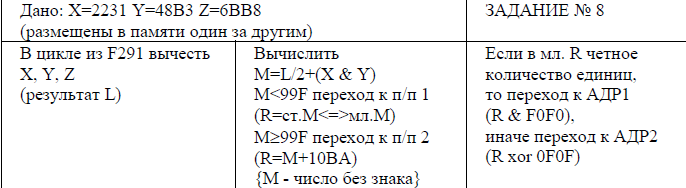
Лабораторная работа №2

Вариант 8

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил: | студент группы ИНБс-3301-01 |  |  | \_\_\_\_\_\_\_\_\_\_\_ |  | К.А. Титов |
|  |  |  |  |  |  |  |
| Проверил: | преподаватель |  |  | \_\_\_\_\_\_\_\_\_\_\_ |  | М.А. Земцов |

Киров 2024

**Цель работы:** написать программу на языке программирования Assembler, решающую уравнение заданное по варианту



Код программы

.model flat, stdcall

.stack 100h

.data

X dw 2231h

Y dw 48B3h

Z dw 6BB8h

L dw ?

R dw ?

;M dw ?

G dw ?

.code

ExitProcess PROTO STDCALL : DWORD

point1 PROC

mov bx, ax

rol bx, 8

mov ax, bx

ret

point1 ENDP

point2 PROC

add ax, 10BAh

jmp check\_even\_count

ret

point2 ENDP

Start:

xor ax, ax

xor dx, dx

xor cx, cx

xor bx, bx

; Загрузка адреса начала массива

lea esi, X

mov cx, 0F291h ; сохраняем значение F291 в регистр CX

; Цикл вычитания X, Y, Z

subtract\_loop:

mov ax, [esi] ; Загрузка очередного элемента в регистр AX

sub cx, ax ; Вычитание F291 из текущего значения

mov [esi], ax ; Сохранение результата обратно в память

add esi, 2 ; Переход к следующему элементу в массиве

cmp esi, OFFSET L ; Проверка, достигли ли конца массива

jne subtract\_loop ; Повторяем цикл для остальных элементов

mov L, cx ; Сохраняем результат в L

xor cx, cx

; Вычисление M

mov ax, X

and ax, Y

mov bx, L

shr bx, 1

add ax, bx

;Альтернативная ветвь

;sub ax, 07D0h

xor bx, bx

; Проверка условия M < 99F

cmp ax, 099Fh

jb label1 ; Если M < 99F, переходим к п/п 1

; Если M >= 99F, переходим к п/п 2

label2:

call point2

label1:

; Перестановка старших и младших байтов M

call point1

check\_even\_count:

; Подсчет количества единиц в младшем байте R

mov bx, ax

and bx, 00FFh ; Оставляем только младший байт

xor cx, cx ; Счетчик единиц

count\_bits\_loop:

test bx, 1 ; Проверяем младший бит

jz bit\_is\_zero ; Если бит равен 0, пропускаем инкремент счетчика

inc cx ; Инкремент счетчика

bit\_is\_zero:

shr bx, 1 ; Сдвигаем регистр вправо

jnz count\_bits\_loop ; Повторяем цикл, пока регистр не станет нулевым

; Проверка четности счетчика единиц

test cx, 1

jz adr1 ; Если четное количество единиц, переходим к АДР1

; Иначе, переходим к АДР2

xor ax, 0F0Fh

jmp finish

adr1:

and ax, 0F0F0h

finish:

mov R, ax

INVOKE ExitProcess, R

END Start

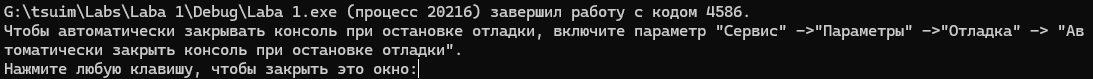


Рисунок 1 – результат вычислений

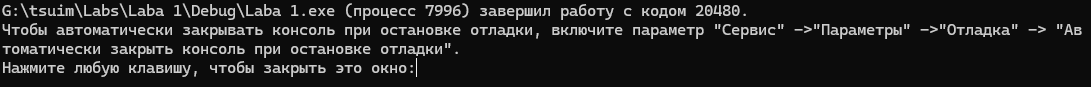


Рисунок 1 – результат вычислений альтернативной ветви

**Валидация:**

1. **Вычитание из F291 X, Y, Z:**

**L=F291-2231-48B3-6BB8=1BF5**

1. **Вычисление M:**

**M=1BF5/2+(2231&48B3)=DFA+31=E2B**

1. **Вычисление R:**

**R=E2B+10BA=** **1EE5**

1. **Вычисление результата программы**

**R xor 0F0F = 11EA = 4586**

**Валидация альтернативной ветви:**

1. **Вычитание из F291 X, Y, Z:**

**L=F291-2231-48B3-6BB8=1BF5**

1. **Вычисление M:**

**M=1BF5/2+(2231&48B3)=DFA+31=E2B**

1. **Для реализации альтернативной ветви вычитаем из М 07D0:**

**E2B – 07D0 = 65B**

1. **Замена старших и младших битов R:**

**R=5B06**

1. **Вычисление результата программы**

**R &F0F0 = 5000 = 20480**

Результаты полученные вручную и значения полученные в результате выполнения программы совпадают, следовательно программа работает исправно.

<https://github.com/dendrokit/TSUIM.git>