Министерство образования и науки Российской Федерации
**Нижнекамский 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 II

**Нижнекамск**
**2012**

**УДК 621.38**
**С 33**

Печатаются 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 II / Г.П. Сечина. – Нижнекамск : Нижнекамский химико-технологический институт (филиал) ФГБОУ ВПО «КНИТУ», 2012. - 46 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

Данное пособие продолжает знакомить студентов с методикой программирования в кодах микропроцессора Intel 80x86.

В пособии приведены программы способов организации условных переходов в микропроцессоре Intel 80x86, маскирования данных, представлены исследования особенностей записи и обращения к подпрограммам, методы использования стека при создании подпрограмм, способы реализации деления и умножения целых двоичных чисел.

При разработке пособия использовались работы [1,2].

## Лабораторная работа 1

# ОРГАНИЗАЦИЯ УСЛОВНЫХ ПЕРЕХОДОВ

Цель работы: изучение программных способов организации условных переходов в микро-ЭВМ.

## ОБЩИЕ СВЕДЕНИЯ

**Флаговый регистр.** Девять из 16 бит флагового регистра являются активными и определяют текущее состояние машины и результаты выполнения команд. Многие арифметические операции и команды сравнения изменяют состояние флагов. Назначение флаговых битов следующее:

*Флаг Назначение*

**O** (Переполнение) Указывает на переполнение старшего бита при арифметических командах.

**D** (Направление) Обозначает левое или правое направление пересылки или сравнения строковых данных.

**I** (Прерывание) Указывает на возможность внешних прерываний.

**T** (Пошаговый режим) Обеспечивает возможность работы процессора в пошаговом режиме.

**S** (Знак) Содержит результирующий знак при арифметических операциях (0 - плюс, 1 - минус).

**Z** (Ноль) Показывает результат арифметических операций и операций сравнения (0 - ненулевой, 1 - нулевой результат).

**A** (Внешний перенос) Содержит перенос из 3-го бита для 8-битовых данных, используется для специальных арифметических операций.

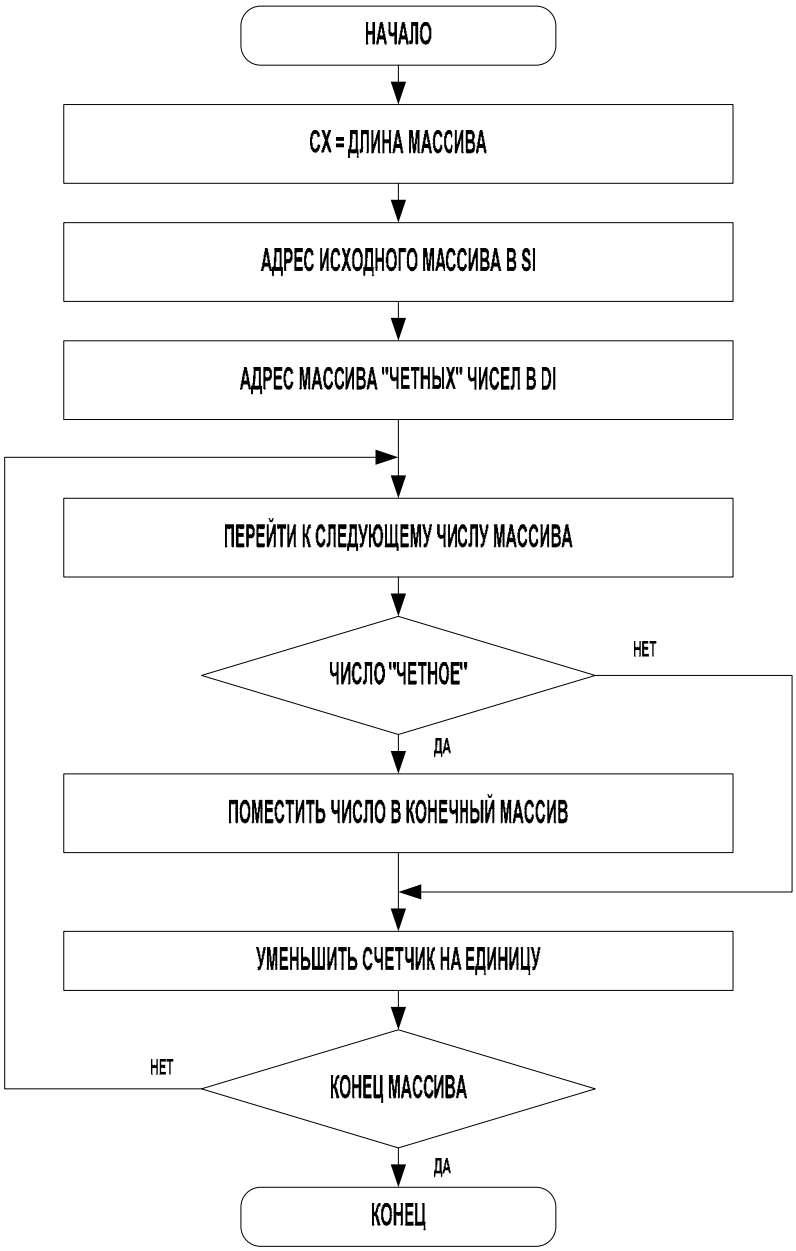
**P** (Контроль четности) Показывает четность младших 8-битовых данных (1 - четное, 0 - нечетное число).

**C** (Перенос) Содержит перенос из старшего бита после арифметических операций, а также последний бит при сдвигах или циклических сдвигах.

В программах флаговый регистр явно не используется, поэтому не имеется его мнемонического обозначения.

## ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ УСЛОВНЫХ ПЕРЕХОДОВ

**Задание:** Из имеющегося массива чисел осуществить выборку тех, которые имеют четное число единиц.


НАЧАЛО
CX = ДЛИНА МАССИВА
АДРЕС ИСХОДНОГО МАССИВА В SI
АДРЕС МАССИВА "ЧЕТНЫХ" ЧИСЕЛ В DI
ПЕРЕЙТИ К СЛЕДУЮЩЕМУ ЧИСЛУ МАССИВА
ЧИСЛО "ЧЕТНОЕ"
НЕТ
ДА
ПОМЕСТИТЬ ЧИСЛО В КОНЕЧНЫЙ МАССИВ
УМЕНЬШИТЬ СЧЕТЧИК НА ЕДИНИЦУ
КОНЕЦ МАССИВА
НЕТ
ДА
КОНЕЦ

**Программная модель**

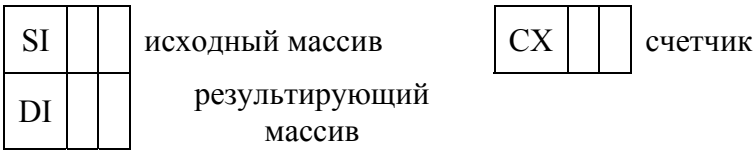



**Таблица 1 - Программа**

| Адрес | Машинный код | Метка | Мнемокод | Комментарий |
|---|---|---|---|---|
| cs:0100 | B90900 | | mov cx,0009 | счетчик – длина массива |
| cs:0103 | BE1601 | | mov si,0116 | адрес исходного массива |
| cs:0106 | BF1F01 | | mov di,011F | адрес результирующего массива |
| cs:0109 | B80000 | m1 | mov ax,0000 | обнулим AX |
| cs:010C | 0204 | | add al,[si] | добавим к AL байт по смещению [SI] |
| cs:010E | 7B01 | | jnp 0111 | если флаг паритета сброшен, то прыгаем на m3 |
| cs:0110 | AA | | stosb | иначе сохраняем AL в память по смещению [DI] |
| cs:0111 | 46 | m3 | inc si | увеличим SI, чтобы он указывал на следующий байт |

| Адрес | Машинный код | Метка | Мнемокод | Комментарий |
|---|---|---|---|---|
| cs:0112 | E2F5 | | loop 0109 | |
| cs:0114 | CD20 | | int 20 | |
| cs:0116<br>cs:0118<br>cs:011A<br>cs:011D | 0102<br>0304<br>050607<br>0809 | | | массив исходных данных |
| cs:011F<br>cs:0121<br>cs:0123<br>cs:0125<br>cs:0127 | 0000<br>0000<br>0000<br>0000<br>0000 | | | массив чисел с четным паритетом |

## 1. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить материалы, изложенные в п.1 и лекции по теме «Команды перехода».
2. Получить индивидуальное задание на выполнение программирования.
3. Выполнить работу по программированию в соответствии с полученным заданием с практической отработкой программы.
4. Оформить отчет.

## 2. СОДЕРЖАНИЕ ОТЧЕТА

1. Задание на выполнение лабораторной работы.
2. Программная модель (использование РОНов и памяти при решении задачи, алгоритм и программа).
3. Исходные данные, используемые при решении программы и полученные результаты.

## 3. ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ

1. Какие бывают команды перехода?
2. Перечислите команды условных переходов.
3. С помощью каких регистров признаков осуществляется каждая из них?
4. Каков формат команд перехода, и к какому способу адресации они относятся?
5. По каким условиям записывается 1 в каждый из разрядов регистра состояния МП?

## Лабораторная работа 2

# МАСКИРОВАНИЕ ДАННЫХ

Цель работы: изучение программных способов маскирования данных.

## ОБЩИЕ СВЕДЕНИЯ

Во многих случаях при выполнении программ необходимо проверять или изменять (маскировать) состояние одного или нескольких разрядов числа. Это можно осуществить с помощью следующих команд логических операций:

1.*Логического умножения*

| Код | 1-й операнд | 2-й операнд (маска) | Изменяемые флаги |
|---|---|---|---|
| and | r<br>r,m | r,m<br>v | C,O,P,S,Z |

где r – операнд, ссылающийся на регистр;

m – операнд, ссылающийся на область памяти;

v – непосредственное значение операнда.

Очищает разряд числа, если в соответствующем разряде маски записан 0, иначе не изменяет его.

2.*Логического сложения*

| Код | 1-й операнд | 2-й операнд (маска) | Изменяемые флаги |
|---|---|---|---|
| or | r<br>r,m | r,m<br>v | C,O,P,S,Z |

Устанавливает разряд числа в 1, если в таком же разряде маски будет записана 1, иначе не изменяет его.

3.*Логического «исключающего ИЛИ»*

| Код | 1-й операнд | 2-й операнд (маска) | Изменяемые флаги |
|---|---|---|---|
| xor | r<br>r,m | r,m<br>v | C,O,P,S,Z |

Инвертирует содержимое разряда числа, если в соответствующем разряде маски записана 1, иначе не изменяет его.

Кроме того есть такая вспомогательная команда как test, которая выполняет проверку байта или слова на определенную битовую комбинацию. Действует как команда and, но не изменяет операнда.

## 1. ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ МАСКИРОВАНИЯ

**Задание 1:** Загрузить программным путем в память число 0A и число 21. Найти слово маски такое, чтобы сумма чисел была равна 1F, поместить его в память, произвести операцию маскирования командой «лог.исключающее ИЛИ». Проверить результат.

РЕШЕНИЕ:

$0A_{16}=10_{10}=1010_2$

$+21_{16}=33_{10}=100001_2$

---

$2B_{16}=43_{10}=101011_2$

маска $34_{16}=52_{10}=110100_2$

---

$1F_{16}=31_{10}=011111_2$

**Таблица 2 - Программа 1**

| Адрес | Машинный код | Метка | Мнемокод | Комментарий |
|---|---|---|---|---|
| cs:0100 | BE1D01 | | mov si,011D | загрузим в SI смещение на первый операнд |
| cs:0103 | C7040A00 | | mov word ptr [si],000A | загрузим в память по смещению [SI] первый операнд |
| cs:0107 | 8B04 | | mov ax,[si] | загрузим в AX из памяти первый операнд |
| cs:0109 | BE1E01 | | mov si,011E | загрузим в SI смещение на второй операнд |
| cs:010C | C7042100 | | mov word ptr [si],0021 | загрузим в память по смещению [SI] второй операнд |
| cs:0110 | 0304 | | add ax,[si] | добавим к AX второй операнд |
| cs:0112 | BE1F01 | | mov si,011F | загрузим в SI смещение на слово маски |
| cs:0115 | C7043400 | | mov word ptr [si],0034 | загрузим в память по смещению [SI] слово маски |

| Адрес | Машинный код | Метка | Мнемокод | Комментарий |
|---|---|---|---|---|
| cs:0119 | 3304 | | xor ax,[si] | «исключающее ИЛИ» для суммы в регистре AX и маски |
| cs:011B | CD20 | | int 20 | |
| cs:011D | 0000 | | | |
| cs:011F | 0000 | | | |

**Задание 2**: Из имеющегося массива чисел осуществить выборку тех, которые имеют единицы в 5 и 1 разрядах.

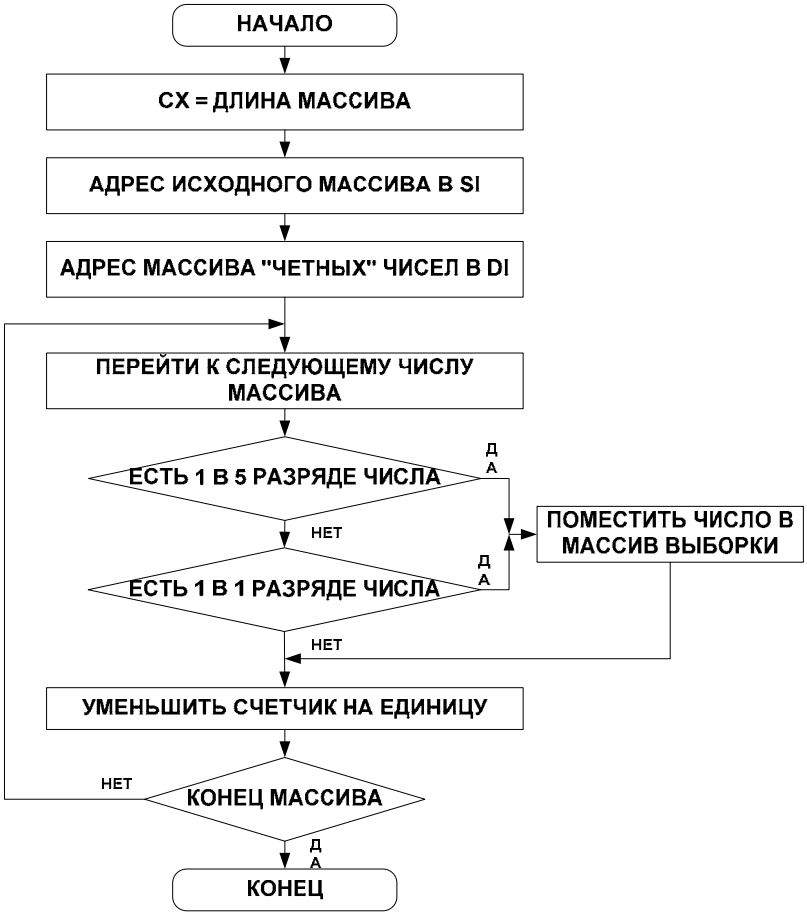

НАЧАЛО
CX = ДЛИНА МАССИВА
АДРЕС ИСХОДНОГО МАССИВА В SI
АДРЕС МАССИВА "ЧЕТНЫХ" ЧИСЕЛ В DI
ПЕРЕЙТИ К СЛЕДУЮЩЕМУ ЧИСЛУ МАССИВА
ЕСТЬ 1 В 5 РАЗРЯДЕ ЧИСЛА
ДА
НЕТ
ЕСТЬ 1 В 1 РАЗРЯДЕ ЧИСЛА
ДА
ПОМЕСТИТЬ ЧИСЛО В МАССИВ ВЫБОРКИ
НЕТ
УМЕНЬШИТЬ СЧЕТЧИК НА ЕДИНИЦУ
НЕТ
КОНЕЦ МАССИВА
ДА
КОНЕЦ

**Таблица 3 - Программа 2**

| Адрес | Машинный код | Метка | Мнемокод | Комментарий |
|---|---|---|---|---|
| cs:0100 | B90700 | | mov cx,0007 | |
| cs:0103 | BE1901 | | mov si,0119 | |
| cs:0106 | BF2001 | | mov di,0120 | |
| cs:0109 | B311 | | mov bl,11 | маска |
| cs:010B | AC | m1 | lodsb | загрузим байт в AL |
| cs:010C | 22C3 | | and al,bl | используем маску |
| cs:010E | 3AC3 | | cmp al,bl | сравним результат с маской |
| cs:0110 | 7503 | | jne 0115 | если не равен, то прыгаем на m3 иначе снова загружаем его в AL и сохраняем в результирующий массив |
| cs:0112 | 4E | | dec si | |
| cs:0113 | AC | | lodsb | |
| cs:0114 | AA | | stosb | |
| cs:0115 | E2F4 | m3 | loop 010B | |
| cs:0117 | CD20 | | int 20 | |
| cs:0119 | 1011 | | | |
| cs:011B | 1213 | | | |
| cs:011D | 1415 | | | |
| cs:011F | 16 | | | |
| cs:0120 | 0000 | | | |
| cs:0122 | 0000 | | | |
| cs:0124 | 0000 | | | |
| cs:0126 | 0000 | | | |

## 2. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить материалы, изложенные в п.1 и лекции по теме «Логические команды».
2. Получить индивидуальное задание на выполнение программирования.
3. Выполнить работу по программированию в соответствии с полученным заданием с практической отработкой программы.
4. Оформить отчет.

## 3. СОДЕРЖАНИЕ ОТЧЕТА

1. Задание на выполнение лабораторной работы.
2. Программная модель (использование РОНов и памяти при решении задачи, алгоритм и программа).
3. Исходные данные, используемые при решении программы и полученные результаты.

## ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ

1. Перечислите виды логических операций, выполняемые МП.
2. Для каких целей используется логическая операция «XOR»?
3. Какие разряды регистра состояния задействованы при выполнении логических команд?
4. Куда помещается результат после выполнения команд AND, OR, XOR?
5. Напишите таблицы истинности для команд «лог.И», «лог.ИЛИ», «лог.исключающее ИЛИ».

# Лабораторная работа 3

# ПОДПРОГРАММА И СТЕК

Цель работы: исследование особенностей записи и обращения к подпрограммам, изучение методов использования стека при создании программ.

## 1. ОБЩИЕ СВЕДЕНИЯ

Учитывая ограниченные возможности памяти при разработке программ, нужно стараться сделать их как можно короче. С этой целью часть программы, которые неоднократно повторяются, или программы, которые часто используются, могут быть использованы в виде подпрограмм - последовательностей команд, выполнение которых может быть вызвано из любого места программы любое количество раз. Процесс передачи управления к подпрограмме называется ее вызовом.

Для вызова подпрограмм и возврата из них используются команды CALL<A2>,<A1> и RET.

При работе с подпрограммами используется стек МП. Стек - специально организованная область ОЗУ, используемая для временного сохранения данных или адресов. Число, записанное в стек последним извлекается из него первым. По принятой структуре организации com-программ, стек размещается в конце сегмента, т.е. вершина стека размещается по адресу CS:FFFF.

Команда RET помещает в программный счетчик последнее записанное на данный момент в стеке число. После этого выполнение программы будет осуществляться с этого адреса. Любая подпрограмма должна заканчиваться командой RET.

Автоматическое сохранение и восстановление адреса основной программы при выполнении подпрограмм позволяет сделать подпрограммы вложенными, то есть осуществить вызов одной подпрограммы из другой. Уровень вложенности определяется размером стека.

Существуют также команды условного вызова подпрограмм и возврата из них. Они позволяют вызвать подпрограмму и возвратиться из нее по определенному состоянию заданных разрядов регистра признаков (аналогично командам условного перехода) без использования дополнительных команд.

Помимо команд вызова подпрограмм и возврата из них со стеком можно обмениваться информацией с помощью команд PUSH<R> (записать в стек содержание обозначенного регистра МП БИС) и POP<R> (записать данные из стека в обозначенный регистр МП БИС).

## 2. ПРОГРАММА ИССЛЕДОВАНИЯ ПРОЦЕССА ВЫПОЛНЕНИЯ КОМАНД ВЫЗОВА И ВОЗВРАТА ИЗ ПОДПРОГРАММ, А ТАКЖЕ КОМАНД РАБОТЫ СО СТЕКОМ

**Таблица 4 - Программа**

| Адрес | Машинный код | Метка | Мнемокод | Комментарий |
|---|---|---|---|---|
| cs:0100 | E80200 | | call 0105 | вызов подпрограммы |
| cs:0103 | CD20 | | int 20 | |
| cs:0105 | 9C | m1 | pushf | сохраняем в стеке регистр флагов |

| Адрес | Машинный код | Метка | Мнемокод | Комментарий |
|---|---|---|---|---|
| cs:0106 | 53 | | push bx | сохраняем в стеке регистр BX |
| cs:0107 | 52 | | push dx | сохраняем в стеке регистр DX |
| cs:0108 | 51 | | push cx | сохраняем в стеке регистр CX |
| cs:0109 | B80500 | | mov ax,5 | |
| cs:010C | 8BD8 | | mov bx,ax | |
| cs:010E | 03C0 | | add ax,ax | |
| cs:0110 | 8BD0 | | mov dx,ax | |
| cs:0112 | 8BC8 | | mov cx,ax | |
| cs:0114 | 59 | | pop cx | извлекаем из стека регистр CX |
| cs:0115 | 5A | | pop dx | извлекаем из стека регистр DX |
| cs:0116 | 5B | | pop bx | извлекаем из стека регистр BX |
| cs:0117 | 9D | | popf | извлекаем из стека регистр флагов |
| cs:0118 | C3 | | ret | возврат из подпрограммы |

**Задание :** загрузить программным путем массив чисел. Сделать выборку чисел от 2 до 6 и возвести их в квадрат.


НАЧАЛО
CX = ДЛИНА МАССИВА
АДРЕС ИСХОДНОГО МАССИВА В SI
АДРЕС МАССИВА "ЧЕТНЫХ" ЧИСЕЛ В DI
ПЕРЕЙТИ К СЛЕДУЮЩЕМУ ЧИСЛУ МАССИВА
ЧИСЛО ИЗ МАССИВА < 2
ДА
НЕТ
ЧИСЛО ИЗ МАССИВА > 6
ДА
ВЫЗОВ ПОДПРОГРАММЫ ВЫЧИСЛЕНИЯ ПРОГРАММЫ
УМЕНЬШИТЬ СЧЕТЧИК НА ЕДИНИЦУ
НЕТ
КОНЕЦ МАССИВА
ДА
КОНЕЦ

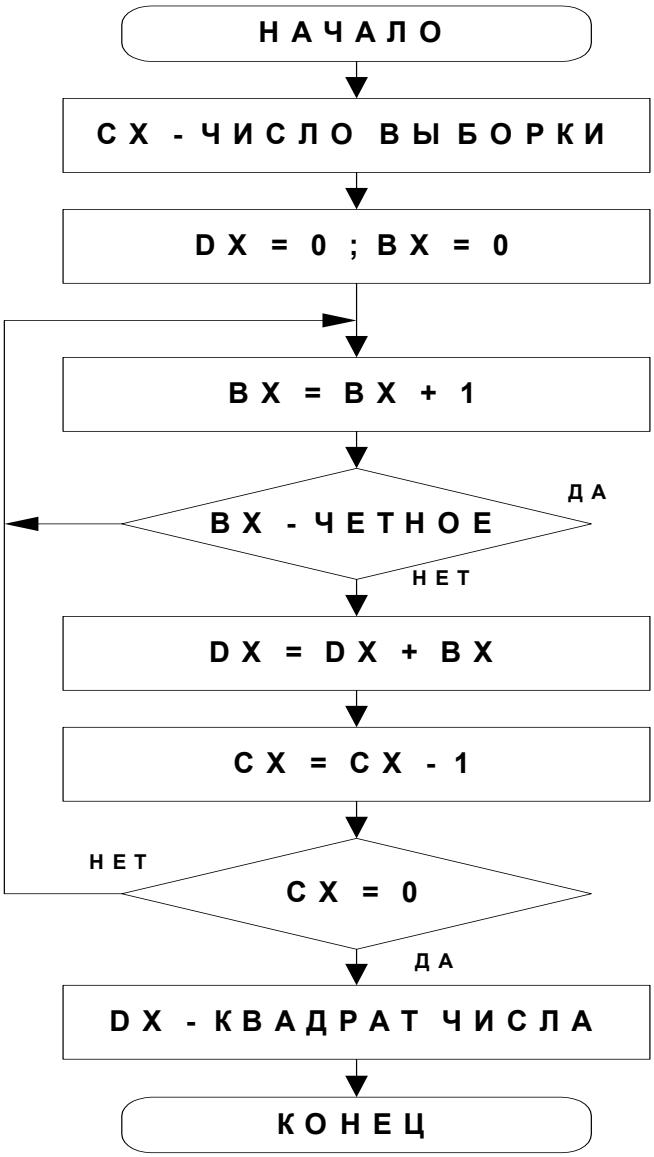

НАЧАЛО
СХ - ЧИСЛО ВЫБОРКИ
DX = 0 ; BX = 0
BX = BX + 1
ДА
BX - ЧЕТНОЕ
НЕТ
DX = DX + BX
CX = CX - 1
НЕТ
CX = 0
ДА
DX - КВАДРАТ ЧИСЛА
КОНЕЦ

**Таблица 5 - Программа 2**

| Адрес | Машинный код | Метка | Мнемокод | Комментарий |
|---|---|---|---|---|
| cs:0100 | B90A00 | | mov cx,000A | |
| cs:0103 | BE3501 | | mov si,0135 | |
| cs:0106 | BF4901 | | mov di,0149 | |
| cs:0109 | AD | m1 | lodsw | загружаем из памяти слово |
| cs:010A | 3D0200 | | cmp ax,0002 | сравниваем его с нижней границей |
| cs:010D | 7209 | | jb 0118 | если меньше, то прыгаем на m2 |
| cs:010F | 3D0600 | | cmp ax,0006 | сравниваем его с верхней границей |
| cs:0112 | 7704 | | ja 0118 | если больше, то прыгаем на m2 |
| cs:0114 | E80500 | | call 011C | вызываем подпрограмму |
| cs:0117 | AB | | stosw | сохраняем результат в памяти |
| cs:0118 | E2EF | m2 | loop 0109 | |
| cs:011A | CD20 | | int 20 | |
| | | sq | | подпрограмма возводит в квадрат содержимое AX |
| cs:011C | 51 | | push cx | сохраняем в стеке CX |

| Адрес | Машинный код | Метка | Мнемокод | Комментарий |
|---|---|---|---|---|
| cs:011D | 8BC8 | | mov cx,ax | счетчик |
| cs:011F | BA0000 | | mov dx,0000 | результат |
| cs:0122 | BB0000 | a1 | mov bx,0000 | счетчик |
| cs:0125 | 43 | | inc bx | увеличим счётчик BX на 1 |
| cs:0126 | 8BC3 | | mov ax,bx | |
| cs:0128 | 250100 | | and ax,0001 | проверяем BX на чётность |
| cs:012B | 74F8 | | je 0125 | если BX - чётное, то прыгаем на a1 |
| cs:012D | 03D3 | | add dx,bx | добавим к результату очередное нечётное число |
| cs:012F | E2F4 | | loop 0125 | |
| cs:0131 | 8BC2 | | mov ax,dx | перенесём результат в AX |
| cs:0133 | 59 | | pop cx | восстановим CX |
| cs:0134 | C3 | | ret | возврат из подпрограммы |
| cs:0135 | 0000 | | | |
| cs:0137 | 0100 | | | |
| cs:0139 | 0200 | | | |
| cs:013B | 0300 | | | |
| cs:013D | 0400 | | | |
| cs:013F | 050006 | | | |
| cs:0142 | 0007 | | | |
| cs:0144 | 0008 | | | |
| cs:0146 | 0009 | | | |

| Адрес | Машин-ный код | Ме тка | Мнемокод | Комментарий |
|---|---|---|---|---|
| cs:0148 | 0000 | | | |
| cs:014A | 0000 | | | |
| cs:014C | 0000 | | | |
| cs:014E | 0000 | | | |
| cs:0150 | 0000 | | | |
| cs:0152 | 0000 | | | |
| cs:0154 | 0000 | | | |
| cs:0156 | 0000 | | | |
| cs:0158 | 0000 | | | |

## 3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить материалы, изложенные в пунктах 1,2 и лекции по теме «Стек и подпрограмма».
2. Выполнить программу по командам, используя режим «Отладки». После каждой команды проверить содержимое всех регистров МП.
3. Заменить в программе команду POPF на команду NOP и проследить, как будет выполняться программа.
4. Получить индивидуальное задание на выполнение программирования с использованием стека.
5. Выполнить работу по программированию в соответствии с полученным заданием и практической отработкой программы.
6. Оформить отчет.

## 4. СОДЕРЖАНИЕ ОТЧЕТА

1. Задание на выполнение лабораторной работы.
2. Программная модель (использование РОНов и памяти при решении задачи, алгоритм и программа).
3. Исходные данные, используемые при решении программы и полученные результаты.

## ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ

1. Укажите порядок выполнения команды RET.
2. Сравните процесс выполнения команд CALL и RET.
3. В какой последовательности сохраняется и извлекается содержимое регистров МП в подпрограмме?
4. Как будет выполняться программа, если вместо команды POPF в ней будет записана команда NOP?

# Лабораторная работа 4

## ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ДЕЛЕНИЯ ЦЕЛЫХ ДВОИЧНЫХ ЧИСЕЛ

Цель работы: освоение способов программной реализации деления целых двоичных чисел.

## 1. ОБЩИЕ СВЕДЕНИЯ

Деление двоичных чисел производится так же, как и деление десятичных чисел. Однако деление двоичных чисел осуществляется проще, так как использование только двух цифр (0 и 1) исключает в каждом цикле деления необходимость определения числа делителей, содержащихся в текущей части делимого (остатке). Для определения очередной цифры частного достаточно только сравнить текущую часть делимого с делителем.

Пусть требуется разделить однобайтное число без знака X= 11010011 на однобайтное число без знака У=00001011. Вручную деление осуществляется следующим образом:

```
 11010011 | 1011
 1011       10011
   10001
    1011
     1101
     1011
       10
```

Рисунок - 1

Выделяется часть делимого, начиная со старшего разряда таким образом, чтобы она была не меньше делителя, и из нее вычитался делитель, а в частное записывается 1. К остатку, полученному из вычитания, сносится следующая цифра делимого и производится сравнение полученного таким образом остатка с делителем. Если остаток не меньше делителя, то в

следующий более младший разряд частного записывается 1. Если остаток меньше делителя - записывается 0. Затем к остатку добавляется следующая цифра делителя и снова производится сравнение остатка с делителем для определения следующей цифры частного. Аналогично производится операции, до тех пор, когда используются все цифры делимого. При делении заданных чисел получается частное Z=10011 и остаток f=10.

Из приведенного примера видно, что при ручном делении часть операций фиксируется на бумаге, то есть выполняется как бы явно (вычитание из текущего остатка делителя, если остаток не меньше делителя), а часть операций выполняется в уме, то есть как бы неявно (выделение текущего остатка с делителем).

При организации автоматического процесса деления все операции должны выполняться явно. Процесс деления тех же чисел с явным выполнением всех операций приведен на рис. 2.

Здесь после образования каждого текущего остатка, полученного добавлением очередной цифры делимого к остатку, производится вычитание из текущего остатка делителя. Если результат вычитания не меньше 0, то в очередной разряд записывается 1. Если результат вычитания - меньше 0, то в очередной разряд частного записывается 0 и производится восстановление остатка путем прибавления к отрицательному остатку делителя. Затем к остатку добавляется очередная цифра делимого, и операции определения очередных цифр частного повторяются аналогично.

```
 11010011 | 1011   .
 1011       00010011
 1010                  0
 1011
   11
  1011
 1000                  0
  1011
   110
   1011
    101                0
   1011
    1101
     1011
      100              1
     1011
       111             0
      1011
       1000
        1011
         11            0
        1011
        10001
          1011
           1101        1
            1011
              10       1
```

Рисунок - 2

## 2. ПРОГРАММИРОВАНИЕ ДЕЛЕНИЯ ЦЕЛЫХ ДВОИЧНЫХ ЧИСЕЛ

Рассмотрим два способа реализации программирования деления с явным и неявным счетчиками.

*2.1 С явным счетчиком*

**Таблица 6 - Программа**

| Адрес | Машинный код | Метка | Мнемокод | Комментарий |
|---|---|---|---|---|
| cs:0100 | B80800 | | mov | делимое |
| cs:0103 | BA0000 | | ax,0008 | частное |
| cs:0106 | BB0000 | | mov | остаток |
| cs:0109 | B91000 | | dx,0000 | счетчик |
| cs:010C | D1E0 | m0 | mov | сдвиг делимого |
| cs:010E | D1D3 | | bx,0000 | влево |
| cs:0110 | 83EB03 | | mov | с переносом |
| cs:0113 | 7903 | | cx,0010 | старшего бита в |
| cs:0115 | 83C303 | | shl ax,1 | остаток |
| cs:0118 | F5 | m1 | rcl bx,1 | вычитаем из |
| cs:0119 | D1D2 | | sub | остатка делитель |
| cs:011B | E2EF | | bx,0003 | если результат |
| cs:011D | CD20 | | jns 0118 | отрицательный, |
| | | | add | то |
| | | | bx,0003 | восстанавливаем |
| | | | cmc | остаток |
| | | | rcl dx,1 | инвертируем |
| | | | loop | флаг C |
| | | | 010C | получаем |
| | | | int 20 | очередной |
| | | | | разряд частного |
| | | | | если деление не |
| | | | | закончено, то |
| | | | | идём на m0 |

*2.1 С неявным счетчиком*

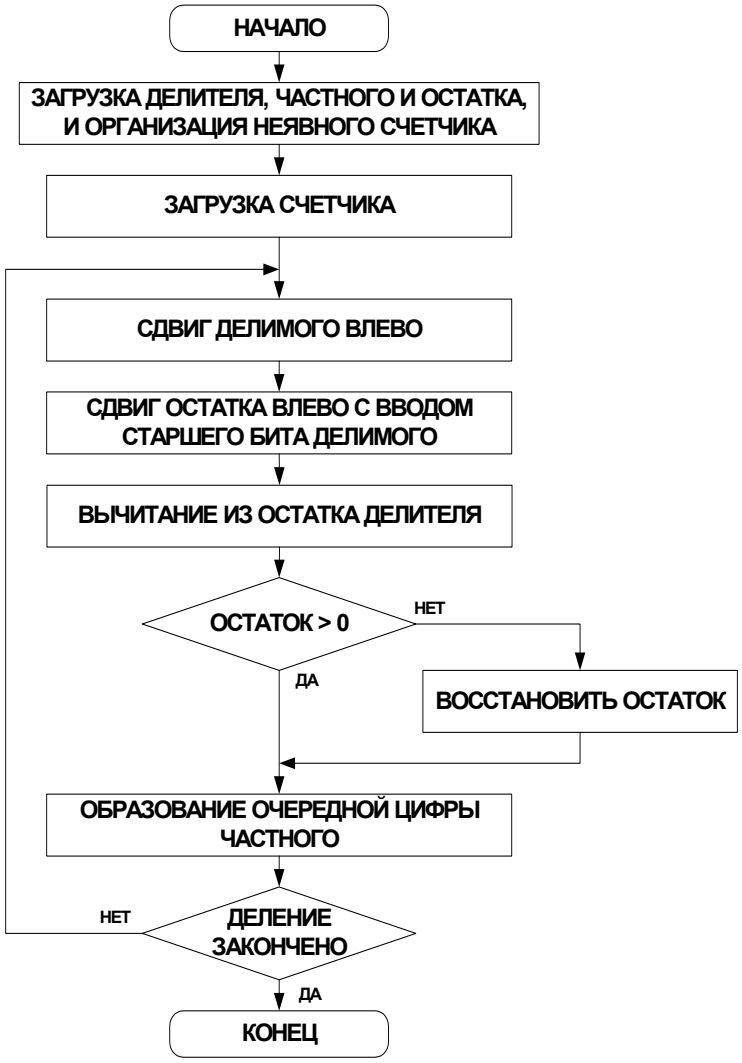


**Таблица 7 - Программа**

| Адрес | Машинный код | Метка | Мнемокод | Комментарий |
|---|---|---|---|---|
| cs:0100 | B80800 | | mov | делимое |
| cs:0103 | B90300 | | ax,0008 | делитель |
| cs:0106 | BA0100 | | mov | частное |
| cs:0109 | BB0000 | | cx,0003 | остаток |
| cs:010C | D1E0 | m0 | mov | сдвиг делимого |
| cs:010E | D1D3 | | dx,0001 | влево |
| cs:0110 | 2BD9 | | mov | с переносом |
| cs:0112 | 7902 | | bx,0000 | старшего бита в |
| cs:0114 | 03D9 | | shl ax,1 | остаток |
| cs:0116 | F5 | m1 | rcl bx,1 | вычитаем из |
| cs:0117 | D1D2 | | sub bx,cx | остатка делитель |
| cs:0119 | 73F1 | | jns 0116 | если результат |
| cs:011B | CD20 | | add bx,cx | отрицательный, то |
| | | | cmc | восстанавливаем |
| | | | rcl dx,1 | остаток |
| | | | jnb 010C | инвертируем флаг |
| | | | int 20 | C |
| | | | | получаем |
| | | | | очередной разряд |
| | | | | частного |
| | | | | если деление не |
| | | | | закончено, то идём |
| | | | | на m0 |

## 3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить материалы, изложенные в пунктах 1,2.
2. Получить задание на выполнение программирования.
3. Выполнить работу по программированию в соответствии с полученным заданием и практической отработкой программы.
4. Оформить отчет.

## 4. СОДЕРЖАНИЕ ОТЧЕТА

1. Задание на выполнение лабораторной работы.
2. Программная модель (использование РОНов и памяти при решении задачи, алгоритм и программа).
3. Исходные данные, используемые при решении программы и полученные результаты.

## ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ

1. На чем основан алгоритм программы деления чисел?
2. Чем отличается программы 1 и 2?
3. Каким образом организуется образование цифры частного в программах 1 и 2?
4. Объясните организацию неявного счетчика в программе 2?

# Лабораторная работа 5

## ПРОГРАММНАЯ РЕАЛИЗАЦИЯ УМНОЖЕНИЯ ЦЕЛЫХ ДВОИЧНЫХ ЧИСЕЛ

Цель работы: освоение способов программной реализации умножения целых двоичных чисел.

## 1. ОБЩИЕ СВЕДЕНИЯ

Существует несколько алгоритмов умножения чисел. При неявном алгоритме умножение можно заменить многократным сложением, например 14*3=14+14+14. Существенный недостаток этого способа - значительная длительность процесса вычисления. При втором алгоритме умножение осуществляется в столбец. Этот алгоритм применим для умножения двоичных чисел. Пусть требуется умножить число 0110(6) на число 0011(3). Умножение в столбец производится аналогично умножению десятичных чисел:

```
   0110 = 6
   0011 = 3
   0110
  +0110
  0000
 0000
00010010 = 18
```

## 2. ПРОГРАММИРОВАНИЕ УМНОЖЕНИЯ ЦЕЛЫХ ДВОИЧНЫХ ЧИСЕЛ

При вычислении результата по второму алгоритму необходимо осуществить многократное суммирование со сдвигом влево множимого при одновременной проверке содержимого разрядов множителя, начиная со стороны

младшего разряда. При этом если в очередном разряде множителя записана 1, то множимое добавляется к сумме и сдвигается влево на один разряд, а если в разряде записан 0, то произойдет только сдвиг множимого. Сдвиг множимого влево можно заменить сдвигом суммы вправо.

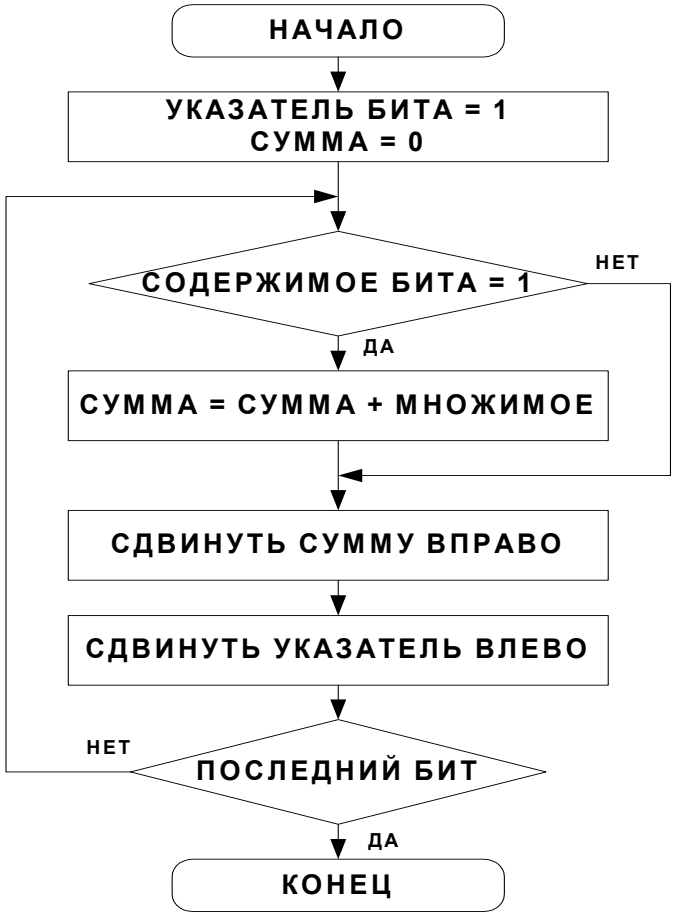


**Таблица 8 - Программа**

| Адрес | Машинный код | Метка | Мнемокод | Комментарий |
|---|---|---|---|---|
| cs:0100 | BB0500 | | mov | загрузка |
| cs:0103 | B90100 | | bx,0005 | множителя |
| cs:0106 | BA0000 | | mov | загрузка |
| cs:0109 | B80000 | | cx,0001 | указателя |
| cs:010C | 50 | | mov | разряда |
| cs:010D | 8BC3 | m0 | dx,0000 | обнуление DX |
| cs:010F | 23C1 | | mov | обнуление AX |
| cs:0111 | 7403 | | ax,0000 | |
| cs:0113 | 83C203 | | push ax | |
| cs:0116 | D1FA | m1 | mov | проверка |
| cs:0118 | 58 | | ax,bx | очередного |
| cs:0119 | D1D8 | | and ax,cx | разряда |
| cs:011B | 50 | | je 0116 | |
| cs:011C | D1E1 | | add | |
| cs:011E | 73ED | | dx,0003 | сдвиг старшего |
| cs:0120 | CD20 | | sar dx,1 | слова результата |
| | | | pop ax | |
| | | | rcr ax,1 | сдвиг младшего |
| | | | push ax | слова результата |
| | | | shl cx,1 | |
| | | | jnb 010D | сдвиг указателя |
| | | | int 20 | разряда |
| | | | | если произошел |
| | | | | перенос, то |
| | | | | конец цикла |

## 3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить материалы, изложенные в пунктах 1,2.
2. Получить задание на выполнение программирования.
3. Выполнить работу по программированию в соответствии с полученным заданием и практической отработкой программы.
4. Оформить отчет.

## 4.СОДЕРЖАНИЕ ОТЧЕТА

1. Задание на выполнение лабораторной работы.
2. Программная модель (использование РОНов и памяти при решении задачи, алгоритм и программа).
3. Исходные данные, используемые при решении программы и полученные результаты.

## ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ

1. Приведите примеры существующих алгоритмов умножения.
2. На чем основан алгоритм работы программы умножения?
3. Что такое «сдвиг влево», «сдвиг вправо»?

# ПРИЛОЖЕНИЯ

## 1. Варианты заданий для лабораторной работы №1

| № вар. | Исходный массив |
|---|---|
| 1 | 12,13,14,15,16,17,18,19,20,21 |
| 2 | 22,23,24,25,26,27,28,29,30,31 |
| 3 | 32,33,34,35,36,37,38,39,40,41 |
| 4 | 42,43,44,45,46,47,48,49,50,51 |
| 5 | 52,53,54,55,56,57,58,59,60,61 |
| 6 | 62,63,64,65,66,67,68,69,70,71 |
| 7 | 72,73,74,75,76,77,78,79,80,81 |
| 8 | 82,83,84,85,86,87,88,89,90,91 |
| 9 | 92,93,94,95,96,97,98,99,100,101 |
| 10 | 102,103,104,105,106,107,108,109,110,111 |
| 11 | 112,113,114,115,116,117,118,119,120,121 |
| 12 | 122,123,124,125,126,127,128,129,130,131 |
| 13 | 132,133,134,135,136,137,138,139,140,141 |
| 14 | 142,143,144,145,146,147,148,149,150,151 |
| 15 | 152,153,154,155,156,157,158,159,160,161 |
| 16 | 162,163,164,165,166,167,168,169,170,171 |
| 17 | 172,173,174,175,176,177,178,179,180,181 |
| 18 | 182,183,184,185,186,187,188,189,190,191 |
| 19 | 192,193,194,195,196,197,198,199,200,201 |
| 20 | 202,203,204,205,206,207,208,209,210,211 |
| 21 | 212,213,214,215,216,217,218,219,220,221 |
| 22 | 222,223,224,225,226,227,228,229,230,231 |
| 23 | 232,233,234,235,236,237,238,239,240,241 |
| 24 | 242,243,244,245,246,247,248,249,250,251 |
| 25 | 252,253,254,255,0,1,2,3,4,5 |

## 2. Варианты заданий для лабораторной работы №2а

| № вар. | Первое число | Второе число | Сумма |
|---|---|---|---|
| 1 | 5 | 25 | 61 |
| 2 | 10 | 1 | 43 |
| 3 | 34 | 46 | 27 |
| 4 | 33 | 17 | 28 |
| 5 | 23 | 4 | 8 |
| 6 | 59 | 49 | 41 |
| 7 | 41 | 3 | 58 |
| 8 | 56 | 39 | 45 |
| 9 | 19 | 52 | 23 |
| 10 | 62 | 2 | 61 |
| 11 | 52 | 28 | 46 |
| 12 | 7 | 19 | 21 |
| 13 | 29 | 28 | 36 |
| 14 | 45 | 0 | 12 |
| 15 | 51 | 6 | 12 |
| 16 | 26 | 9 | 35 |
| 17 | 40 | 42 | 149 |
| 18 | 30 | 25 | 54 |
| 19 | 16 | 9 | 40 |
| 20 | 40 | 23 | 50 |
| 21 | 29 | 36 | 16 |
| 22 | 59 | 21 | 33 |
| 23 | 42 | 26 | 51 |
| 24 | 2 | 23 | 5 |
| 25 | 14 | 49 | 18 |

## 3. Варианты заданий для лабораторной работы №2б

| № вар. | Массив чисел | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 6 | 39 | 51 | 23 | 10 | 43 | 38 |
| 2 | 32 | 53 | 54 | 47 | 46 | 32 | 57 |
| 3 | 7 | 30 | 45 | 37 | 22 | 27 | 10 |
| 4 | 48 | 18 | 21 | 34 | 16 | 18 | 57 |
| 5 | 57 | 41 | 46 | 15 | 41 | 13 | 34 |
| 6 | 38 | 29 | 49 | 30 | 32 | 54 | 19 |
| 7 | 5 | 58 | 54 | 54 | 30 | 58 | 20 |
| 8 | 33 | 59 | 52 | 54 | 38 | 26 | 26 |
| 9 | 61 | 40 | 42 | 29 | 19 | 14 | 40 |
| 10 | 57 | 58 | 39 | 1 | 52 | 58 | 63 |
| 11 | 58 | 52 | 5 | 2 | 51 | 44 | 17 |
| 12 | 43 | 40 | 14 | 63 | 11 | 36 | 39 |
| 13 | 13 | 26 | 23 | 14 | 54 | 36 | 16 |
| 14 | 1 | 21 | 52 | 50 | 18 | 49 | 13 |
| 15 | 16 | 40 | 36 | 15 | 59 | 6 | 29 |
| 16 | 48 | 28 | 58 | 44 | 6 | 26 | 32 |
| 17 | 9 | 22 | 58 | 4 | 36 | 27 | 32 |
| 18 | 31 | 49 | 49 | 18 | 44 | 47 | 28 |
| 19 | 39 | 59 | 4 | 17 | 28 | 45 | 7 |
| 20 | 39 | 48 | 9 | 57 | 44 | 59 | 36 |
| 21 | 39 | 40 | 47 | 58 | 18 | 2 | 8 |
| 22 | 24 | 26 | 40 | 58 | 18 | 2 | 8 |
| 23 | 25 | 44 | 38 | 3 | 31 | 44 | 50 |
| 24 | 51 | 44 | 47 | 48 | 2 | 23 | 38 |

## 4. Варианты заданий для лабораторной работы №3

| № вар. | Массив | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 6 | 3 | 5 | 4 | 0 | 6 | 7 | 7 | 2 |
| 2 | 1 | 0 | 6 | 8 | 8 | 3 | 3 | 3 | 8 | 7 |
| 3 | 2 | 8 | 2 | 7 | 5 | 8 | 2 | 8 | 0 | 3 |
| 4 | 8 | 5 | 0 | 3 | 9 | 5 | 5 | 2 | 2 | 0 |
| 5 | 6 | 1 | 8 | 1 | 7 | 0 | 8 | 5 | 4 | 3 |
| 6 | 6 | 8 | 2 | 1 | 3 | 6 | 6 | 7 | 1 | 5 |
| 7 | 0 | 3 | 2 | 5 | 8 | 4 | 1 | 7 | 5 | 6 |
| 8 | 1 | 3 | 3 | 5 | 4 | 5 | 2 | 4 | 0 | 3 |
| 9 | 5 | 8 | 8 | 0 | 6 | 2 | 7 | 0 | 4 | 9 |
| 10 | 8 | 7 | 3 | 1 | 1 | 6 | 9 | 1 | 0 | 9 |
| 11 | 8 | 5 | 1 | 2 | 2 | 2 | 6 | 0 | 7 | 9 |
| 12 | 8 | 6 | 2 | 3 | 2 | 4 | 8 | 6 | 4 | 7 |
| 13 | 8 | 0 | 0 | 8 | 4 | 1 | 1 | 3 | 1 | 5 |
| 14 | 0 | 9 | 4 | 1 | 6 | 1 | 8 | 4 | 4 | 6 |
| 15 | 8 | 9 | 5 | 6 | 7 | 5 | 3 | 4 | 8 | 2 |
| 16 | 3 | 2 | 9 | 6 | 7 | 9 | 4 | 6 | 0 | 4 |
| 17 | 9 | 5 | 4 | 4 | 4 | 8 | 6 | 4 | 2 | 0 |
| 18 | 8 | 0 | 5 | 7 | 8 | 4 | 2 | 0 | 6 | 6 |
| 19 | 1 | 0 | 8 | 5 | 2 | 5 | 5 | 4 | 8 | 6 |
| 20 | 3 | 1 | 6 | 4 | 0 | 8 | 1 | 9 | 8 | 3 |
| 21 | 4 | 7 | 7 | 3 | 8 | 1 | 4 | 9 | 2 | 4 |
| 22 | 3 | 6 | 7 | 1 | 7 | 8 | 6 | 5 | 0 | 1 |
| 23 | 1 | 0 | 2 | 8 | 9 | 2 | 5 | 1 | 5 | 9 |
| 24 | 4 | 7 | 0 | 0 | 3 | 5 | 3 | 0 | 0 | 4 |
| 25 | 6 | 2 | 5 | 0 | 3 | 6 | 5 | 4 | 0 | 4 |

## 5. Варианты заданий для лабораторной работы №4

| № вар. | Делимое | Делитель |
|---|---|---|
| 1 | 78 | 10 |
| 2 | 125 | 15 |
| 3 | 28 | 9 |
| 4 | 118 | 15 |
| 5 | 67 | 4 |
| 6 | 193 | 6 |
| 7 | 218 | 7 |
| 8 | 220 | 14 |
| 9 | 223 | 5 |
| 10 | 237 | 14 |
| 11 | 142 | 12 |
| 12 | 50 | 11 |
| 13 | 56 | 8 |
| 14 | 40 | 6 |
| 15 | 169 | 7 |
| 16 | 69 | 6 |
| 17 | 227 | 9 |
| 18 | 5 | 2 |
| 19 | 118 | 3 |
| 20 | 32 | 15 |
| 21 | 36 | 15 |
| 22 | 205 | 11 |
| 23 | 233 | 4 |
| 24 | 22 | 11 |
| 25 | 146 | 15 |

## 6. Варианты заданий для лабораторной работы №5

| № вар. | Множимое | Множитель |
|---|---|---|
| 1 | 791 | 470 |
| 2 | 24330 | 28184 |
| 3 | 49113 | 15186 |
| 4 | 50384 | 38422 |
| 5 | 58132 | 15232 |
| 6 | 56731 | 40598 |
| 7 | 48685 | 9535 |
| 8 | 10868 | 31008 |
| 9 | 61462 | 31997 |
| 10 | 52607 | 22324 |
| 11 | 62177 | 2228 |
| 12 | 36173 | 15710 |
| 13 | 6761 | 62327 |
| 14 | 46810 | 36387 |
| 15 | 27843 | 2505 |
| 16 | 52527 | 46811 |
| 17 | 43760 | 33134 |
| 18 | 22792 | 56098 |
| 19 | 52725 | 44901 |
| 20 | 8752 | 33216 |
| 21 | 20918 | 26132 |
| 22 | 15607 | 21084 |
| 23 | 18806 | 45598 |
| 24 | 3949 | 30510 |
| 25 | 22318 | 13627 |

# 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 OLIKO», 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 II

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

Сдано в набор 02.05.2012
Подписано в печать 05.06.2012.
Бумага писчая. Гарнитура Таймс.
Усл.печ.л. 2,8. Тираж 100.
Заказ №34.

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

46