«Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчёт защищён с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель Боровцов Е.Г.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Отчёт

Лабораторной работе №6

«Введение в архитектуру IBM PC»

Студент группы ПИ 92 В.М. Шульпов

Преподаватель доцент, к.т. н. Боровцов Е.Г.

Барнаул 2021

**Цель лабораторной работы:**

Данная лабораторная работа посвящена знакомству с архитектурой и системой

команд процессоров семейства Intel80x86.

**Задание к лабораторной работе:**

- ознакомиться с базовой моделью программирования процессоров семейства

Intel80x86, распределением адресного пространства, системой команд, методами

адресации;

- ознакомиться с системой команд и использованием системного отладчика Debug

(см. Приложение 4);

- для заданного варианта задания реализовать программы, используя систему

команд семейства процессоров Intel80x86;

- используя отладчик, ввести программы в память компьютера и выполнить их в

непрерывном и пошаговом режиме, наблюдая результаты выполнения команд по

содержимому регистров процессора и оперативной памяти;

- оформить отчет по лабораторной работе.

Задание:

Y:= F(X,Y) + F(Y,X) - F(2,3), где F(A,B)= (B2-1) + (A+B)2;

|  |  |
| --- | --- |
| 27 | Программные прерывания на ввод/вывод  1. апп. прерывание – изменение знаков “-” на “+” и “+” на “-” в алгоритме подпрограммы и рестарт программы с очисткой стека и рабочих переменных;  2. апп. прерывание – вывод текущего значения регистров F и SP и продолжение работы программы. |

**Мнемокод:**

jmp begin ; безусловный переход на начало программы

.org 2 ; зарезервировать соотвествующие ячейки

.db 080 ; программных прерываний

.org 3 ;

.db 090 ;

.org 8 ;

.db 0A5 ;

.org 9 ;

.db 0E0 ;

begin: mvi a,0 ;

stor a,FLAG ; флаг использования функции (с измененными знаками или нет)

mvi a,2 ; кол-во параметров для ввода

mvi b,X ;

int 2 ; вызов программного прерывания ввода

begin2: load a,X ; начало программы, где X,Y уже введены

load b,Y ;

call FUNC ;

push a ;

load a,Y ;

load b,X ;

call FUNC ;

pop b ;

adr a,b ;

push a ;

mvi a,2 ;

mvi b,3 ;

call FUNC ;

pop b ;

xchg ;

sbr a,b ;

int 3 ; вызов программного прерывания вывода

stop ;

X: .ds 1 ;

Y: .ds 1 ;

FLAG: .ds 1 ;

FUNC: push b ; ФУНКЦИЯ ПОДСЧЕТ F(A,B)

; на вход - регистры rA, rB

; на выход - rA

push a ;

load a,FLAG; ;

cmi a, 0 ; a-0 результат в рег. признаков

jz FUNC2 ; переход по нулю на алгоритм с неизмененными знаками

FUNC1:pop a ; алгоритм с измененными знаками

sbr a,b ; \*

xchg ;

mur a,b ;

pop a ;

push b ;

xchg ;

mur a,b ;

adi b,1 ; \*

pop a ;

sbr a,b ; \*

jmp endfunc ; безусловный переход на выход из подпрограммы

FUNC2:pop a ; алгоритм с неизмененными знаками

adr a,b ; \*

xchg ;

mur a,b ;

pop a ;

push b ;

xchg ;

mur a,b ;

sbi b,1 ; \*

pop a ;

adr a,b ; \*

endfunc:ret ;

.org 080 ; 1 ПРОГРАММНОЕ ПРЕРЫВАНИЕ (ВВОД)

di ;

CYCLIN: push a ;

in 0 ;

stor a,00(b) ;

adi b,1 ;

pop a ;

loop a,CYCLIN ;

ei ;

rin ;

.org 090 ; 2 ПРОГРАММНОЕ ПРЕРЫВАНИЕ (ВЫВОД)

di ;

out 2 ;

ei ;

rin ;

.org 0A5 ; 1 АППАРАТНОЕ ПРЕРЫВАНИЕ

; прерывание – изменение знаков “-” на “+”

; и “+” на “-” в алгоритме подпрограммы

; и рестарт программы с очисткой стека и рабочих переменных

di ;

mvi a,1 ;

stor a,FLAG ; флаг для выбора алгоритма

exit1: mvi sp, 0fd ; почистить стек

mvi a, begin2 ;

stor a, 01(sp) ; в sp команду из начала программы

ei ;

rin ;

.org 0E0 ; 2 АППАРАТНОЕ ПРЕРЫВАНИЕ

di ;

load a,01(sp) ; загрузка SP (счётчика)

out 6 ;

load a,00(sp) ; загрузка F (флагов/признаков)

out 7 ;

ei ;

rin ;

Listing

1 00 b8 0a jmp begin ; безусловный переход на начало программы

2 02 00 .org 2 ; зарезервировать соотвествующие ячейки

3 02 80 .db 080 ; программных прерываний

4 03 00 .org 3 ;

5 03 90 .db 090 ;

6 04 00 .org 8 ;

7 08 a5 .db 0A5 ;

8 09 00 .org 9 ;

9 09 e0 .db 0E0 ;

11 0a 30 00 begin: mvi a,0 ;

12 0c 18 30 stor a,FLAG ; флаг использования функции (с измененными знаками

13 0e 30 02 mvi a,2 ; кол-во параметров для ввода

14 10 31 2e mvi b,X ;

15 12 d2 int 2 ; вызов программного прерывания ввода

16 13 10 2e begin2: load a,X ; начало программы, где X,Y уже введены

17 15 11 2f load b,Y ;

18 17 c0 31 call FUNC ;

19 19 20 push a ;

20 1a 10 2f load a,Y ;

21 1c 11 2e load b,X ;

22 1e c0 31 call FUNC ;

23 20 29 pop b ;

24 21 51 adr a,b ;

25 22 20 push a ;

26 23 30 02 mvi a,2 ;

27 25 31 03 mvi b,3 ;

28 27 c0 31 call FUNC ;

29 29 29 pop b ;

30 2a 38 xchg ;

31 2b 69 sbr a,b ;

32 2c d3 int 3 ; вызов программного прерывания вывода

33 2d f8 stop ;

35 2e 00 X: .ds 1 ;

36 2f 00 Y: .ds 1 ;

37 30 00 FLAG: .ds 1 ;

39 31 21 FUNC: push b ; ФУНКЦИЯ ПОДСЧЕТ F(A,B)

40 32 00 ; на вход - регистры rA, rB

41 32 00 ; на выход - rA

42 32 20 push a ;

43 33 10 30 load a,FLAG; ;

44 35 b0 00 cmi a, 0 ; a-0 результат в рег. признаков

45 37 ba 47 jz FUNC2 ; переход по нулю на алгоритм с неизмененными знаками

46 39 28 FUNC1:pop a ; алгоритм с измененными знаками

47 3a 69 sbr a,b ; \*

48 3b 38 xchg ;

49 3c 81 mur a,b ;

50 3d 28 pop a ;

51 3e 21 push b ;

52 3f 38 xchg ;

53 40 81 mur a,b ;

54 41 61 01 adi b,1 ; \*

55 43 28 pop a ;

56 44 69 sbr a,b ; \*

57 45 b8 53 jmp endfunc ; безусловный переход на выход из подпрограммы

58 47 28 FUNC2:pop a ; алгоритм с неизмененными знаками

59 48 51 adr a,b ; \*

60 49 38 xchg ;

61 4a 81 mur a,b ;

62 4b 28 pop a ;

63 4c 21 push b ;

64 4d 38 xchg ;

65 4e 81 mur a,b ;

66 4f 79 01 sbi b,1 ; \*

67 51 28 pop a ;

68 52 51 adr a,b ; \*

69 53 c8 endfunc:ret ;

71 54 00 .org 080 ; 1 ПРОГРАММНОЕ ПРЕРЫВАНИЕ (ВВОД)

72 80 f0 di ;

73 81 20 CYCLIN: push a ;

74 82 40 in 0 ;

75 83 1c 40 stor a,00(b) ;

76 85 61 01 adi b,1 ;

77 87 28 pop a ;

78 88 e0 81 loop a,CYCLIN ;

79 8a e8 ei ;

80 8b d8 rin ;

82 8c 00 .org 090 ; 2 ПРОГРАММНОЕ ПРЕРЫВАНИЕ (ВЫВОД)

83 90 f0 di ;

84 91 4a out 2 ;

85 92 e8 ei ;

86 93 d8 rin ;

88 94 00 .org 0A5 ; 1 АППАРАТНОЕ ПРЕРЫВАНИЕ

89 a5 00 ; прерывание – изменение знаков “-” на “+”

90 a5 00 ; и “+” на “-” в алгоритме подпрограммы

91 a5 00 ; и рестарт программы с очисткой стека и рабочих переме

92 a5 f0 di ;

93 a6 30 01 mvi a,1 ;

94 a8 18 30 stor a,FLAG ; флаг для выбора алгоритма

95 aa 32 fd exit1: mvi sp, 0fd ; почистить стек

96 ac 30 13 mvi a, begin2 ;

97 ae 1c 81 stor a, 01(sp) ; в sp команду из начала программы

98 b0 e8 ei ;

99 b1 d8 rin ;

101 b2 00 .org 0E0 ; 2 АППАРАТНОЕ ПРЕРЫВАНИЕ

102 e0 f0 di ;

103 e1 14 81 load a,01(sp) ; загрузка SP (счётчика)

104 e3 4e out 6 ;

105 e4 14 80 load a,02(sp) ; загрузка F (флагов/признаков)

106 e6 4f out 7 ;

107 e7 e8 ei ;

108 e8 d8 rin ;

\*\*\*\*\* ’ Ў«Ёж  ЇҐаҐЄаҐбв­ле ббл«®Є \*\*\*\*\*

Ё¬п ®ЇаҐ¤Ґ«Ґ­ЁҐ ббл«ЄЁ

Y 2f 1b 16

X 2e 1d 14 11

FUNC2 47 38

FUNC1 39

FUNC 31 28 1f 18

FLAG 30 a9 34 d

EXIT1 aa

ENDFUNC 53 46

CYCLIN 81 89

BEGIN2 13 ad

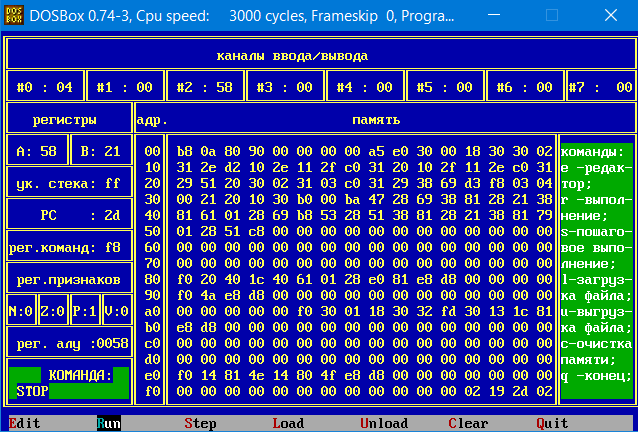
BEGIN a 1

\*\*\*\* ЌҐ ®Ў­ аг¦Ґ­® ®иЁЎ®Є. “бЇҐи­®Ґ § ўҐаиҐ­ЁҐ \*\*\*\*

Тесты :

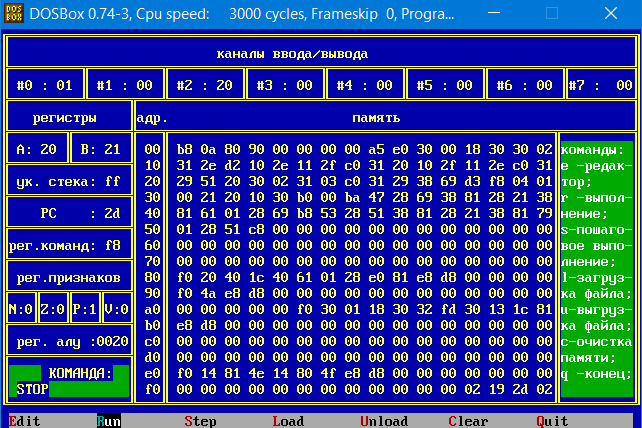
X=3, Y=4

F(3,4)+F(4,3)-F(2,3) = 8810 = 5816

****

X=4, Y=1

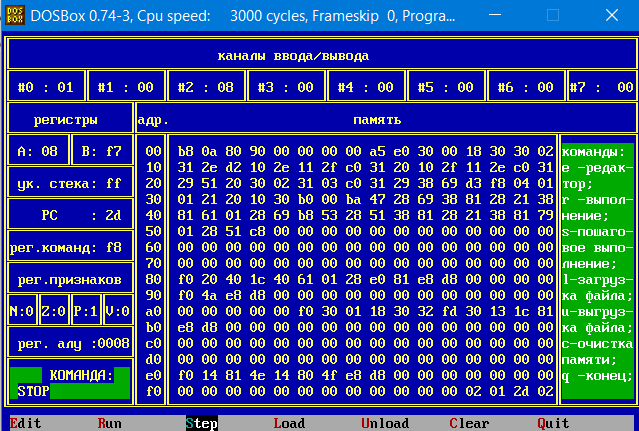
F(4,1)+F(1,4)-F(2,3) = 3210 = 2016



**Запуск первого апп. прерывания:**

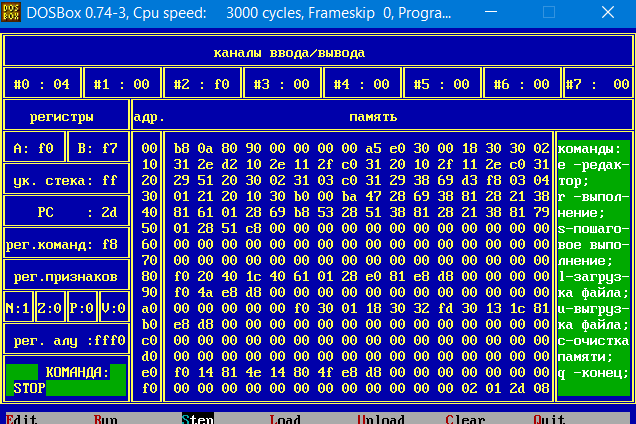
X=4, Y=1

F2(4,1)+F2(1,4)-F2(2,3) = 810 = 816

****

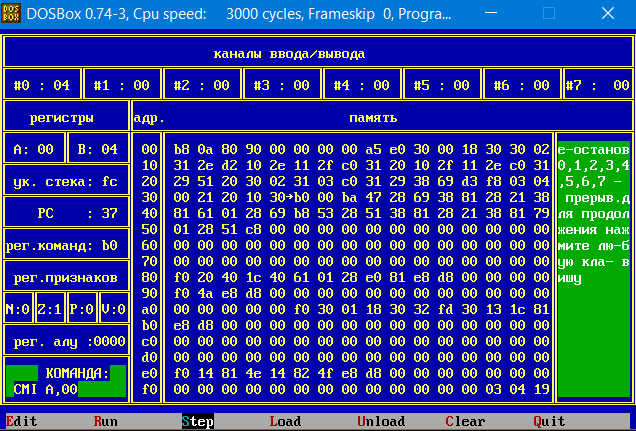
X=3, Y=4

F2(3,4)+F2(4,3)-F2(2,3) = -1610 = F016 (N=1)

****

**Запуск второго апп. прерывания:**

Счётчик PC и значения регистров F, хранящиеся в стеке

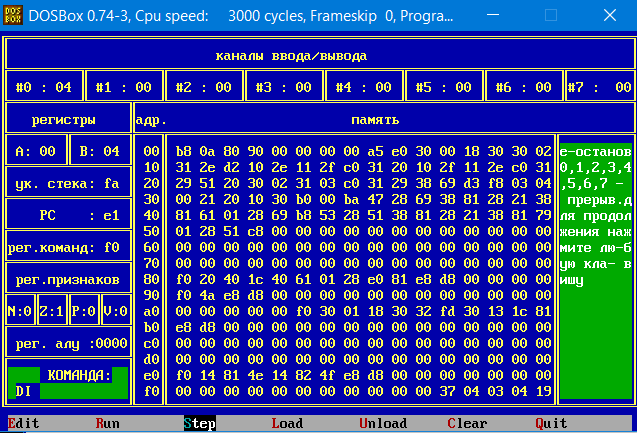
**До вызова прерывания:**

F – 0100

SP – fc

PC – 37

Stack – 00 00 ^00 03 04 19



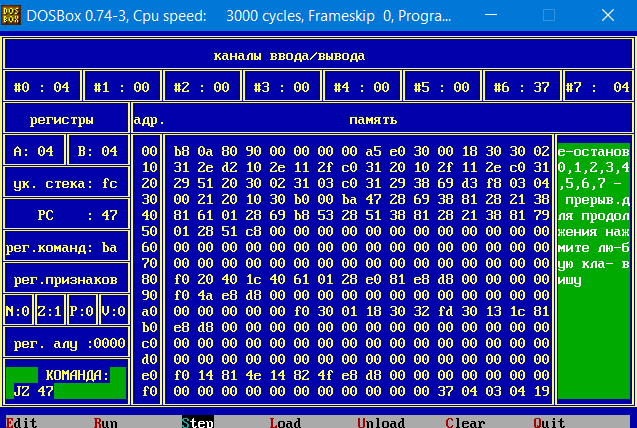
**После вызова прерывания:**

F – 0100

SP – fa

PC – e1

Stack – ^00 37 04 03 04 19



**После выхода из прерывания:**

F – 0100 (вернулось, хотя не изменялось)

SP – fc (вернулось)

PC – 47 (ушел вперед, т.к. выполнились команды)

Stack – 00 37 ^04 03 04 19 (данные не испорчены)

Алгоритм:

Алгоритм линейный за исключением функции FUNC: F(A,B)...

FLAG устанавливается нулем в начале программы, но ему присваивается 1 в первом аппаратном прерывании.

На входе у функции A и B в общих регистрах, на выходе – результат в rA.

