Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра информатики

Отчет по лабораторной работе №1

Основные конструкции. Написание простейшей программы

Выполнила: Студент: гр. 053502

Юрьев Вадим Андреевич

Руководитель: ст. преподаватель

Шиманский В.В.

Минск 2021

СОДЕРЖАНИЕ

1. Введение
2. Постановка задачи
3. Программная реализация
4. Выводы

Литература

Приложение

1. Введение

Целью данной работы является изучить следующий материал:

1) Регистры процессора 8086.

2) Логика работы команд MOV, ADD, SUB, MUL, DIV.

3) Логические операции AND, OR, XOR, NOT.

4) Команды сдвига SHL и SHR.

5) Работа команд CMP и TEST.

6) Последовательное выполнение команд. Назначение регистра IP.

7) Логика работы следующих команд условного и безусловного переходов: JMP, JE, JNE, JC, JNC.

Назначение флагов CF и ZF. Использование меток.

8) Размещение данных в сегменте данных. Размерность данных: DB, DW, DD. Работа с переменными, определенными в сегменте данных.

9) Компилирование, линковка, выполнение и отладка ассемблерных программ.

1. Постановка задачи
   1. Текст задания

Если a = c ^ 2 то  
      Результат = с / (a \* b) + d  
  Иначе  
      Если  c - b <> a ^ 2 + b ^ 3 то  
       Результат = a AND (b - c)  
    Иначе  
       Результат = a/b/c+(b^2+c^3)/d

* 1. Условие задания

В каждом из заданий переменные a, b, c, d определяются в сегменте данных

и имеют размерность слово. Необходимо выполнить над ними заданные

арифметические и логические операции, а результат поместить в регистр AX.

При выполнении умножения считаем, что результат вмещается в слово. При

выполнении деления считаем, что оно целочисленное. Выполнение

программы необходимо показать в Отладчике.

1. Программная реализация

3.1. Значения переменных устанавливаются при объявлении сегмента данных. Программа разбита при помощи меток на несколько логических частей, каждая из которых выполняет определенную ветку условия.

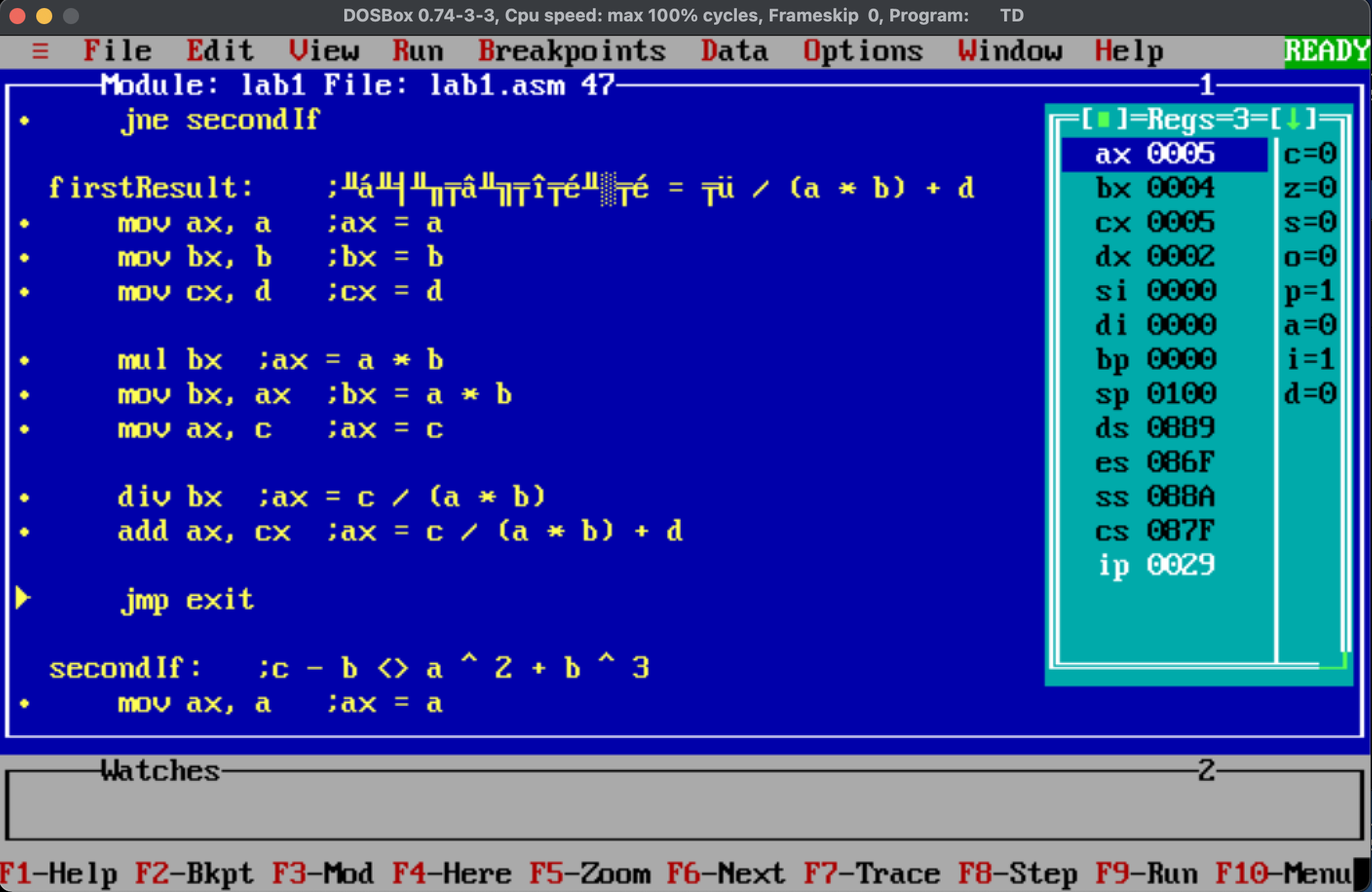
* 1. Результат можно видеть в отладчике в регистре AX
  2. Примеры:

3.3.1 Отладка ветки №1

Тест для ветки

« Результат = с / (a \* b) + d »

a = 4 b = 1 c = 2 d = 5



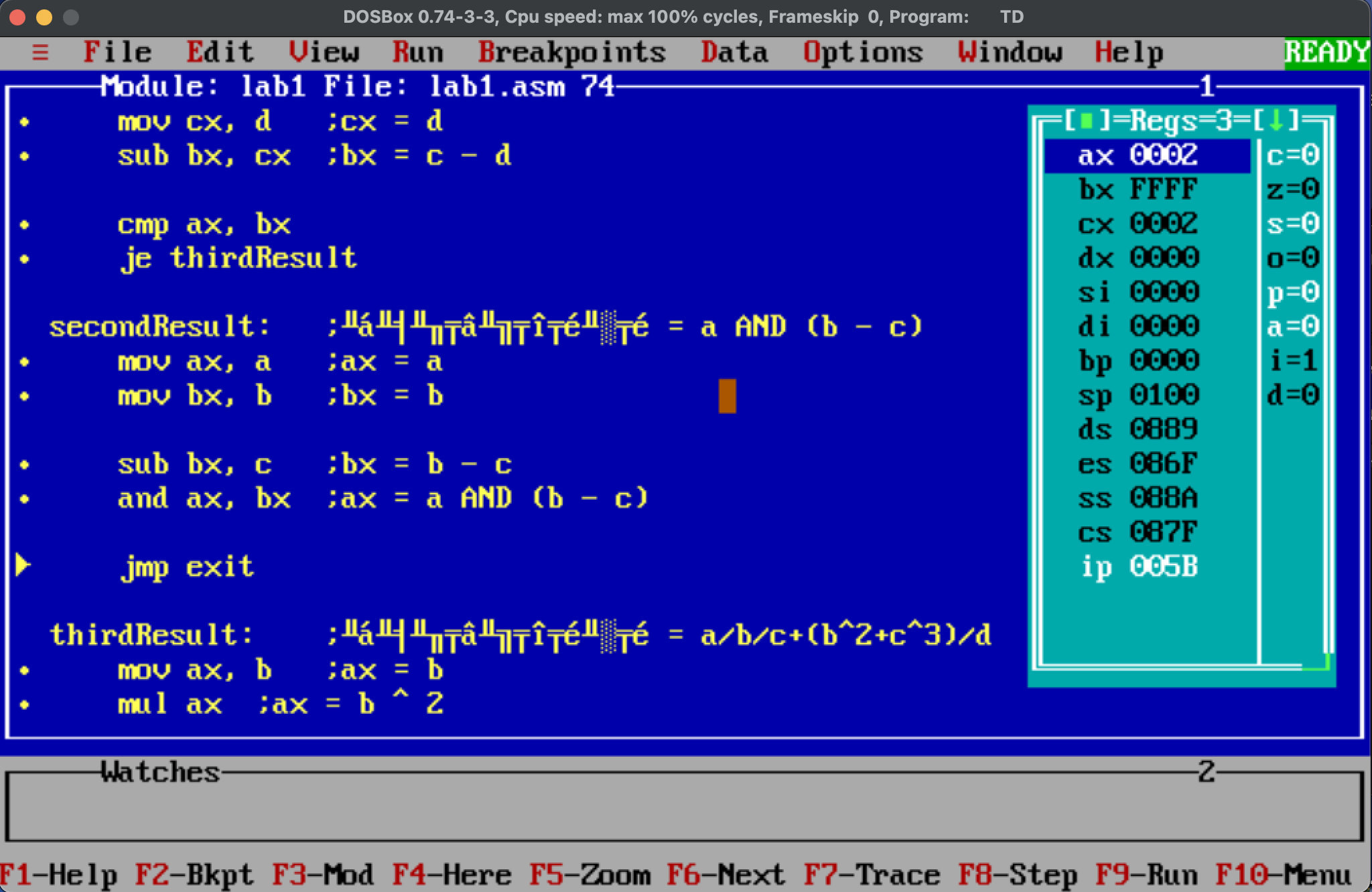
Ответ: 5

3.3.2 Отладка ветки №2

Тест для ветки

« Результат = a AND (b - c) »

a = 2 b = 3 c = 4 d = 2



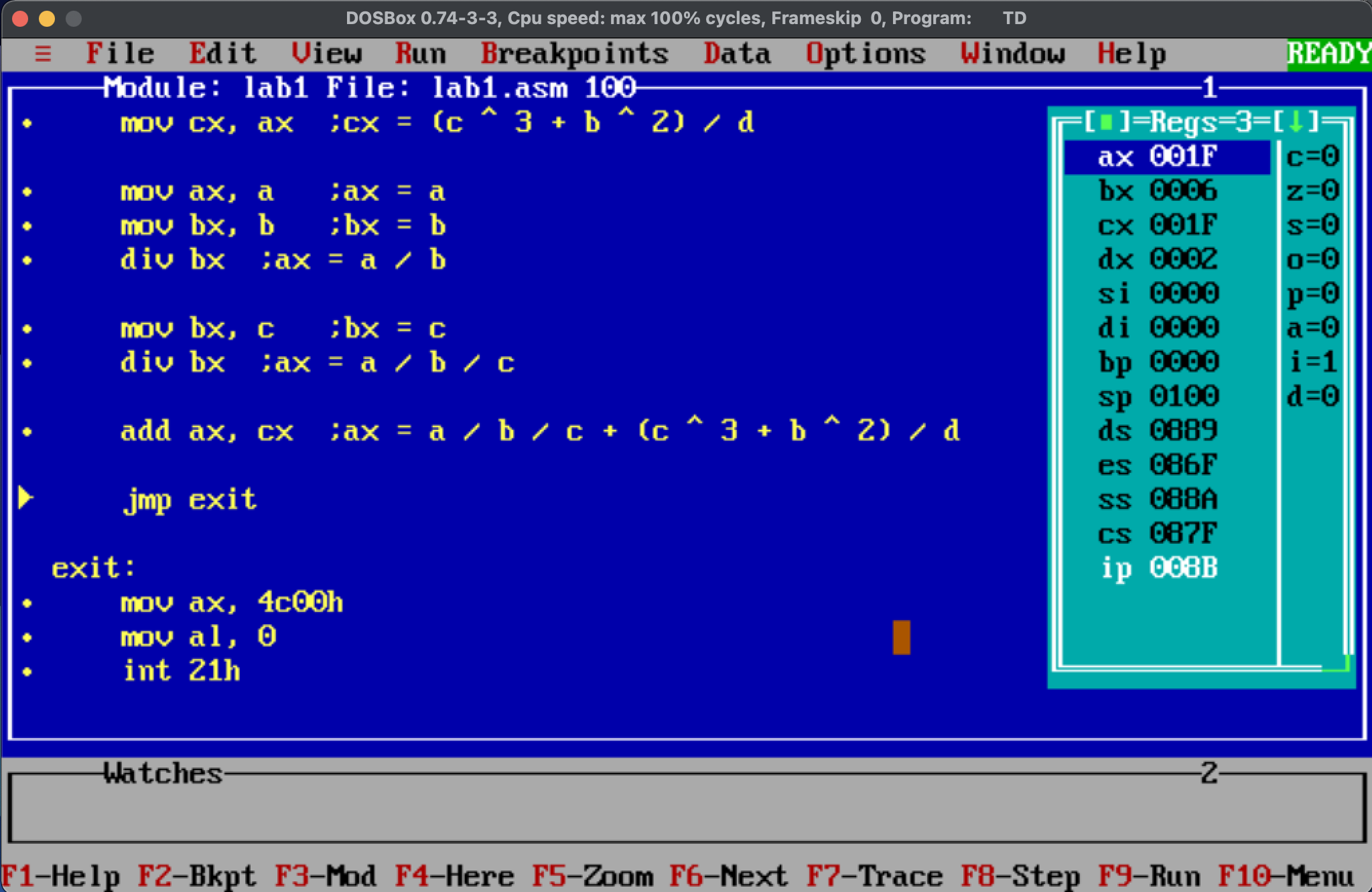
Ответ:2

3.3.3 Отладка ветки №3

Тест для ветки

« Результат = a/b/c+(b^2+c^3)/d»

a = 2 b = 1 c = 6 d = 7



Ответ: 001F16 = 3110

1. Выводы

На практике было изучено и опробовано, в соответствии с поставленной задачей: регистры процессора 8086, логика команд MOV, ADD, SUB, MUL, DIV, логические операции AND, OR, XOR, NOT, работа команды CMP, использование меток, логика работы команд условного и безусловного переходов JMP, JE, JNE, JC, JNC, размещение данных в сегменте данных, размерность данных: DB, DW, DD.

В процессе выполнения лабораторной работы мной были освоены DosBox, TurboDebuger. С помощью TurboDebuger вы можете отслеживать ход выполнения работы программы.

Литература

1. Юров В.И. – «Assembler. Учебник для вузов. 2-ое издание, 2003 год».

2. Юров В.И. – «Assembler. Практикум. 2-ое издание, 2006 год».

3. Калашников О.А. – «Ассемблер - это просто. 2-ое издание, 2011 год».

Приложение

.model small

.stack 256

.data

a dw 2

b dw 1

c dw 6

d dw 7

.code

start:

mov ax, @data

mov ds, ax

firstIf: ;a = c ^ 2

mov ax, c ;ax = c

mul ax ;ax = c ^ 2

mov cx, ax ;cx = c ^ 2

mov ax, a ;ax = a

cmp ax, cx

jne secondIf

firstResult: ;Результат = с / (a \* b) + d

mov ax, a ;ax = a

mov bx, b ;bx = b

mov cx, d ;cx = d

mul bx ;ax = a \* b

mov bx, ax ;bx = a \* b

mov ax, c ;ax = c

div bx ;ax = c / (a \* b)

add ax, cx ;ax = c / (a \* b) + d

jmp exit

secondIf: ;c - b <> a ^ 2 + b ^ 3

mov ax, a ;ax = a

mul ax ;ax = a ^ 2

mov cx, ax ;cx = a ^ 2

mov bx, b ;bx = b

mov ax, b ;ax = b

mul ax ;ax = b ^ 2

mul bx ;ax = b ^ 3

add ax, cx ;ax = a ^ 2 + b ^ 3

mov bx, c ;bx = c

mov cx, b ;cx = b

sub bx, cx ;bx = c - b

cmp ax, bx

je thirdResult

secondResult: ;Результат = a AND (b - c)

mov ax, a ;ax = a

mov bx, b ;bx = b

sub bx, c ;bx = b - c

and ax, bx ;ax = a AND (b - c)

jmp exit

thirdResult: ;Результат = a/b/c+(b^2+c^3)/d

mov ax, b ;ax = b

mul ax ;ax = b ^ 2

mov bx, ax ;bx = b ^ 2

mov ax, c ;ax = c

mov cx, c ;cx = c

mul ax ;ax = c ^ 2

mul cx ;ax = c ^ 3

add ax, bx ;ax = c ^ 3 + b ^ 2

mov bx, d ;bx = d

div bx ;ax = (c ^ 3 + b ^ 2) / d

mov cx, ax ;cx = (c ^ 3 + b ^ 2) / d

mov ax, a ;ax = a

mov bx, b ;bx = b

div bx ;ax = a / b

mov bx, c ;bx = c

div bx ;ax = a / b / c

add ax, cx ;ax = a / b / c + (c ^ 3 + b ^ 2) / d

jmp exit

exit:

mov ax, 4c00h

mov al, 0

int 21h

end start