«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 №2

«Cистема команд процессоров и методы адресации»

Студент группы ПИ 92 В.М. Шульпов

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

Барнаул 2020

Цель лабораторной работы:

Цель данной лабораторной работы состоит в дальнейшем изучении особенностей

выполнения команд процессоров различных групп и знакомстве с методами адресации

памяти при разработке программ на машинном языке.

Задание к лабораторной работе:- изучить соответствующий теоретический материал, используя конспекты и

литературу;

- на примере функциональной модели ЭВМ-2 (приложение 2) ознакомиться с

системой команд процессора и особенностями выполнения команд различных групп

(особое внимание уделить реализации команд умножения и деления, переходов и

организации цикла со счетчиком);

- рассмотреть допустимые в системе команд формы адресации и их реализацию;

- составить программы для реализации заданных выражений в кодах модели или

используя язык ассемблера модели, при этом учесть то, что программа для реализации

линейного выражения должна быть реализована тремя способами - с использованием

абсолютной формы адресации, с использованием относительной формы и с

использованием только регистров и стэковой памяти; программа для реализации

ветвящегося процесса должна обязательно использовать команды сравнения;

циклическая программа должна быть реализована любыми двумя из трех способов - по

схеме цикла с предусловием, цикла с постусловием, и цикла со счетчиком;

- ввести полученные программы в моделируемую память; выполнить программы

для различных вариантов исходных данных, наблюдая, как изменяется состояние

компонентов процессора ЭВМ и памяти при выполнении каждой команды; разобраться,

почему и как модифицируются те или иные объекты, как выполняются команды и как

осуществляется доступ к операндам;

- составить отчет по лабораторной работе;

Заданное выражение: 

1. Абсолютная адресация

in 0 ;ввод числа 'A' из порта с адресом 0 в РОН a

stor a,A ;запомнить число 'A' по символическому адресу A

in 1 ;ввод числа 'B' из порта с адресом 1 в РОН a

stor a,B ;запомнить число 'B' по символическому адресу B

in 2 ;ввод числа 'C' из порта с адресом 2 в РОН a

stor a,C ;запомнить число 'C' по символическому адресу C

in 3 ;ввод числа 'D' из порта с адресом 3 в РОН a

stor a,D ;запомнить число 'D' по символическому адресу D

load b,A ;в rB загрузить 'A'

mui a, 1 ;расширить знак(преобразовать слово в двойное слово)

dvm a,B ;(rA,rB)/B-> частное в rB, остаток в rA

stor b,tmp ;запомнить результат деления по адресу tmp

load b,D ;в rB загрузить 'D'

mum a,D ;(rB)\*(F)-> (rA,rB)

stor b,tmp2 ;запомнить результат умножения по адресу tmp2

load b,C ;в rB загрузить 'C'

mui a, 1 ;расширить знак(преобразовать слово в двойное слово)

dvm a,tmp2 ;(rA,rB)/B-> частное в rB, остаток в rA

stor b,tmp2 ;запомнить результат деления по адресу tmp2

load a,tmp ;в rA загрузить 'tmp'

SBR a,b ;вычесть из (rA) (rB)

out 7 ;вывод в 7 поле

stop ;стоп

A: .ds 1 ;ячейка (1 зарезервированный байт) для A

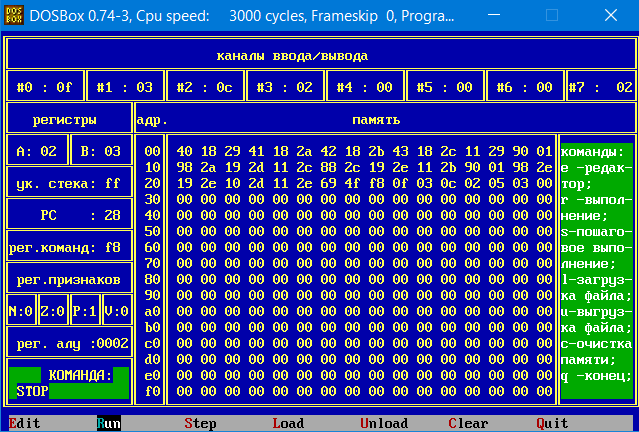
B: .ds 1 ;ячейка (1 зарезервированный байт) для B

C: .ds 1 ;ячейка (1 зарезервированный байт) для C

D: .ds 1 ;ячейка (1 зарезервированный байт) для D

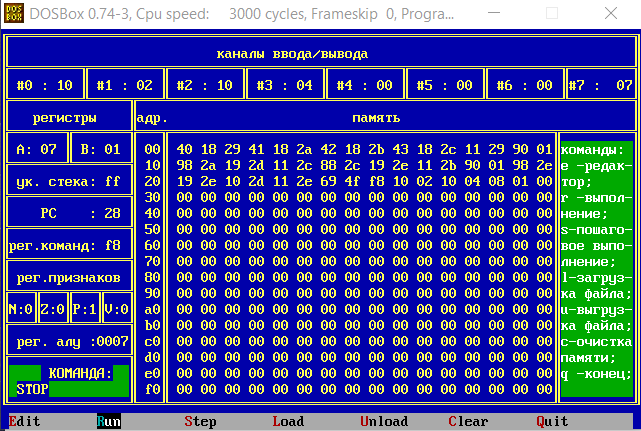
tmp: .ds 1 ;ячейка (1 зарезервированный байт) для tmp

tmp2: .ds 1 ;ячейка (1 зарезервированный байт) для tmp2



16-ая СС: f/3-c/(2\*2)=2

10-ая СС: 15/3-12/(2\*2)=5-3=2



16-ая СС: 10/2-10/(4\*4)=7

10-ая СС: 16/2-16/(4\*4)=8-1=7

1. Относительная адресация

.base b ; rB - базовый

start: mvi b,start; Относительный адрес - в rB: rB - базовый

in 0 ;ввод числа 'A' из порта с адресом 0 в РОН a

stor a,A ;запомнить число 'A' по символическому адресу A

in 1 ;ввод числа 'B' из порта с адресом 1 в РОН a

stor a,B ;запомнить число 'B' по символическому адресу B

in 2 ;ввод числа 'C' из порта с адресом 2 в РОН a

stor a,C ;запомнить число 'C' по символическому адресу C

in 3 ;ввод числа 'D' из порта с адресом 3 в РОН a

stor a,D ;запомнить число 'D' по символическому адресу D

push b ;(rB)v (sp) сохранить базовый адpес в стеке

load a,B ;в rA загрузить 'A'

push a ;(rB)v (sp) сохранить базовый адpес в стеке

load b,A ; загpузить в rB A, используя относительный адpес

mui a,1 ; pаспpостpанить знак

dvm a,1(sp) ; pазделить (rA,rB) на содержимое веpхушки стека

mov a,b ; пеpеслать частное в rA

pop b ; (sp)^(rB) : вытолкнуть из стека в rB B

pop b ; (sp)^(rb) : восстановить из стека базовый адрес

stor a,tmp ; запомнить по относительному адресу результат A/B

push b ;(rB)v (sp) сохранить базовый адpес в стеке

load b,D ;в rB загрузить 'D'

mur a,b ;(rB)\*(rB)-> (rA,rB)

mov a,b ; пеpеслать частное в rA

pop b ; (sp)^(rb) : восстановить из стека базовый адрес

push b ; вернуть 0 в стек

push a ; положить D\*D в стек

load b,c ; загрузить ‘C’ в rB

mui a, 1 ;расширить знак(преобразовать слово в двойное слово)

dvm a,1(sp) ;(rA,rB)/B-> частное в rB, остаток в rA

mov a,b ; пеpеслать частное в rA

pop b ; (sp)^(rB) : вытолкнуть из стека в rB B

pop b ; (sp)^(rb) : восстановить из стека базовый адрес

push b ; вернуть 0 в стек

push a ; положить C/(D\*D) в стек

load a,tmp ;в rA загрузить 'tmp'

pop b ; вытянуть C/(D\*D) в rB

sbr a,b ;вычесть из (rA) (rB)

out 7 ;вывод в 7 поле

stop ;стоп

A: .ds 1 ;ячейка (1 зарезервированный байт) для A

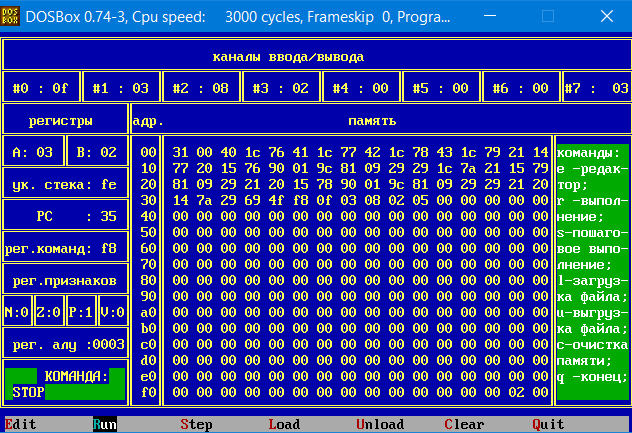
B: .ds 1 ;ячейка (1 зарезервированный байт) для B

C: .ds 1 ;ячейка (1 зарезервированный байт) для C

D: .ds 1 ;ячейка (1 зарезервированный байт) для D

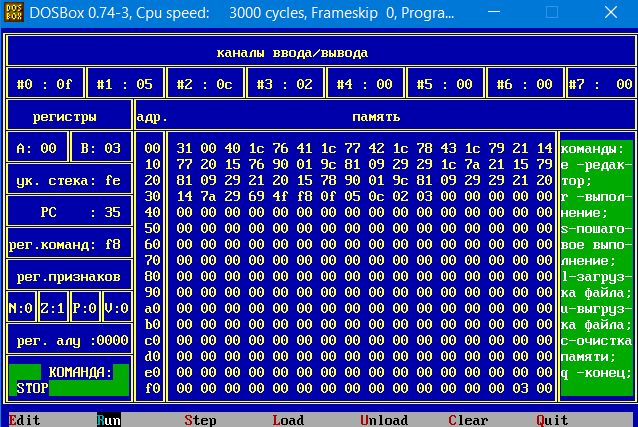
tmp: .ds 1 ;ячейка (1 зарезервированный байт) для tmp

tmp2: .ds 1 ;ячейка (1 зарезервированный байт) для tmp2



16-ая СС: f/3-8/(2\*2)=3

10-ая СС: 15/3-8/(2\*2)=5-2=3



16-ая СС: f/5-c/(2\*2)=0

10-ая СС: 15/5-12/(2\*2)=3-3=0