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-16

Принципы выполнения команд ветвления, организация циклов и подпрограмм

| Выполнил: | студент группы ИНБб-31 |  | \_\_\_\_\_\_\_\_\_\_\_ |  | А.И. Парфенов |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Проверил: | доцент кафедры РЭС |  | \_\_\_\_\_\_\_\_\_\_\_ |  | М. А. Земцов |

Киров 2023

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

**Исходные данные**:

| Дано: X = FC6A; Y = F639; Z = 6132 | | |
| --- | --- | --- |
| 1) В цикле у Х установить в 1 биты 3, 7, 11  у Y сбросить биты 1, 6, 11  (результат Х' и Y') | 2) Вычислить  M=(X'-Z) & (Y'-Z)  Если М отрицательное,  то п/п 1 (R=M+123),  Если М положительное,  то п/п 2 (R=M-999) | 3) Если R<=09,  то переход к АДР1  (R(<4c<)),  при R>09 переход к АДР2  (R or 1001)  {R - число со знаком} |

**Ход работы:**

1. Для выполнения 1-го задания необходимо организовать цикл, в первой итерации которого будет получено Y’, а во второй X’. Для сброса битов 1, 6, 11 необходимо произвести операцию AND с маской, в которой данные биты равны 0 (1111 0111 1011 1101). Для установления соответствующих битов в единицу необходимо произвести операцию OR с маской (0000 1000 1000 1000). На рисунках 1-2 приведен код цикла на ассемблере и контрольные значения после выполнения( число X - в регистре ax, Y - в bx).

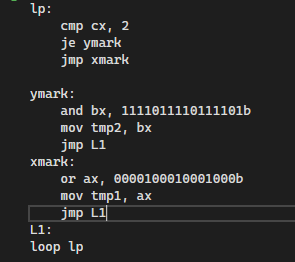


Рисунок 1 - Цикл

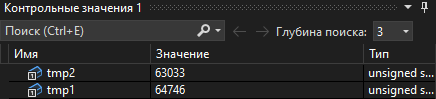


Рисунок 2 - Контрольные значения

Переменная tmp1 - это значения X’, а tmp2 - Y’. Проверку можно осуществить в калькуляторе (рисунок 3).

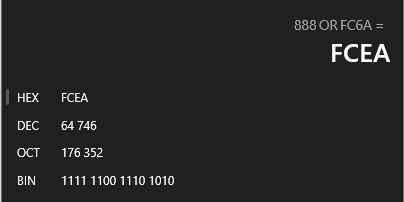


Рисунок 3 - Проверка полученных значений

Как видно, полученные значения совпали.

1. Для выполнения второго задания сначала вычисляется число М и затем происходит переход в зависимости от того, отрицательное или положительное число М. Для организации ветвления используется команда сравнения CMP и команды условных переходов по результату сравнения. Если число M < 0, то выполняется переход к подпрограмме 1 и вычисляется R, иначе происходит переход к подпрограмме 2. На рисунке 4 представлен код на ассемблере и контрольные значения М и R.

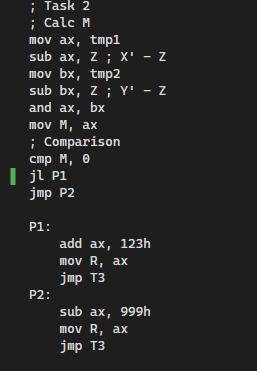


Рисунок 4 - Код и контрольные значения

Проверку можно осуществить в калькуляторе (рисунок 5):

(FCEA - 6132) and (F639 - 6132) = 910016 (37120);

М отрицательное, следовательно R = M + 123= 9100 + 123 = 922316 (37 411)

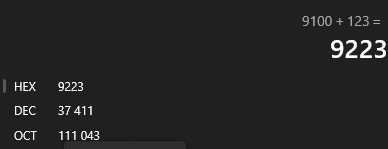
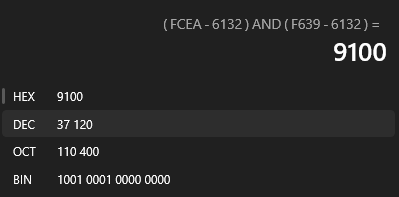


Рисунок 5 - Проверка полученных значений

Полученные значения совпали.

3) В задании 3 необходимо выполнить сравнение числа R с константой 09 и выполнить переход при выполнении одного из условий по адресу. Так как число R рассматривается как число со знаком, то полученное во втором задании число 8767 является отрицательным (старший бит равен единице). Поэтому будет осуществлен переход по адресу 1. На рисунке 6 представлен код для задания 3.

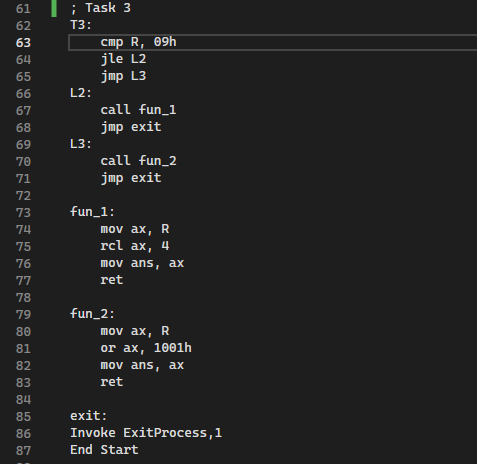


Рисунок 6 - Код для задания 3

4) Для того, чтобы выполнялись противоположные условия в заданиях 1 и 2 можно использовать значение X равное 700016, тогда X’ = 788816 (30856), М = (7888 - 6132) AND (F639 - 6132) = 150616 (5382) - положительное, следовательно выполняется подпрограмма 2 и R = 1506 - 999 = B6D16 (2925) тоже положительное и R > 09, и переход выполняется по адресу 2. На рисунке 7 представлены контрольные значения.

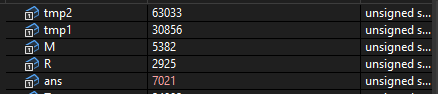


Рисунок 7 - Контрольные значения

**Вывод:** в ходе выполнения лабораторной работы были изучены принципы ветвления, организации циклов и подпрограмм микропроцессоров с архитектурой x86.

**Код программы:**

.686

.model flat,stdcall

.stack 100h

.data

; X new 7000h

;X dw 0FC6Ah

;Y dw 0F639h

Z dw 6132h

M dw ?

R dw ?

ans dw ?

tmp1 dw ? ; X'

tmp2 dw ? ; Y'

arr dw 0FC6Ah, 0F639h

.code

ExitProcess PROTO STDCALL :DWORD

Start:

mov eax, 0

mov ebx, 0

mov ecx, 2

mov ax, arr[0]

mov bx, arr[0 + 2]

; Task 1

lp:

cmp cx, 2

je ymark

jmp xmark

ymark:

and bx, 1111011110111101b

mov tmp2, bx

jmp L1

xmark:

or ax, 0000100010001000b

mov tmp1, ax

jmp L1

L1:

loop lp

; Task 2

; Calc M

mov ax, tmp1

sub ax, Z ; X' - Z

mov bx, tmp2

sub bx, Z ; Y' - Z

and ax, bx

mov M, ax

; Comparison

cmp M, 0

jl P1

jmp P2

P1:

add ax, 123h

mov R, ax

jmp T3

P2:

sub ax, 999h

mov R, ax

jmp T3

; Task 3

T3:

cmp R, 09h

jle L2

jmp L3

L2:

call fun\_1

jmp exit

L3:

call fun\_2

jmp exit

fun\_1:

mov ax, R

rcl ax, 4

mov ans, ax

ret

fun\_2:

mov ax, R

or ax, 1001h

mov ans, ax

ret

exit:

Invoke ExitProcess,1

End Start