

Министерство образования и науки Российской Федерации  
**Нижекамский химико-технологический институт (филиал)**  
Федерального государственного бюджетного образовательного учреждения  
высшего профессионального образования  
«Казанский национальный исследовательский технологический университет»

**Г.П. Сечина**

# **МИКРОПРОЦЕССОРЫ И МИКРО-ЭВМ**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
К ЛАБОРАТОРНЫМ РАБОТАМ**

**ЧАСТЬ I**

**Нижекамск  
2012**

**УДК 621.38**  
**С 33**

Печатается по решению редакционно-издательского совета Нижнекамского химико-технологического института (филиал) ФГБОУ ВПО «КНИТУ».

**Рецензенты:**

**Саримов Н.Н.**, кандидат физико-математических наук;  
**Изотов П.В.**, нач. АСУТП завода «Этилен».

**Сечина, Г.П.**

**С 33** Микропроцессоры и микро-ЭВМ : методические указания к лабораторным работам. Часть I / Г.П. Сечина. – Нижнекамск : Нижнекамский химико-технологический институт (филиал) ФГБОУ ВПО «КНИТУ», 2012. - 39 с.

Рассмотрены вопросы практического освоения методики программирования в кодах микропроцессора Intel 80x86. Приведены задания для самостоятельной работы студентов.

Предназначены для лабораторного обеспечения курсов «Микропроцессорные средства» по специальности 230102 «Автоматизированные системы обработки информации и управления» и «Микропроцессоры и микро-ЭВМ» для специальности 220301 «Автоматизация технологических процессов и производств».

Подготовлены на кафедре автоматизации технологических процессов и производств Нижнекамского химико-технологического института КГТУ.

**УДК 621.38**

© Сечина Г.П., 2012

© Нижнекамский химико-технологический институт (филиал) ФГБОУ ВПО «КНИТУ», 2012

## СОДЕРЖАНИЕ

Введение.....	4
1. Рекомендации по освоению методики программирования.....	5
2. ЛАБОРАТОРНЫЕ РАБОТЫ	
2.1. Лабораторная работа 1.....	7
2.2. Лабораторная работа 2.....	10
2.3. Лабораторная работа 3.....	17
Приложения .....	24
Литература.....	39

## **ВВЕДЕНИЕ**

Данное пособие представляет собой введение в методику программирования в кодах микропроцессора Intel 80x86.

В первом разделе пособия приводятся рекомендации по вопросам практического освоения методики программирования.

Во втором разделе рассматриваются непосредственно лабораторные работы, общие сведения и задания для самостоятельной работы. Приведены программы сложения простых чисел и длинных чисел, а также программы пересылки массива информации из одной области памяти в другие и выборки из массива длинных чисел данного интервала.

В конце каждой лабораторной работы студентам предлагается ответить на контрольные вопросы.

В пособии имеются приложения, включающие в себя варианты заданий к лабораторным работам.

При разработке пособия использовались работы [1,2].

## 1. Рекомендации по освоению методики программирования

При выполнении лабораторных работ для ввода данных в память компьютера и их пошагового выполнения можно пользоваться системной утилитой `debug.exe`, однако рекомендуется использовать программы `Str2Hex.exe` и `TD.exe`, которые находятся в каталоге `MPS_on_PC`.

Программа `Str2Hex.exe` предназначена для преобразования текстовых файлов, содержащих тексты программ в машинных кодах (шестнадцатиричные числа), в программы типа `.com`.

1. Для получения `com`-программы вышеуказанным образом необходимо: Загрузить Norton Commander (`Dos Navigator`, `Far Manager` или т.п.).
2. Перейти в каталог `MPS_on_PC`.
3. Создать текстовый файл. Для этого:
  - 1) Нажать `<Shift> + <F4>`;
  - 2) Ввести имя создаваемого файла и нажать `<Enter>`;
  - 3) Ввести текст программы в машинных кодах (шестнадцатиричные числа);

*Примечание:* Допускается в конце каждой строки вводить комментарии, которые отделяются от текста основной программы символом `';`.

4. Нажать `<Esc>` и ответить 'Да' на предложение сохранить файл;
5. Преобразовать текстовый файл в программу типа `.com`. Для этого: Набрать в командной строке команду: `Str2Hex.exe <имя_вашего_файла>`
6. Загрузить созданную программу в отладчик `Turbo Debugger (td.exe)`. Для этого наберите в командной строке команду: `td.exe <имя_com-файла>`
7. Нажимая клавишу `<F7>`, проследите за пошаговым выполнением программы.

*Примечание:* Для получения дополнительной информации, обратитесь к встроенной справке.

8. Для выхода из TD нажмите <Alt> + <X>.

## 2.1 Лабораторная работа 1

### Изучение работы микропроцессора Intel 80x86

Цель работы: приобрести практические навыки работы с микропроцессором Intel 80x86, изучить возможности устройства Intel 80x86, практически освоить режимы его работы.

Продолжительность лабораторного занятия - 4 часа.

Самостоятельная подготовка - 4 часа.

#### 2.1.1. САМОСТОЯТЕЛЬНАЯ РАБОТА

- 1.1. Изучить и практически освоить представление чисел в двоичной и шестнадцатиричной системах счисления.
- 1.2. Изучить назначение и структуру процессора Intel 80x86.
- 1.3. Изучить методику использования программы-отладчика.
- 1.4. Изучить последовательность действий при выполнении программы в пошаговом режиме и режиме с остановом по контрольным точкам.
- 1.5. Вычислить выражение:  $N = n * 9$ , где  $n$  - Ваш номер в журнале преподавателя. Полученное в результате перемножения число представить в шестнадцатиричной и двоичной системах счисления.
- 1.6. Ознакомиться с приведенной в таблице 1 программой сложения трех чисел.
- 1.7. Согласно приведенному выше примеру сложить число, полученное Вами при выполнении операции перемножения (пункт 1.5) с числами  $N1$  и  $N2$  (см. Приложение 1). Порядок выполнения операции сложения записать в виде программы и оформить в виде таблицы.

В программе нахождения суммы трех чисел для Intel 80x86 использованы следующие обозначения и исходные данные:

0002 - первое слагаемое

0003 - второе слагаемое  
0005 - третье слагаемое  
100 - адрес начала программы

**Таблица 1 - Программа сложения ряда чисел на Intel 80x86**

<b>Адрес</b>	<b>Машин- ный код</b>	<b>Мнемокод</b>	<b>Комментарий</b>
100	B80200	Mov ax,0002	Загрузка первого слагаемого в регистр аккумулятор AX
103	B90300	Mov cx,0003	Загрузка второго слагаемого в регистр аккумулятор CX
106	BA0500	Mov dx,0005	Загрузка третьего слагаемого в регистр аккумулятор DX
109	03C1	Add ax,cx	$ax=ax+cx$
010B	03C2	Add ax,dx	$Ax=ax+dx$
010D	CD20	Int 20	Окончание программы

## **2.12. СОДЕРЖАНИЕ ОТЧЕТА**

1. Краткое описание цели работы и результатов самостоятельной подготовки к лабораторному занятию.
2. Тексты программ выполнения лабораторной работы.



### **2.13. КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Что такое микро-ЭВМ?
2. Что такое микропроцессор? Его существенные отличия от микро-ЭВМ.
3. Какие области памяти доступны программисту?
4. Какие адреса памяти относятся к ПЗУ и ОЗУ?
5. В чем заключается принципиальное отличие ПЗУ от ОЗУ?
6. Что происходит при попытке записи данных в ПЗУ?
7. Сохраняется ли в памяти Intel 80X86 программа после выключения источника питания?
8. Что такое программа-отладчик и ее функции?

## **2.2 Лабораторная работа 2**

### **Изучение арифметических команд и команд пересылки данных**

Цель работы: изучить команды пересылки данных и арифметические команды, способы адресации данных при выполнении простых программ по нахождению суммы ряда чисел и сложении (вычитании) с переносом (заемом).

Продолжительность лабораторного занятия - 4 часа.

Самостоятельная подготовка - 4 часа.

#### **2.2.1. САМОСТОЯТЕЛЬНАЯ РАБОТА**

- 1.1. Ознакомиться с настоящим описанием.
- 1.2. Изучить способы адресации данных Intel 80x86:
  - 1) непосредственная;
  - 2) прямая;
  - 3) косвенная;
  - 4) регистровая.
- 1.3. Изучить назначение разрядов регистра флагов (признаков) Intel 80x86, а также группу арифметических команд.
- 1.4. В соответствии с заданием, выдаваемым преподавателем, подготовить данные с указанием для каждой вновь изучаемой команды ее характеристики:
  - 1) шестнадцатичный код;
  - 2) мнемокод;
  - 3) формат команды;
  - 4) используемый способ адресации;
  - 5) действие, выполняемое командой;
  - 6) влияние результата выполнения команды на регистр признаков.

### 2.2.2. СОДЕРЖАНИЕ ЛАБОРАТОРНОЙ РАБОТЫ

Задачи, решаемые на ЭВМ, не сводятся к обработке отдельного элемента данных с помощью одной операции. Напротив, они требуют обработки многих элементов данных (например, массива или блока данных), которые занимают отдельные ячейки памяти. Программа может выполнять одну и ту же операцию над содержимым ячеек, причем выполнять ее многократно с помощью программных циклов.

Например, необходимо найти сумму ряда чисел. При этом задача состоит в том, чтобы осуществить суммирование нескольких чисел сразу. Эти числа могут представлять собой совокупности входных сигналов, находящихся под управлением системы, число изделий (или число сообщений), изготовленных (или принятых) за определенный промежуток времени. Предположим, что сумма не превышает 65535 и для ее хранения достаточно одной 16-битовой ячейки памяти.

В программе суммирования ряда чисел для Intel 80x86 использованы следующие исходные данные и обозначения:

0117 – адрес ячейки памяти, где указана длина массива.

cx - счетчик, указатель длины массива.

bx - адресный указатель данных.

ax - регистр-аккумулятор.

0119 - адрес ячейки памяти, где хранится первый элемент массива.

100 – адрес начала программы.

**Таблица 2 - Программа сложения ряда чисел**

Адрес	Машинный код	Мнемокод	Мет-ка	Комментарий
0100	BE1701	mov si,0117		Загрузка счётчика
0103	AD	Lodsw		
0104	8BC8	Mov cx,ax		
0106	BB1901	Mov bx,0119		Загрузка адресного указателя
0109	2BC0	Sub ax,ax		Обнуление ax
010B	0307	Add ax,[bx]		Сложение элемента массива с ax
010D	49	Dec cx		Уменьшение счётчика
010E	7405	Je 0115		Сложение элементов закончено?
0110	83C302	Add bx,0002		Переход к следующему адресу
0113	EBF6	Jmp 010B		Организация цикла
0115	CD20	Int 20	M1	Окончание программы
0117	0300			Данные (длина массива)
0119	0100			Данные (длина массива)
011B	0200			Данные (длина массива)
01D	0300			Данные (длина массива)

Исходные данные см. Приложение 2.

В большинстве случаев точность вычислений, обеспечиваемая одной ячейкой памяти, недостаточна. Поэтому рассмотрим арифметику для чисел, занимающих несколько ячеек. Задача состоит в том, чтобы сложить два числа длиной более 16

бит каждое. Исходные числа располагаются в области памяти таким образом, что сначала идут младшие разряды, а затем более старшие. Полученную сумму необходимо поместить в те ячейки памяти, где хранилось первое число

В программе суммирования двух длинных чисел для Intel 80x86 использованы следующие исходные данные и обозначения:

0003 - длина чисел в байтах;

cx - счетчик, указатель длины чисел;

di - адресный указатель первого слагаемого;

0115 - адрес ячейки памяти, где хранится младший байт первого числа;

si - адресный указатель второго слагаемого;

011B - адрес ячейки памяти, где хранится младший байт второго числа;

100 - адрес начала программы.

**Таблица 3 - Программа сложения длинных чисел**

Адрес	Машинный код	Мнемокод	Метка	Комментарий
100	B90300	mov cx,0003		Загрузка счётчика
103	BF1501	mov di,0115		Загрузка адресного указателя 1-го числа
106	BE1B01	mov si,011B		Загрузка адресного указателя 2-го числа
109	8B05	mov ax,[di]	M1	Загрузка 1-го операнда в ax
010B	1304	adc ax,[si]		Сложение
010D	AB	stosw		Запоминание результата сложения
010E	46	inc si		Увеличение si на 1

Адрес	Машин- ный код	Мнемокод	Мет ка	Комментарий
010F	46	inc si		– “ – “ –
110	49	dec cx		Уменьшение счётчика на единицу
111	75F6	jne 0109		Организация цикла
113	CD20	int 20		Окончание программы
115	FF			Данные
116	FF			Данные
117	FF			Данные
118	FF00			Данные
011A	0001			Данные
011C	0000			Данные
011E	0000			Данные
120	0000			Данные

Исходные данные см. Приложение 3.

### 2.2.3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

По заданию преподавателя подготовить и выполнить программы сложения (вычитания) ряда чисел и сложения длинных чисел с переносом (заемом). Исходные данные необходимо взять из Приложений 2 и 3 соответственно.

### 2.2.4. СОДЕРЖАНИЕ ОТЧЕТА

1. Краткое описание цели работы и результатов самостоятельной подготовки к лабораторному занятию.
2. Тексты программ выполнения лабораторной работы.

### 2.2.5. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое команда МП?
2. Что такое система команд МП?
3. Назовите группы команд Intel 80x86?

4. Из каких частей состоит команда МП?
5. Что такое мнемокод?
6. Назовите способы адресации памяти в Intel 80x86, приведите примеры команд с соответствующими способами адресации.
7. Для чего служат команды пересылки данных?
8. Назовите значения разрядов регистра признаков Intel 80x86?
9. Какие разряды формируются при выполнении команды пересылки данных?
10. После выполнения команды пересылки данных из регистра `bx` в регистр `sx` (`MOV sx,bx`) сохраняются ли исходные данные в регистре `bx`?
11. Приведите примеры команд пересылки данных с прямой, косвенной, регистровой адресацией.
12. В каких байтах располагаются младшие и старшие разряды адреса в командах, содержащих адрес ячейки памяти?
13. В ячейке памяти с адресом 1314 находится число F4, Какие команды можно применить для записи его в регистр `sx`?
14. В регистре `dx` находится число E4. Какие команды можно использовать для записи его в ячейку памяти с адресом 1147?
15. В каких регистрах МП Intel 80x86 может храниться адрес ячейки памяти?
16. Для чего служат арифметические (логические команды)?
17. Данные какой длины может обрабатывать МП Intel 80x86?
18. Где хранятся операнды, и куда помещается результат после выполнения арифметических (логических) команд?
19. Какие разряды в регистре признаков формируются при выполнении арифметических (логических) команд?
20. Для чего применяются команды положительного и отрицательного приращения (`INC/DEC`)?
21. Как выполняется команда сравнения двух чисел? Где формируется результат выполнения этой команды?
22. Как выполняются команды простого и циклического сдвига содержимого аккумулятора?

## **2.3 Лабораторная работа 3**

### **Обработка массивов информации и организация циклов**

Цель работы: Освоить методику программирования в кодах микропроцессора Intel 80x86 при обработке массивов, выполнении и отладке программ на устройстве Intel 80x86.

Продолжительность лабораторного занятия 4 часа.

Самостоятельная подготовка - 4 часа.

#### **2.3.1. САМОСТОЯТЕЛЬНАЯ РАБОТА**

- 1.1. Ознакомиться с настоящим описанием.
- 1.2. Изучить группу команд передачи управления Intel 80x86.
- 1.3. Изучить рекомендуемую литературу [1; 2; 3] и разделы конспекта лекций по программированию в кодах МП Intel 80x86.
- 1.4. В соответствии с вариантом задания, выдаваемого преподавателем, составить программу в кодах МП Intel 80x86 и подготовить на нее полную документацию, включающую в себя:
  - схему алгоритма;
  - распределение памяти;
  - текст программы в кодах МП Intel 80x86.
- 1.5. Результаты выполнения самостоятельной работы представить в виде отчета.

#### **2.3.2. СОДЕРЖАНИЕ ЛАБОРАТОРНОЙ РАБОТЫ**

При программировании часто возникает необходимость в пересылке массива данных из одной области памяти в другую. Такая операция может понадобиться при инициализации



элементов массива, при пересылке или формировании строки данных, выводимой на экран дисплея.

Рассмотрим следующую задачу. Массив чисел расположен в области памяти, начиная с адреса NNNN, и состоит из N элементов. Необходимо переслать массив в другую область памяти, начиная с адреса DDDD. В программе для МП Intel 80x86 в качестве адресного регистра используются регистры si и di. Необходимо учитывать, что при каждом проходе программа должна изменять содержимое обоих адресных регистров.

В программе пересылки массива из одной области памяти в другую использованы следующие исходные данные и обозначения:

- 0002 - количество элементов массива;
- cx - счетчик, указатель длины массива;
- si - адресный указатель исходного массива;
- 010D - адрес ячейки памяти, где хранится первое число исходного массива;
- di - адресный указатель результирующего массива
- 0111 - адрес ячейки памяти, с которой начинается результирующий массив;
- 100 - адрес начала программы.

**Таблица 4 - Пересылка массива информации из одной области памяти в другую**

Адрес	Машинный код	Мнемокод	Комментарий
0100	B90200	mov cx,0002	Загрузка счётчика
0103	BE0D01	mov si,010D	Загрузка адресного указателя исходного массива
0106	BF1101	mov di,0111	Загрузка адресного указателя результирующего массива
0109	F2A5	repnz movsw	Пересылка данных
010B	CD20	int 20	Окончание программы
010D	FF		Данные
010E	FF		Данные
010F	FF		Данные
0110	FFAAAAAAAA		Данные
0114	AA		Данные

Исходные данные см. Приложение 4.

В ряде случаев возникает необходимость выбрать из массива информации данные, представляющие собой ряд максимальных, либо минимальных величин. Предположим, что необходимо написать программу для решения следующей задачи. Дан массив  $A1$ , состоящий из  $N$  однобайтовых чисел. Необходимо переписать из массива  $A1$  в массив  $B1$  все числа в диапазоне от  $H1$  до  $H2$ .

В программе выборки из массива чисел заданного интервала использованы следующие исходные данные и обозначения:

- 000A - количество элементов исходного массива;
- cx - счетчик, указатель длины исходного массива;
- si - адресный указатель исходного массива;
- 0119 - адрес ячейки памяти, где хранится первое число исходного массива;
- di - адресный указатель результирующего массива;
- 012C - адрес ячейки памяти, куда будет записано первое число результирующего массива;
- 0002 - нижний предел диапазона выборки;
- 0006 - верхний предел диапазона выборки;
- 100 - адрес начала программы.

**Таблица 5 - Выборка из массива данных чисел заданного интервала**

Адрес	Машинный код	Мнемокод	Метка	Комментарий
0100	B90A00	mov cx,000A		Загрузка счётчика
0103	BE1901	mov si,0119		Загрузка адресного указателя исходного массива
0106	BF2C01	mov di,012C		Загрузка адресного указателя результирующего массива
0109	AD	lodsw		Выборка числа из массива A1
010A	3D0200	cmp ax,0002		Сравнение с нижним пределом
010D	7206	jb 0115		Переход на метку M1, если число < 0002
010F	3D0600	cmp ax,0006		Сравнение с верхним пределом

Адрес	Машин- ный код	Мнемокод	Мет ка	Комментарий
0112	7701	ja 0115		Переход на метку M2, если число > 0006
0114	AB	stosw		Запись числа, попавшего в интервал
0115	E2F2	loop 0109		Организация цикла
0117	CD20	int 20		Окончание программы
0119	0000			Данные
011B	0100			Данные
011D	0200			Данные
011F	0300			Данные
0121	0400			Данные
0123	050006			Данные
0126	0007			Данные
0128	0008			Данные
012A	0009			Данные
012C	0000			Поле для записи числа
012E	0000			Поле для записи числа
130	0000			Поле для записи числа
132	0000			Поле для записи числа
134	0000			Поле для записи числа
136	0000			Поле для записи числа
0138	0000			Поле для записи числа
013A	0000			Поле для записи числа
013C	0000			Поле для записи числа
013E	0000			Поле для записи числа

Исходные данные см. Приложение 5.

### **2.3.3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ**

По заданию преподавателя подготовить и выполнить программы пересылки массива информации из одной области памяти в другую и выбрать из массива данных числа заданного интервала.

Исходные данные необходимо взять из Приложений 4 и 5 соответственно.

### **2.3.4. СОДЕРЖАНИЕ ОТЧЕТА**

1. Краткое описание цели работы и результатов самостоятельной подготовки к лабораторному занятию.
2. Тексты программ выполнения лабораторной работы.

### **2.3.5. КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Назовите назначение разрядов регистра признаков МП Intel 80x86?
2. По каким условиям записывается 1 в каждый из разрядов регистра признаков МП Intel 80x86?
3. Какие функции выполняют команды перехода?
4. Какие существуют способы изменения последовательности выполнения команд?
5. Как осуществляется выполнение команды перехода?
6. В каких байтах располагаются младшие и старшие разряды в командах перехода?
7. По алгоритму задачи необходимо принять решение «больше»? Какими командами можно это реализовать?
8. По алгоритму задачи необходимо принять решение «равно»? Какими командами можно это реализовать?
9. Как организуется цикл в системе команд МП Intel 80x86?
10. Какое максимальное число команд может быть в теле цикла?
11. Какое максимальное число команд имеет параметр цикла?
12. Как организуется работа с массивами?

## ПРИЛОЖЕНИЯ

### 1. Варианты заданий к лабораторной работе № 1

№ п/п	1-ое слагаемое	2-ое слагаемое	3-ое слагаемое
1	0009	2A4D	631E
2	0012	EF6E	0938
3	001B	0D2C	858B
4	0024	55BA	38DC
5	002D	9058	0D0C
6	0036	B89E	358A
7	003F	A721	2BA2
8	0048	4D36	15E9
9	0051	26F1	6C4F
10	005A	3CCD	3C9A
11	0063	B771	2F81
12	006C	2604	A694
13	0075	57E2	4CEA
14	007E	5361	8387
15	0087	ADF8	3F6E
16	0090	3190	2244
17	0099	B29D	048E
18	00A2	4F85	6E54
19	00AB	A869	32DF
20	00B4	8597	1C1D
21	00BD	A98A	51E5
22	00C6	7F14	2094
23	00CF	CEC3	033C
24	00D8	29A8	579E
25	00E1	4372	2703

## 2. Варианты заданий к лабораторной работе № 2

№ п/п	Исходный ряд чисел
1	046E,05E7,240F,3031,0820,1AF9,1F32
2	0931,4C05,3388,207A,10FB,0867,045D
3	22CC,18C0,2489,0566,165A,35BB,2030
4	0967,0DE8,002B,6ABB,229A,3B3C,1732
5	1D0D,3914,1093,1324,06ED,0887,34EA
6	1A5B,488C,000E,026B,30BF,106A,00B7
7	242C,0693,4D4F,2825,0DF3,3234,18E6
8	46D4,0984,45E8,0872,0D16,1158,3276
9	1033,06D8,02BD,74A4,038F,43B1,04F5
10	2EEE,1F3F,2A8D,2551,1E1C,0002,0A7A
11	2537,0339,217D,54D9,1FDB,0032,0003
12	1D09,310F,103B,2563,07F8,546B,0246
13	1DED,06AF,6FCB,1899,1799,2798,0EED
14	1081,2A1D,1BF1,3F90,3DCD,08CD,160D
15	0627,1DB4,1C39,1860,0DF5,1946,2F86
16	4354,0F0F,20FE,49AB,24EE,149E,0051
17	152B,1AF0,0F95,3F40,6FC2,0413,05EA
18	24A1,2AAA,08A7,0863,2E1E,1C70,23D0
19	0A00,119E,42E8,0875,2309,22A0,08E0
20	02AE,38C5,0136,2369,14C1,52B0,3258
21	3C94,0A89,2073,08F2,0DB2,2A88,3D49
22	12D2,0128,57B0,217B,3A2D,0E99,0CFF
23	14A0,0B17,17AE,1971,1C3B,270C,14CD
24	05E8,3198,0DAF,0BEE,0BFD,477B,2430
25	2CD0,2A84,11DF,1D28,1FC7,2F94,1B24

### 3. Варианты заданий к лабораторной работе № 2

№ п/п	1-ое слагаемое	2-ое слагаемое
1	D4241C879DAB	DB893E0731C5
2	14B24F878A00	C8804A82661C
3	B23CCD13CC0F	4BFA71E5F1B2
4	088BB20A9635	B3F59D213819
5	63599097F0DC	26FC7651829B
6	6C186C6EF7CB	6C96DA2F5B89
7	1A48269F6625	E4FFFDBF8B6F
8	5D6D9039AEF9	820DEBE78C8C
9	43ECE23D1FD2	23B3E1532C35
10	174291286446	986C92E74E01
11	6A80C9DD04AE	86ABBA91A754
12	0B552168FB5A	5D954BE52779
13	69B794532210	8D4C5D819B45
14	53B03825B43B	5DCDBF87A32E
15	6455FB87AA29	28FD82561B36
16	109587FE899B	B5945E3367AB
17	47B1374A7EEF	2311010B288F
18	36FF17C53AD3	BA06EA01D356
19	3C8E6EB84E89	1FA28C853490
20	DF76F5DD4A94	D7B37B6B581D
21	8D307B7D26FC	F01215625845
22	73913AE9B10A	6F4F8F9AA3D6
23	852ED0435357	4A1F4252C167
24	62CB381DBC11	35A415DC4937
25	7ACA989E5E6A	2AB5010D8AC2



#### 4. Варианты заданий к лабораторной работе № 3

№ п/п	Исходный массив
1	D424,1C87,9DAB,DB89,3E07,31C5,14B2,4F87,8A00
2	C880,4A82,661C,B23C,CD13,CC0F,4BFA,71E5,F1B2
3	088B,B20A,9635,B3F5,9D21,3819,C3D7,A7A7,FAF4
4	9AC0,0C85,D2CE,6359,9097,F0DC,26FC,7651,829B
5	AA82,63D5,CF4A,8BDE,DA7D,EEE1,6C18,6C6E,F7CB
6	6C96,DA2F,5B89,1A48,269F,6625,E4FF,FDBF,8B6F
7	C965,3DE9,594F,530B,1236,BC89,E701,1515,C1EB
8	5F81,B7A7,5C15,5D6D,9039,AEF9,820D,EBE7,8C8C
9	43EC,E23D,1FD2,23B3,E153,2C35,2623,B274,82BA
10	E41A,57D0,8A8C,6980,5C3C,498A,315B,0635,B3D5
11	F5B5,21D0,7233,A723,FCAC,FD94,1580,3CAB,3736
12	597B,54E3,9BC4,1742,9128,6446,986C,92E7,4E01
13	994D,C04A,95A0,AD54,DA35,F4FF,33C5,31C9,04F1
14	DFB3,1C47,EC07,6A80,C9DD,04AE,86AB,BA91,A754
15	0B55,2168,FB5A,5D95,4BE5,2779,69B7,9453,2210
16	8D4C,5D81,9B45,D690,2172,8E2F,312C,B261,DCC1
17	53B0,3825,B43B,5DCD,BF87,A32E,6455,FB87,AA29
18	28FD,8256,1B36,1095,87FE,899B,B594,5E33,67AB
19	47B1,374A,7EEF,2311,010B,288F,D596,08A0,E004
20	481B,7272,DB33,AA1C,0554,2623,A062,F76B,85B1
21	4CBE,0806,686B,A173,6D29,F909,95E5,8385,08B6
22	0F79,B738,D8B1,36FF,17C5,3AD3,BA06,EA01,D356
23	CCEA,BE87,F67E,9255,887D,E017,3C8E,6EB8,4E89
24	1FA2,8C85,3490,7F8D,148D,240B,438F,9341,D360
25	DF76,F5DD,4A94,D7B3,7B6B,581D,77E4,DD57,82B6

### 5. Варианты заданий к лабораторной работе № 3

№ п/п	Исходный массив	Нижний Предел	Верх- ний предел
1	D424,1C87,9DAB,DB89,3E07,31C5,14B2	24C3	4433
2	C880,4A82,661C,B23C,CD13,CC0F,4BFA	5624	5B3D
3	088B,B20A,9635,B3F5,9D21,3819,C3D7	7DBA	8041
4	9AC0,0C85,D2CE,6359,9097,F0DC,26FC	38CB	6F52
5	AA82,63D5,CF4A,8BDE,DA7D,EEE1,6C18	61F0	652E
6	6C96,DA2F,5B89,1A48,269F,6625,E4FF	7BA1	E2FB
7	C965,3DE9,594F,530B,1236,BC89,E701	0E69	1306
8	5F81,B7A7,5C15,5D6D,9039,AEF9,820D	5CEA	A93D
9	43EC,E23D,1FD2,23B3,E153,2C35,2623	5089	9DB5
10	E41A,57D0,8A8C,6980,5C3C,498A,315B	300	0587
11	F5B5,21D0,7233,A723,FCAC,FD94,1580	0CF6	3C18
12	597B,54E3,9BC4,1742,9128,6446,986C	1B3C	5963
13	994D,C04A,95A0,AD54,DA35,F4FF,33C5	00EB	2FA5
14	DFB3,1C47,EC07,6A80,C9DD,04AE,86AB	706F	AC03
15	0B55,2168,FB5A,5D95,4BE5,2779,69B7	1361	91A2
16	8D4C,5D81,9B45,D690,2172,8E2F,312C	80ED	9582
17	53B0,3825,B43B,5DCD,BF87,A32E,6455	7D16	BC2F
18	28FD,8256,1B36,1095,87FE,899B,B594	1B0E	42D0
19	47B1,374A,7EEF,2311,010B,288F,D596	064B	0732
20	481B,7272,DB33,AA1C,0554,2623,A062	6EA4	D3DB
21	4CBE,0806,686B,A173,6D29,F909,95E5	045A	7FF2
22	0F79,B738,D8B1,36FF,17C5,3AD3,BA06	A385	C614
23	CCEA,BE87,F67E,9255,887D,E017,3C8E	20B5	6A9C
24	1FA2,8C85,3490,7F8D,148D,240B,438F	42BD	50D4
25	DF76,F5DD,4A94,D7B3,7B6B,581D,77E4	6C8B	D494

## 6. Система команд для PC

### ОПЕРАЦИИ НАД ЦЕЛЫМИ ЧИСЛАМИ

#### *ПЕРЕСЫЛКА ДАННЫХ И АДРЕСОВ*

##### **Пересылка данных без преобразования:**

MOV	Пересылка операнда
PUSH	Запись операнда в стек
PUSHA	Запись в стек содержимого всех регистров
POP	Чтение операнда из стека
POPA	Чтение из стека содержимого всех регистров
XCHG	Обмен между регистрами или памятью и регистром
XLAT	Преобразование кодов
BSWAP	Перестановка байтов

##### **Пересылка данных с преобразованием:**

MOVSX	Пересылка байта или слова с расширением знака
MOVZX	Пересылка байта или слова с расширением нулями

##### **Ввод-вывод данных:**

IN	Ввод операнда из порта в аккумулятор
OUT	Вывод операнда из аккумулятора в порт

##### **Загрузка адреса и селектора:**

LEA	Загрузка эффективного адреса EA в регистр
LDS	Загрузка селектора в регистр DS
LES	Загрузка селектора в регистр ES
LFS	Загрузка селектора в регистр FS
LGS	Загрузка селектора в регистр GS
LSS	Загрузка селектора в регистр SS

## ***АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ***

### **Сложение:**

ADD	Сложение операндов
ADC	Сложение операндов с признаком CF (перенос)
XADD	Обмен операндами и сложение
INC	Инкремент операнда
AAA	ASCII-коррекция результата сложения
DAA	Десятичная коррекция результата сложения

### **Вычитание:**

SUB	Вычитание операндов
SBB	Вычитание операндов с признаком CF (заем)
DEC	Декремент операнда
CMR	Сравнение операндов
CMRCHG	Сравнение и обмен операндами
NEG	Изменение знака операнда (с переводом в дополнительный код)
AAS	ASCII-коррекция результата вычитания
DAS	Десятичная коррекция результата вычитания

### **Умножение:**

MUL	Беззнаковое умножение
IMUL	Знаковое (целочисленное) умножение
AMM	ASCII-коррекция результата умножения

### **Деление:**

DIV	Беззнаковое деление
IDIV	Знаковое (целочисленное) деление
AAD	ASCII-коррекция результата деления

### **Изменения разрядности путем расширения знака:**

CBW	Преобразование байта (AL) в слово (AX)
CWDE	Преобразование слова (AX) в двойное слово (EAX)
CWD	Преобразование слова (AX) в двойное слово (DX, AX)
CDQ	Преобразование двойного слова (EAX) в учетверенное слово (EDX, EAX)

### **ЛОГИЧЕСКИЕ ОПЕРАЦИИ И СДВИГИ**

#### **Логические операции:**

NOT	Инверсия операнда (логическое НЕ)
AND	Конъюнкция операндов (логическое И)
OR	Дизъюнкция операндов (логическое ИЛИ)
XOR	Неравнозначность операндов (исключающее ИЛИ)
TEST	Логическое сравнение операндов (установка признаков ZF, SF, PF)

#### **Сдвиги:**

SHL/SAL	Сдвиг влево
SHR	Логический сдвиг вправо
SAR	Арифметический сдвиг вправо
SHLD	Двухоперандный сдвиг влево
SHRD	Двухоперандный сдвиг вправо
ROL	Циклический сдвиг влево
ROR	Циклический сдвиг вправо
RCL	Циклический сдвиг влево через перенос (CF)
RCR	Циклический сдвиг вправо через перенос (CF)

#### **Операции с битами:**

BT	Проверка бита
BTS	Проверка и установка бита
BTR	Проверка и сброс бита

BTC	Проверка и инверсия бита
BSF	Прямое сканирование битов
BSR	Обратное сканирование битов

#### **Операции со строками символов:**

LODS	Загрузка символа в аккумулятор
STOS	Запись символа из аккумулятора
INS	Ввод символа
OUTS	Вывод символа
MOVS	Пересылка символа
CMPS	Сравнение символов
SCAS	Сканирование строки символов

## **ОПЕРАЦИИ УПРАВЛЕНИЯ**

### ***УПРАВЛЕНИЕ ПРОГРАММОЙ***

#### **Безусловная передача управления:**

JMP	Безусловный переход
CALL	Вызов подпрограммы
RET	Возврат из подпрограммы

#### **Условные переходы без учета знака:**

JA/JNBE	Переход, если выше (не ниже и не равно)
JAE/JNB/JNC	Переход, если (CF)=0: выше или равно (не ниже)
JB/JNAE/JC	Переход, если (CF)=1: ниже (не выше и не равно)
JBE/JNA	Переход, если ниже или равно (не выше)
JE/JZ	Переход, если равно (нуль): (ZF)=1
JNE/JNZ	Переход, если не равно (не нуль): (ZF)=0
JP/JPE	Переход, если четность: (PF)=1
JNP/JPO	Переход, если нечетность: (PF)=0

### Условные переходы с учетом знака:

JG/JNLE	Переход, если больше (не меньше и не равно)
JGE/JNL	Переход, если больше или равно (не меньше)
JL/JNGE	Переход, если меньше (не больше и не равно)
JLE/JNG	Переход, если меньше или равно (не больше)
JS	Переход, если отрицательно: (SF)=1
JNS	Переход, если положительно (SF)=0
JO	Переход, если переполнение: (OF)=1
JNO	Переход, если нет переполнения (OF)=0

### Условная установка байта:

SETA/SETNBE	Установка, если выше (не ниже и не равно)
SETAE/SETNB	Установка, если выше или равно (не ниже)
SETB/SETNAE	Установка, если ниже (не выше и не равно)
SETBE/SETNA	Установка, если ниже или равно (не выше)
SETE/SETZ	Установка, если равно (нуль): (ZF)=1
SETNE/SETNZ	Установка, если не равно (не нуль): (ZF)=0
SETP/SETPE	Установка, если четность: (PF)=1
SETNP/SETPO	Установка, если нечетность: (PF)=0
SETG/SETNLE	Установка, если не больше (не меньше и не равно)
SETGE/SETNL	Установка, если больше или равно (не меньше)
SETL/SETNGE	Установка, если меньше (не больше и не равно)
SETLE/SETNG	Установка, если меньше и не равно (не больше)
SETS	Установка, если отрицательно: (SF)=1
SETNS	Установка, если положительно: (SF)=0
SETO	Установка, если переполнение: (OF)=1
SETNO	Установка, если нет переполнения: (OF)=0

### Прерывания:

INT	Прерывание
INT0	Прерывание по переполнению: (OF)=1
INT3	Прерывание в контрольной точке

IRET	Возврат из подпрограммы прерывания
CLI	Запрещение прерываний
STI	Разрешение прерываний

#### **Организация циклов:**

LOOP	Реализация циклов, пока (ECX)≠0
LOOPE/LOOPZ	Реализация циклов, пока (ECX)≠0 или (ZF)=1
LOOPNE/LOOPNZ	Реализация циклов, пока (ECX)≠0 или (ZF)=0
JCXZ (JECXZ)	Реализация циклов, пока (CX)=0 или (ECX)=0

#### **Операции над признаками:**

LAHF	Загрузка признаков в регистр AH
SAHF	Запись содержимого AH в регистр признаков
PUSHF	Запись содержимого регистра признаков в стек
POPF	Чтение содержимого регистра признаков из стека
CLC	Сброс признака переноса: (CF)=0
STC	Установка признака переноса: (CF)=1
CMC	Инвертирование признака переноса (CF)
CLD	Сброс признака направления: (DF)=0
CTD	Установка признака направления: (DF)=1

#### ***ПОДДЕРЖКА ЯЗЫКА ВЫСОКОГО УРОВНЯ***

BOUND	Проверка границ массива
ENTER	Обращение к процедуре
LEAVE	Выход из процедуры

#### ***ОРГАНИЗАЦИЯ ЗАЩИТЫ ПАМЯТИ***

SGDT	Запись содержимого регистра таблицы глобальных дескрипторов
------	---



SLDT	Запись содержимого регистра таблицы локальных дескрипторов
SIDT	Запись содержимого регистра таблицы дескрипторов прерываний
STR	Запись содержимого регистра задачи
LGD	Загрузка регистра таблицы глобальных дескрипторов
LLDT	Загрузка регистра таблицы локальных дескрипторов
LIDT	Загрузка регистра таблицы дескрипторов прерываний
LTR	Загрузка регистра задачи
CLTS	Сброс признака переключения задачи: (TS)=0
ARPL	Коррекция запрошенного уровня привилегий
LAR	Загрузка прав доступа
LSL	Загрузка границы сегмента
VERR	Проверка доступности сегмента при чтении
VERRW	Проверка доступности сегмента при записи
LMSW	Загрузка слова состояния машины (MSW)
SMSW	Запись слова состояния машины (MSW)

### ***УПРАВЛЕНИЕ ПРОЦЕССОРОМ***

NOP	Отсутствие операции
HLT	Останов
MOV	Пересылка содержимого управляющих, отладчиков, тестирующих регистров
WAIT	Ожидание до поступления сигнала BUSY=1
INVD	Аннулирование содержимого кэш-памяти
WBINVD	Обратная запись и аннулирование кэш-памяти
INVLPG	Аннулирование входа в таблицу страниц

## ***ПРЕФИКСЫ***

LOCK	Блокировка магистрали
SEG	Замена сегмента
AS	Изменение разрядности адреса
OS	Изменение разрядности операнда
REP, REPE/REPZ, REPNE/REPZ	Повторение операций со строками символов

## **ОПЕРАЦИИ НАД ЧИСЛАМИ С ПЛАВАЮЩЕЙ ТОЧКОЙ**

### ***ПЕРЕСЫЛКА ДАННЫХ***

FLD	Загрузка вещественного числа
FILD	Загрузка целого числа
FBLD	Загрузка двоично-десятичного числа
FST	Запись в память вещественного числа
FIST	Запись в память целого числа
FSTP	Запись в память вещественного числа с выталкиванием из стека
FISTP	Запись в память целого числа с выталкиванием из стека
FBSTP	Запись в память двоично-десятичного числа с выталкиванием из стека
FXCH	Обмен данными между регистрами стека
FLDZ	Загрузка нуля +0.0
FLD1	Загрузка единицы +1.0
FLDPI	Загрузка $\pi$ (пи)
FLDL2T	Загрузка $\log_2 10$
FLDL2E	Загрузка $\log_2 e$

FLDLG2 Загрузка lg 2

FLDLN2 Загрузка ln 2

### ***АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ***

FADD Сложение вещественных чисел

FADDP Сложение вещественных чисел с выталкиванием из стека

FIADD Сложение целых чисел

FSUB Вычитание вещественных чисел

FSUBP Вычитание вещественных чисел с выталкиванием из стека

FSUBR Обратное вычитание вещественных чисел

FSUBRP Обратное вычитание вещественных чисел с выталкиванием из стека

FISUB Вычитание целых чисел

FISUBR Обратное вычитание целых чисел

FMUL Умножение вещественных чисел

FMULP Умножение вещественных чисел с выталкиванием из стека

FIMUL Умножение целых чисел

FDIV Деление вещественных чисел

FDIVP Деление вещественных чисел с выталкиванием из стека

FDIVR Обратное деление вещественных чисел

FDIVRP Обратное деление вещественных чисел с выталкиванием из стека

FIDIV Деление целых чисел

FIDIVR Обратное деление целых чисел

### ***ОПЕРАЦИИ СРАВНЕНИЯ***

FCOM	Сравнение вещественных чисел
FCOMP	Сравнение вещественных чисел с выталкиванием из стека
FCOMPP	Сравнение вещественных чисел с двойным выталкиванием из стека
FICOM	Сравнение целых чисел
FICOMP	Сравнение целых чисел с выталкиванием из стека
FTST	Сравнение с нулем
FUCOM	Сравнение неупорядоченных чисел
FUCOMP	Сравнение неупорядоченных чисел с выталкиванием из стека
FUCOMPP	Сравнение неупорядоченных чисел с двойным выталкиванием из стека
FXAM	Анализ вещественного числа

### ***СПЕЦИАЛЬНЫЕ ОПЕРАЦИИ***

FSQRT	Вычисление квадратного корня
FSCALE	Масштабирование (изменение порядка числа)
FXTRACT	Выделение мантиссы и порядка
FPREM	Нахождение частичного остатка от деления
FPREM1	Нахождение частичного остатка в стандарте IEEE
FRNDINT	Округление до целого числа
FABS	Получение абсолютного значения
FCHS	Изменение знака числа
FSIN	Вычисление синуса
FCOS	Вычисление косинуса
FSINCOS	Вычисление синуса и косинуса
FPTAN	Вычисление частичного тангенса
FPATAN	Вычисление частичного арктангенса
F2XM1	Вычисление функции $2^x - 1$

FYL2X     Вычисление функции  $Y \cdot \log_2 X$   
FYL2XP1     Вычисление функции  $Y \cdot \log_2 (X + 1)$

### ***ОПЕРАЦИИ УПРАВЛЕНИЯ***

FINIT     Инициализация FPU  
FSTWAX     Передача слова состояния в регистр AX  
FSTSW     Передача слова состояния в память  
FLDCW     Загрузка управляющего слова  
FCLEX     Сброс флагов в регистре состояния SR  
FSTENV     Запись в память содержимого вспомогательных  
              регистров  
FLDENV     Загрузка содержимого вспомогательных регистров  
FSAVE     Запись в память содержимого вспомогательных  
              регистров и арифметического стека  
FRSTOR     Загрузка содержимого вспомогательных регистров и  
              арифметического стека  
FINCSTP     Инкремент содержимого указателя стека  
FDECSTP     Декремент содержимого указателя стека  
FFREE     Освобождение регистра ST(i)  
FWAIT (WAIT)     Ожидание готовности процессора с плавающей  
                      точкой

## Литература

1. *Саримов, Н.Н.* Программирование для процессора Intel. Методические указания к выполнению лабораторных работ по курсу «Электронные вычислительные машины» / Н.Н.Саримов. – Казань: Казан. матем. общ-во, 1999. - 86 с.
2. *Крикун, Н.Г.* Микропроцессоры и микро-ЭВМ: Метод, указания / Н.Г. Крикун, Р.Н. Гайнуллин, А.Р. Герке, С.М. Вайнер, В.А. Фафурин. – Казань: Казан. гос. технол. ун-т, 1996. - ч. 1. 25 с.
3. *Овечкин, Ю.А.* Микропроцессоры: Справочное пособие / Ю.А. Овечкин. Л.: Судостроение, 1987. - 519 с.
4. *Гилмор, Ч.* Введение в микропроцессорную технику / Ч. Гилмор. - М.: Мир, 1984. - 331 с.
5. *Преснухин, Л.Н.* Микропроцессоры / Л.Н. Преснухина. М.: Высшая школа, 1986. - Т. 3.350 с.
6. *Богумирский, С.П.* Руководство пользователя ПЭСМ в 2-х томах / С.П. Богумирский. - СП: «Ассоциация ОЛКО», 1992. - 764 с.
7. *Мячев, И.С.* Интерфейсы средств вычислительной техники / И.С. Мячев. - М.: Радио и связь, 1993. - 254 с.
8. *Бродин, В.Б.* Микропроцессор i486. Архитектура, программирование, интерфейс / В.Б. Бродин, И.И. Шагурин. - М.: «Диалог-мифи», 1993. - 240 с.

**Учебное издание**

**Сечина Галина Павловна**

**МИКРОПРОЦЕССОРЫ И  
МИКРО-ЭВМ**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
К ЛАБОРАТОРНЫМ РАБОТАМ**

**ЧАСТЬ I**

Корректор Габдурахимова Т.М.  
Худ.редактор Федорова Л.Г.

Сдано в набор 02.05.2012  
Подписано в печать 04.06.2012.  
Бумага писчая. Гарнитура Таймс.  
Усл.печ.л. 2,4. Тираж 100.  
Заказ №33.

НХТИ (филиал) ФГОУ ВПО «КНИТУ»,  
г.Нижнекамск, 423570, ул.30 лет Победы, д.5а.

