Министерство высшего образования и науки РФ

Вятский государственный университет

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра радиоэлектронных средств

Отчёт по предмету «Цифровые устройства и микропроцессоры» Лабораторная работа № 2

«Принципы выполнения команд ветвления, организация циклов и подпрограмм»

Вариант 10

Выполнил: студент группы ИНБб-3301-02-00 Кощеев Д.М.

Проверил: преподаватель Земцов М. А.

Киров 2025

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

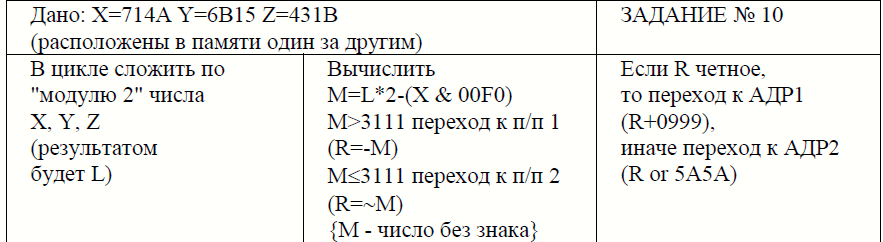


Таблица 1 – Выражение, которое надо реализовать на Ассемблере

Код программы:

.686

.model flat,stdcall

.stack 100h

.data

X dd 714Ah ; Значение X

Y dd 6B15h ; Значение Y

Z dd 431Bh ; Значение Z

L dd ? ; Переменная для результата сложения

M dd ? ; Переменная для результата вычислений

R dd ? ; Переменная для результата подпрограмм

TEMP dd ? ; Временная переменная

.code

ExitProcess PROTO STDCALL :DWORD

Start PROC

; 1. Сложение по модулю 2 (X XOR Y XOR Z)

MOV EAX, X ; EAX = X

XOR EAX, Y ; EAX = X XOR Y

XOR EAX, Z ; EAX = X XOR Y XOR Z

MOV L, EAX ; L = EAX (результат сложения)

; 2. Вычисление M

MOV EAX, L ; EAX = L

SHL EAX, 1 ; EAX = L \* 2

MOV EBX, X ; EBX = X

AND EBX, 00F0h ; EBX = X & 00F0 (маска для получения 4-х бит)

SUB EAX, EBX ; EAX = AX - BX

MOV M, EAX ; M = AX

; 3. Проверка условия M > 3111 (беззнаковое сравнение)

MOV EAX, M ; EAX = M

CMP EAX, 3111h ; Сравнение M с 3111

JA CallSubP1 ; Если M > 3111, переход к п/п 1

JLE CallSubP2 ; Иначе переход к п/п 2

CallSubP1:

;Сохраняем EAX

PUSH EAX

Call SubP1

POP EAX

JMP CheckEven ; Переход к проверке четности R

CallSubP2:

PUSH EAX

Call SubP2

POP EAX

CheckEven:

MOV EAX, R

AND EAX, 1 ; Проверяем младший бит (четное или нечетное)

JZ Addr1Jump ; Переход, если R четное (младший бит = 0)

JMP Addr2Jump ; Переход, если R нечетное (младший бит = 1)

Addr1Jump:

; R = R + 0999h

MOV EAX, R

ADD EAX, 0999h

; здесь код, который выполняется по адресу ADR1 (можно положить в EAX результат операции)

MOV TEMP, EAX

JMP EndProgram

Addr2Jump:

; R = R OR 5A5Ah

MOV EAX, R

OR EAX, 5A5Ah

; здесь код, который выполняется по адресу ADR2 (можно положить в EAX результат операции)

MOV TEMP, EAX

JMP EndProgram

EndProgram:

; Завершение программы

exit:

Invoke ExitProcess, 0

Start ENDP

SubP1 PROC

; R = -M

MOV EAX, M

NEG EAX

MOV R, EAX

RET

SubP1 ENDP

SubP2 PROC

; R = ~M

MOV EAX, M

NOT EAX

MOV R, EAX

RET

SubP2 ENDP

END Start

* X = 714Ah = 29002
* Y = 6B15h = 27413
* Z = 431Bh = 17179

L = X XOR Y XOR Z = 5944 = 22852

X & 00F0 = 40 = 64

L\*2 = B288 = 45704

M = L\*2 – (X&00F0) = B248 = 45640

4294921656 = FFFF 4DB8 = -45640

4294924113 = FFFF5751 = -43183

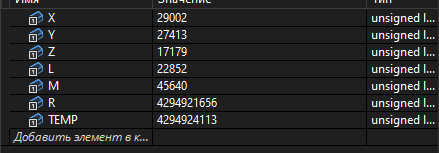


Рисунок 1 –Результаты решения

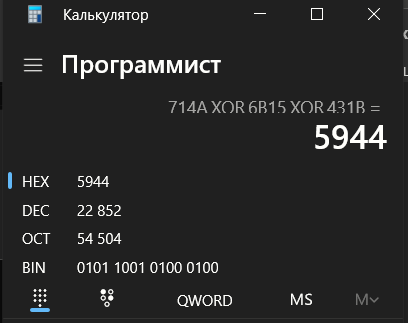


Рисунок 2 – Значение L (XOR значения A, Y, Z)

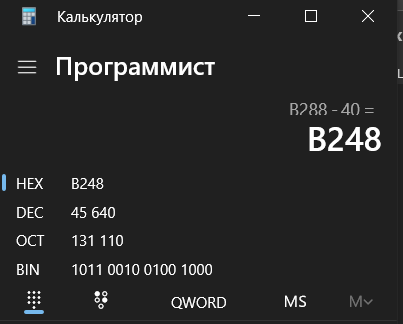


Рисунок 3 – Значение M (2\*L – X&00F0)

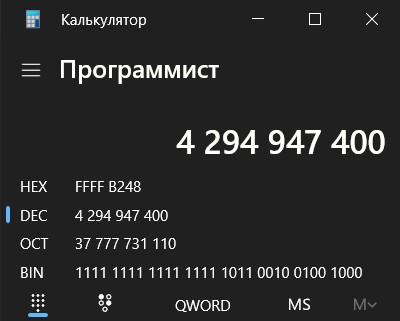


Рисунок 4 – Значение R (отрицаем значение M)

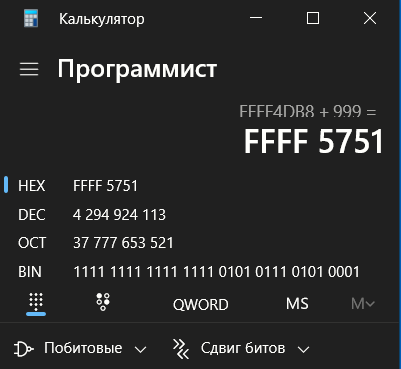


Рисунок 5 – Сумма R и 0999

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