

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:

(подпись)

Викулова Е.Н.
(фамилия, и.,о.)

СТУДЕНТ:

(подпись)

Сухоруков В.А.
(фамилия, и.,о.)

19-В-2
(шифр группы)

Работа защищена «__» _____

С оценкой _____

Нижний Новгород 2021

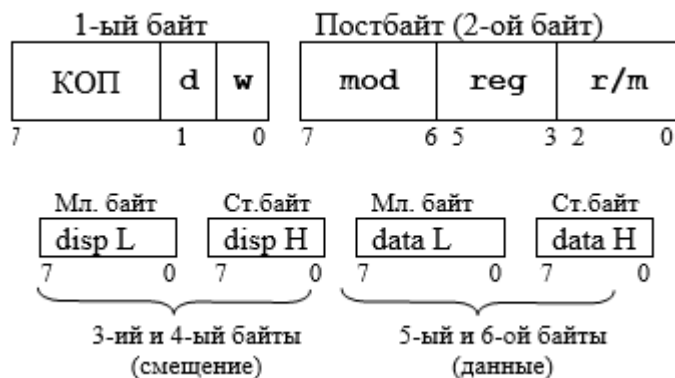
Цель

Изучить программную модель процессора *Intel 8086/88* (программно-доступные регистры, модель и адресацию памяти, 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;

2. **reg** («регистр», 3 бита) – определяет регистр, участвующий в команде или операции, т.е. содержит код регистра;

3. **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 1

r/m	mod					
	00		01		10	
000	DS:	BX +SI	DS:	BX+SI+disp8	DS:	BX+SI+disp16
001	DS:	BX+DI	DS:	BX+DI+disp8	DS:	BX+DI+disp16
010	SS:	BP+SI	SS:	BP+SI+disp8	SS:	BP+SI+disp16
011	SS:	BP+DI	SS:	BP+DI+disp8	SS:	BP+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).

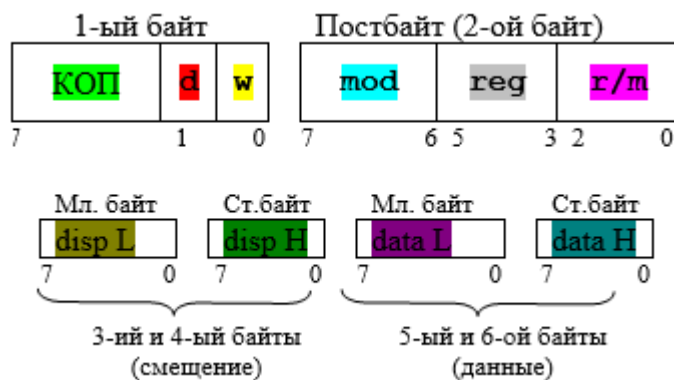
Для случая регистр-регистр: один из регистров кодируется полем reg постбайта, другой – полями mod=11 и r/m (коды регистров – табл. 2).

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

Таблица 2

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

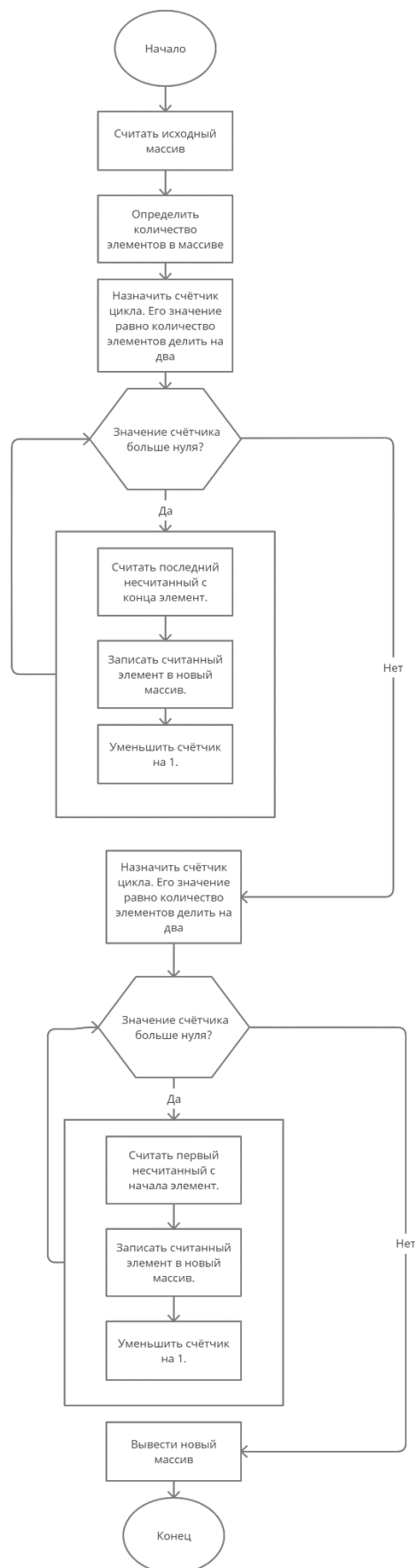


Метод адресации	Инструкция	Код операции	Код операции в двоичном виде	Описание инструкции
Непосредственная	add dx,5	83C205	<div>10000011</div> <div>11000010</div> <div>00000101</div>	Добавление к содержимому регистра dx константы 5.
Регистровая	mov bx, ax	8BD8	<div>10001011</div> <div>11011000</div>	Запись в регистр bx содержимого регистра ax
Прямая	mov dx,[0200]	8B160002	<div>10001011</div> <div>00010110</div> <div>00000000</div> <div>00000010</div>	Запись в регистр dx содержимого ячейки памяти 200h.
Косвенно базовая	mov dx, [bx]	8B17	<div>10001011</div> <div>00000111</div>	Записать в регистр dx содержимое ячейки памяти, адрес которой хранится в bx.
Косвенно-индексная	mov [di],dx	8915	<div>10001001</div> <div>00010101</div>	Записать по адресу di содержимое регистра dx.
Базово-индексная	mov dx, [bx+si]	8B10	<div>10001011</div> <div>00010000</div>	Записать в регистр dx содержимое ячейки памяти, адрес которой вычисляется суммой содержимого регистров bx и si.
Неявная	mov sb	A4	<div>10100100</div>	Записать в ячейку по адресу es:di байт из ячейки с адресом ds:si.

Вариант задания

Перестановка $a(n), a(n-1), a(n-2), \dots, a(n/2), a(1), a(2), \dots, a(n/2-1)$.

Алгоритм решения задачи



Текст программы, входные и выходные данные, пример выполнения программы

C:\Users\1\Desktop\tasm\TD.EXE

File Edit View Run Breakpoints Data Options Window Help

READY

[]-CPU 80486

Address	Instruction	Register	Value
cs:0200	BE0001	mov	si,0100
cs:0203	BF1001	mov	di,0110
cs:0206	B90400	mov	cx,0004
cs:0209	BB0700	mov	bx,0007
cs:020C	8A00	mov	al,[bx+si]
cs:020E	8805	mov	[dil,al]
cs:0210	47	inc	di
cs:0211	4B	dec	bx
cs:0212	E2F8	loop	020C
cs:0214	B90400	mov	cx,0004
cs:0217	8A04	mov	al,[si]
cs:0219	8805	mov	[dil,al]
cs:021B	47	inc	di
cs:021C	46	inc	si
cs:021D	E2F8	loop	0217

Register	Value
ax	0000
bx	0000
cx	0000
dx	0000
si	0103
di	0000
bp	0000
sp	0080
ds	0000
es	0000
ss	0000
cs	0000
ip	0100

Address	Value
528C:0100	31 32 33 34 35 36 37 38 12345678
528C:0108	39 00 00 00 00 00 00 00 9
528C:0110	00 00 00 00 00 00 00 00
528C:0118	00 00 00 00 00 00 00 00

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

C:\Users\1\Desktop\tasm\TD.EXE

File Edit View Run Breakpoints Data Options Window Help

READY

[]-CPU 80486

Address	Instruction	Register	Value
cs:0200	BE0001	mov	si,0100
cs:0203	BF1001	mov	di,0110
cs:0206	B90400	mov	cx,0004
cs:0209	BB0700	mov	bx,0007
cs:020C	8A00	mov	al,[bx+si]
cs:020E	8805	mov	[dil,al]
cs:0210	47	inc	di
cs:0211	4B	dec	bx
cs:0212	E2F8	loop	020C
cs:0214	B90400	mov	cx,0004
cs:0217	8A04	mov	al,[si]
cs:0219	8805	mov	[dil,al]
cs:021B	47	inc	di
cs:021C	46	inc	si
cs:021D	E2F8	loop	0217

Register	Value
ax	0034
bx	0003
cx	0001
dx	0000
si	0103
di	0117
bp	0000
sp	0080
ds	528C
es	528C
ss	528C
cs	528C
ip	021B

Address	Value
ds:0100	31 32 33 34 35 36 37 38 12345678
ds:0108	39 00 00 00 00 00 00 00 9
ds:0110	38 37 36 35 31 32 33 34 87651234
ds:0118	00 00 00 00 00 00 00 00

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

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.