МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

ОТЧЕТ

по лабораторной работе №1

«Программная модель процессора»

Вариант 17

по дисциплине

«Принципы и методы организации системных

программных средств»

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ Викулова Е.Н.

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ Сухоруков В.А.\_\_\_

(подпись) (фамилия, и.,о.)

\_\_\_\_\_\_19-В-2\_\_\_\_\_\_\_\_

(шифр группы)

Работа защищена «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2021

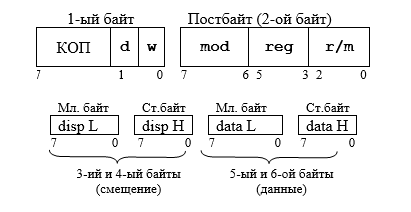
# Цель

Изучить программную модель процессора *Intel* 8086/88 (программно-доступные регистры, модель и адресацию память, модель внешних устройств, систему команд). Изучить структуру и форматы команд. Получить навыки работы с отладчиком. Разработать простейшую программу на ассемблере и выполнить ее в пошаговом режиме.

Примеры ассемблерных инструкций и их кодов для различных методов адресации

## Форматы команд

Общий формат команды (6 байтов):



Первый байт команды содержит код операции КОП и имеет два однобитных поля **w** и **d**. Поле **w** (word) определяет тип операнда:

если **w** =1, то операнд – слово (2 байта),

если **w** =0, то операнд – байт.

Поле **d** определяет направление передачи данных относительно регистра, адресуемого полем **reg** постбайта:

**d** =1 – пересылка в регистр, код которого в **reg**;

**d** =0 – пересылка из регистра, код которого в **reg**.

Постбайт (табл.1,2) определяет операнды, участвующие в команде, и имеет поля:

1. **mod** («режим», 2 бита) – определяет как интерпретировать поле **r/m**

(т.е. регистр или память?):

Если **mod=**11, то операнд в регистре (r),

иначе – в памяти (m):

00 – без смещения,

01 – 1 байт смещения,

10 – 2 байта смещения;

1. **reg** («регистр», 3 бита) – определяет регистр, участвующий в команде или операции, т.е. содержит код регистра;
2. **r/m** («регистр/память», 3 бита) - совместно с полем **mod** указывает местоположение операнда: если операнд находится в регистре, то в **r/m** указан код регистра, если операнд в памяти, то задается метод адресации (табл. 1).

Таблица

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **r/m** | **mod** | | | | | |
| **00** | | **01** | | **10** | |
| 000 | DS: | ВХ +SI | DS: | ВХ+SI+disp8 | DS: | ВХ+SI+disp16 |
| 001 | DS: | ВХ+DI | DS: | ВХ+DI+disp8 | DS: | ВХ+DI+disp16 |
| 010 | SS: | ВР+SI | SS: | ВР+SI+disp8 | SS: | ВР+SI+disp16 |
| 011 | SS: | ВР+DI | SS: | ВР+DI+disp8 | SS: | ВР+DI+disp16 |
| 100 | DS: | SI | DS: | SI+disp8 | DS: | SI+disp16 |
| 101 | DS: | DI | DS: | DI+disp8 | DS: | DI+disp16 |
| 110 | DS: | disp 16 | SS: | BP+disp8 | SS: | BP+disp16 |
| 111 | DS: | BX | DS: | BX+disp8 | DS: | BX+disp16 |

Поле reg постбайта используется для адресации тогда, когда в команде задаются 2 операнда (регистр-регистр, регистр-память).

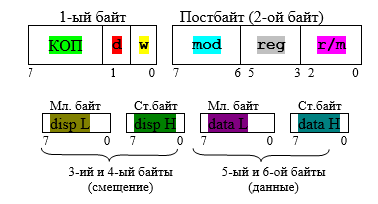
Для случая регистр-регистр: один из регистров кодируется полем reg постбайта, другой – полями mod=11 и r/m (коды регистров – табл. 2).

Для случая регистр-память: регистр кодируется полем reg постбайта, память – полями mod=11 и r/m в соответствии с таблицами 1, 2.

Таблица

|  |  |  |
| --- | --- | --- |
| **reg**  **r/m (при mod=11)** | **Байт (w=1)** | **Слово (w=0)** |
| 000 | AL | AX |
| 001 | CL | CX |
| 010 | DL | DX |
| 011 | BL | BX |
| 100 | AH | SP |
| 101 | CH | BP |
| 110 | DH | SI |
| 111 | BH | DI |

## Примеры инструкций

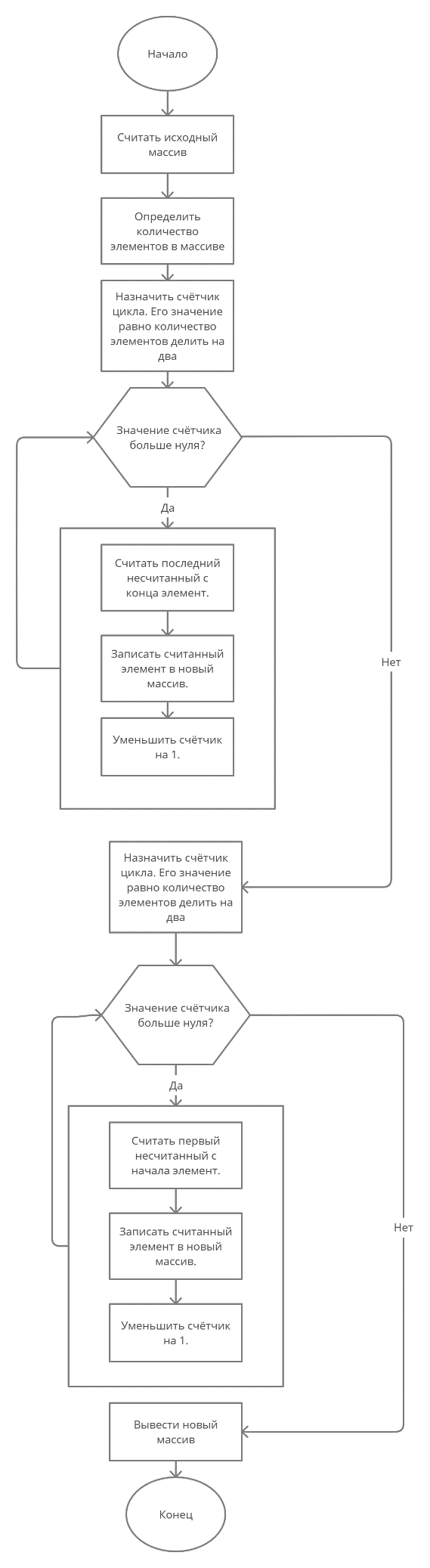


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Метод адресации** | **Инструкция** | **Код операции** | **Код операции в двоичном виде** | **Описание инструкции** |
| Непосредственная | add dx,5 | 83C205  83 - 1-ый байт  С2 - постбайт  05 – data L | 10000011  11000010  00000101 | Добавление к содержимому регистра cx константы 5. |
| Регистровая | mov bx, ax | 8BD8  8B - 1-ый байт  D8 - постбайт | 10001011  11011000 | Запись в регистр bx содержимого регистра ax |
| Прямая | mov dx,[0200] | 8B160002  8B - 1-ый байт  16 – постбайт  00 - disp L  02 - disp H | 10001011  00010110  00000000  00000010 | Запись в регистр dx содержимого ячейки памяти 200h. |
| Косвенно базовая | mov dx, [bx] | 8B17  8B - 1-ый байт  17 - постбайт | 10001011  00000111 | Записать в регистр dx содержимое ячейки памяти, адрес которой хранится в bx. |
| Косвенно-индексная | mov [di],dx | 8915  89 - 1-ый байт  15 - постбайт | 10001001  00010101 | Записать по адресу di содержимое регистра al. |
| Базово-индексная | mov dx, [bx+si] | 8B10  8B - 1-ый байт  10 - постбайт | 10001011  00010000 | Записать в регистр dx содержимое ячейки памяти, адрес которой вычисляется суммой содержимого регистров bx и si. |
| Неявная | mov sb | A4  A4 - 1-ый байт | 10100100 | Записать в ячейку по адресу es:di байт из ячейки с адресом ds:si. |

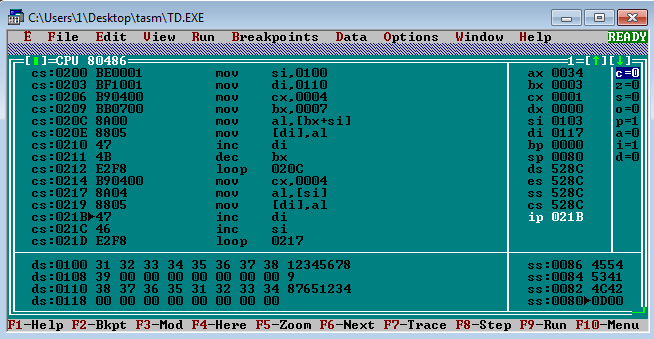
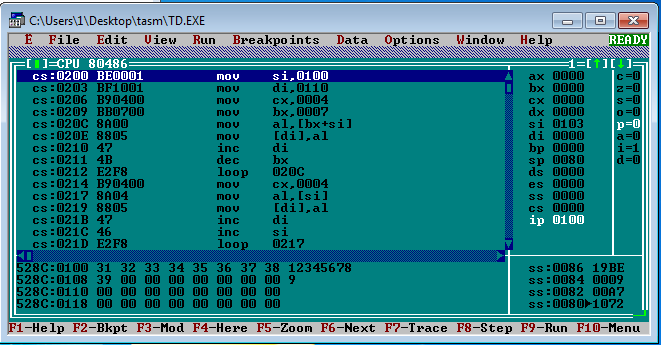
# Вариант задания

Перестановка a(n), a(n-1), a(n-2), ..., a(n/2), a(1), a(2), ... ,a(n/2-1).

# Алгоритм решения задачи



# Текст программы, входные и выходные данные, пример выполнения программы



# Вывод

В ходе лабораторной работы изучил программную модель процессора *Intel* 8086/88. Изучил структуру и форматы команд. Получил навыки работы с отладчиком. Разработал простейшую программу на ассемблере и выполнить ее в пошаговом режиме.