

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
Томский государственный университет систем управления и
радиоэлектроники

Кафедра автоматизированных систем управления

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Методические указания по самостоятельной и
индивидуальной работе студентов всех форм обучения
для направлений бакалавриата
090301 – Информатика и вычислительная техника

Томск-2020

Алфёров С.М., Лукьянов А.К.

Вычислительная техника: методические указания по самостоятельной и индивидуальной работе студентов всех форм обучения направлений бакалавриата 090301 – Информатика и вычислительная техника/ – Томск: ТУСУР, 2020. – 13 с.

Методические указания разработаны в соответствии с решением кафедры автоматизированных систем управления

Составители:

доцент каф. АСУ, С.М. Алфёров, доцент каф. АСУ, А.К. Лукьянов

Методические указания утверждены на заседании кафедры автоматизированных систем управления, протокол № 10, 31 октября 2020 года.

© ТУСУР, каф. АСУ, 2020

© Алфёров С.М., Лукьянов А.К., 2020

СОДЕРЖАНИЕ

1. Лабораторные работы.....	4
1.1 Двоичная арифметика.....	4
1.2 Ассемблерные вставки на языках высокого уровня. Команды пересылки данных.....	6
1.3 Арифметические команды Ассемблера.....	9
1.4 Команды условных и безусловных переходов.....	11
1.5. HTML и JavaScript.....	13
1.6. Создание Web-приложений.....	15
1.7. Массивы и векторные операции.....	17
2. Список рекомендуемой литературы.....	22
2.1. Основная литература.....	22
2.2. Дополнительная литература.....	22
2.3. Перечень пособий, методических указаний и материалов, используемых в учебном процессе.....	22

1. Лабораторные работы

1.1 Двоичная арифметика

1. ЦЕЛЬ РАБОТЫ

Получить представление о способах хранения числовых данных в памяти ЭВМ.

2. ЗАДАНИЕ

Написать программу по выводу битовых данных (битов), числовых значений, хранящихся в переменных. Задать значения и вывести на экран биты: целых знаковых и беззнаковых чисел длиной 1, 2 и 4 байта; вещественных чисел с плавающей запятой длиной 4 и 8 байт. Объяснить результат. Типы данных для целых чисел в различных средах программирования приведены в таблице 1. Вещественные 4-х байтные числа в языках программирования Pascal/Object Pascal объявляются типом single, в языках программирования C/C++ объявляются типом float. Вещественные 8-ми байтные числа во всех языках программирования объявляются типом double.

Таблица 1 – Типы переменных.

Тип Среда	4-х байтные		2-х байтные		однобайтные	
	беззнаков ые	знаковые	беззнаков ые	знаковые	беззнаков ые	знаковые
Borland Pascal	-	longint	word	integer	byte	shortint
Borland C++	unsigned long int	long int	unsigned int	int	unsigned char	char
Delphi, Embarcade ro (Object Pascal)	cardinal	integer	word	smallint	byte	shortint
Builder, Embarcade ro (C++), Visual Studio C++	unsigned int	int	unsigned short int	short int	Unsigned char	char

Пояснения к выполнению работы.

Для выполнения задания рекомендуется воспользоваться следующими операциями C/C++ (аналогичные операции есть и в языках Pascal/Object Pascal, только называются по другому):

Побитовый сдвиг вправо: >>

Побитовый сдвиг влево: <<

Взятие адреса: & (унарная)
 Побитовое «И»: & (бинарная)
 Взятие остатка от целочисленного деления: %
 Преобразование типа: ()
 Например:

```

void main() {
int x; // наша переменная
int *p; // указатель на ячейку типа int
char *pc; // указатель на ячейку типа char
p = &x; // указателю p присвоили адрес переменной x,
теперь p указывает на x
pc = (char*)p; // указателю pc присвоили адрес переменной
x, теперь,
// через указатель pc можно обращаться к 4-х байтной
ячейке x,
// как к массиву из четырех ячеек типа char
// pc[0] - 1-й (самый младший) байт переменной x
// . . .
// pc[3] - 4-й (самый старший) байт переменной x
unsigned char *pb; // указатель на ячейку типа unsigned
char
unsigned char b;
pb=(unsigned char *)&x; // указателю pb присвоили адрес
переменной x
b=pb[0]&2; // переменной b присвоили значение самого
младшего бита
// (под номером 0) из переменной x
b=(pb[0]>>1)&2; // переменной b присвоили значение бита
// под номером 1 из переменной x
// и т.д. до бита под номером 7
b=pb[1]&2; // бит под номером 8
b=(pb[1]>>1)&2; // бит под номером 9
//. . .
}

```

3. СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать: цель, задание, текст программы, результаты работы, вывод. В результатах работы должны быть скриншоты результатов работы программы при различных входных данных и объяснение полученных результатов.

1.2 Ассемблерные вставки на языках высокого уровня. Команды пересылки данных

1. ЦЕЛЬ РАБОТЫ

Ознакомление с языком программирования Ассемблер и его командами пересылки данных `mov` и `xchg`, средствами формирования длины и смещения операндов `byte ptr`, `word ptr`, `dword ptr`, [число], +число.

2. ЗАДАНИЕ

Согласно своему варианту переставить байты в двух или трех переменных. Задание следует решить за минимальное количество команд.

Для наглядности схемы перестановки, значения входных и выходных данных в вариантах задания приведены в 16-тиричной системе счисления, так как один байт соответствует как раз двум 16-тиричным разрядам. Длину (тип) переменной можно определить по значению в варианте.

Выполнение работы рекомендуется в одной из сред программирования: Borland Pascal, Borland C++, Delphi, Builder, Embarcadero RAD Studio, Lazarus, Visual Studio C++. Для удобства вывода рекомендуется использовать беззнаковые типы данных соответствующей длины. Соответствие типов в разных средах программирования приведено в таблице 1.

При необходимости, написать процедуру для перевода чисел в шестнадцатеричную систему счисления, можно использовать для этого метод деления на 16.

3. СОДЕРЖАНИЕ ОТЧЕТА

- 1) Общее задание.
- 2) Задание своего варианта.
- 3) Текст программы своего варианта с описанием каждой строки, каждого оператора и операнда.
- 4) Экранная копия, демонстрирующая работу программы.

4. ВАРИАНТЫ

1) Даны две переменные А и В. Переставить байты в переменных по следующей схеме:

Начальная нумерация байт	После перестановки
A = 11 22 33 44	A = 11 66 33 88
B = 55 66 77 88	B = 22 55 77 44

2) Даны две переменные A и B. Переставить байты в переменных по следующей схеме:

Начальная нумерация байт	После перестановки
A = 11 22 33 44	A = 22 11 44 88
B = 55 66 77 88	B = 55 66 77 33

3) Даны две переменные A и B. Переставить байты в переменных по следующей схеме:

Начальная нумерация байт	После перестановки
A = 11 22 33 44	A = 55 11 66 33
B = 55 66	B = 22 44

4) Даны две переменные A и B. Переставить байты в переменных по следующей схеме:

Начальная нумерация байт	После перестановки
A = 11 22 33 44	A = 44 22 55 33
B = 55 66	B = 66 11

5) Даны две переменные A и B (long int) (int). Переставить байты в переменных по следующей схеме:

Начальная нумерация байт	После перестановки
A = 11 22 33 44	A = 33 44 66 55
B = 55 66	B = 11 22

6) Даны три переменные A B и C (int) (int) (long int). Переставить байты в переменных по следующей схеме:

Начальная нумерация байт	После перестановки
A = 11 22	A = 33 77
B = 33 44	B = 44 88
C = 55 66 77 88	C = 11 22 66 55

7) Даны три переменные A B и C (int) (int) (long int). Переставить байты в переменных по следующей схеме:

Начальная нумерация байт	После перестановки
A = 11 22	A = 88 55
B = 33 44	B = 44 88
C = 55 66 77 88	C = 11 22 66 55

8) Даны две переменные A и B (long int) (long int). Переставить байты в переменных по следующей схеме:

Начальная нумерация байт	После перестановки
A = 11 22 33 44	A = 55 66 77 88
B = 55 66 77 88	B = 11 22 33 44

9) Даны две переменные A и B (long int) (long int). Переставить байты в переменных по следующей схеме:

Начальная нумерация байт	После перестановки
A = 11 22 33 44	A = 11 77 88 44
B = 55 66 77 88	B = 33 66 55 44

10) Даны две переменные A и B (long int) (int). Переставить байты в переменных по следующей схеме:

Начальная нумерация байт	После перестановки
A = 11 22 33 44	A = 55 66 33 44
B = 55 66	B = 22 44

11) Даны две переменные A и B (long int) (int). Переставить байты в переменных по следующей схеме:

Начальная нумерация байт	После перестановки
A = 11 22 33 44	A = 55 22 66 44
B = 55 66	B = 33 11

12) Даны две переменные A и B (long int) (int). Переставить байты в переменных по следующей схеме:

Начальная нумерация байт	После перестановки
A = 11 22 33 44	A = 22 11 55 44
B = 55 66	B = 33 66

13) Даны три переменные A B и C (int) (int) (long int). Переставить байты в переменных по следующей схеме:

Начальная нумерация байт	После перестановки
A = 11 22	A = 33 44
B = 33 44	B = 66 77
C = 55 66 77 88	C = 88 22 11 55

14) Даны три переменные A B и C (int) (int) (long int). Переставить байты в переменных по следующей схеме:

Начальная нумерация байт	После перестановки
A = 11 22	A = 44 11
B = 33 44	B = 66 77
C = 55 66 77 88	C = 55 22 33 88

1.3 Арифметические команды Ассемблера

1. ЦЕЛЬ РАБОТЫ

Ознакомление со встроенным в Borland Pascal языком Ассемблера. Получение навыков работы с командами арифметики (add, adc, sub, sbb, mul, imul) и преобразования данных (cbw, cwd).

2. ЗАДАНИЕ

Вычислить целочисленное выражение, указанное в варианте задания. При этом, и операнды и результаты вычислений следует выводить как в десятичном, так и в шестнадцатеричном виде. Для перевода чисел в шестнадцатеричную систему необходимо написать процедуру на языке высокого уровня. Используйте для перевода метод деления на 16.

3. ВАРИАНТЫ

- 1) A (integer), B (byte), C (longint). Вычислить: $D = C - A/B$; $E = B * B - A + C$.
- 2) A (longint), B (byte), C (integer). Вычислить: $D = (A/B) * C$; $E = (C^2/B) - A$.
- 3) A (byte), B (byte), C (longint). Вычислить: $D = (C - A)/B$; $E = (A * b) + B$.
- 4) A (integer), B (integer), C (longint). Вычислить: $D = (A * B) - C$; $E = (C/B) + A$.
- 5) A (byte), B (integer), C (integer). Вычислить: $D = A^2 - (B/C)$; $E = (C * B) + A$.
- 6) A (byte), B (integer), C (integer). Вычислить: $D = (B * C) + A$; $E = (C * A)/B$.
- 7) A (int), B (longint), C (integer). Вычислить: $D = (C/A) * C + B$; $E = (B - C)/A * C$.
- 8) A (byte), B (byte), C (integer). Вычислить: $D = (A^2) * B/C$; $E = (A - B)/(A - C)$.
- 9) A (byte), B (byte), C (longint). Вычислить: $D = (A * B) * B$; $E = (A - B) * (C - A)/B$.
- 10) A (integer), B (integer), C (byte). Вычислить: $D = (A * B)/C$; $E = (A - B) * (B - C)$.
- 11) A (longint), B (integer), C (byte). Вычислить: $D = (B * C) + A$; $E = (B - C)/B - A$.
- 12) A (byte), B (integer), C (longint). Вычислить: $D = C + A^3$; $E = (C - B) + (B * A)$.
- 13) A (integer), B (byte), C (integer). Вычислить: $D = (B - C) * (B + C)$; $E = (B * A)/C$.

14) A (byte), B (integer), C (integer). Вычислить: $D = A^3 + B$; $E = C^2 * (B - C) / B$.

1.4 Команды условных и безусловных переходов

1. ЦЕЛЬ РАБОТЫ

Получить навыки использования операторов сравнения (cmp), условных (jxx), безусловных (jmp) переходов и циклов (loop).

2. ЗАДАНИЕ

Задание из двух частей. В первом задании необходимо подсчитать количество чисел, соответствующих определенному условию на некотором числовом промежутке. Во втором задании требуется модифицировать свою программу так, чтобы найти и вывести на экран двухсотое число или пару чисел соответствующих заданному условию. Если таких чисел меньше двухсот, то вывести об этом сообщение на экран. Для выполнения задания выделяется 8 часов аудиторного времени.

3. ВАРИАНТЫ

1) На промежутке от 1 до 90 000. Подсчитать количество таких чисел X , что $(X+X-1)$ - простое число. Ответ вывести на экран.

2) На промежутке от 1 до 127. Подсчитать количество таких чисел X , что $((X*X)-1)$ - простое число. Ответ вывести на экран.

3) На промежутке от 1 до 100 000. Подсчитать количество таких чисел X , что $(X+X-2)$ - делится на 3 без остатка. Ответ вывести на экран.

4) На промежутке от 1 до 255. Подсчитать количество таких пар чисел X и Y , что $(X*Y)$ при делении на 303 дает в остатке 1. Ответ вывести на экран.

5) На промежутке от -127 до 127. Подсчитать количество таких пар чисел X и Y , что $(X+Y) = 80$. Ответ вывести на экран.

6) На промежутке от -128 до 127. Подсчитать количество таких пар чисел X и Y , что $(/X/+Y) \leq 70$. Ответ вывести на экран.

7) На промежутке от -128 до 127. Подсчитать количество таких пар чисел X и Y , что $50 < (X-Y) \leq 80$. Ответ вывести на экран.

8) На промежутке от 1 до 100 000. Подсчитать количество таких чисел X , что $(X+X-1) \text{div} 500$ - простое число. Ответ вывести на экран.

9) На промежутке от 1 до 32768. Подсчитать количество таких чисел X , что $((X \text{ div } 21) - 1)$ - четное число. Ответ вывести на экран.

10) На промежутке от -128 до 127. Подсчитать количество таких пар чисел X и Y , что X/Y дает в остатке 0. Ответ вывести на экран.

11) На промежутке от -128 до 127. Подсчитать количество таких пар чисел X и Y , что $(X * Y - (X \text{ div } 2)) = 0$. Ответ вывести на экран.

12) На промежутке от -128 до 127. Подсчитать количество таких пар чисел X и Y , что $(X + Y * X) \neq 5000$. Ответ вывести на экран.

13) На промежутке от -128 до 127. Подсчитать количество таких пар чисел X и Y , что $(|X| + Y) \text{ div } 50 \Rightarrow 100$. Ответ вывести на экран.

14) На промежутке от -128 до 127. Подсчитать количество таких пар чисел X и Y , что $(|X| - |Y|) \bmod 4 = 0$. Ответ вывести на экран.

1.5. HTML и JavaScript.

1. ЦЕЛЬ

Получить навык создания веб-страниц.

2. ЗАДАНИЕ

Создать веб-страницу с двумя формами: формой ввода параметров и формой вывода параметров. В первой форме: поля ввода согласно своему варианту (Таблица 2) и кнопка для вывода введенных значений во вторую форму. Выведенные во вторую форму значения должны соответствовать правила предметной области (ПО), если введенные значения не соответствуют условиям, то автоматически подкорректировать до ближайших разрешенных значений или проигнорировать изменение параметра.

3. СОДЕРЖАНИЕ ОТЧЕТА

В результатах работы отчета должны быть: тексты программ; пример работы каждой программы с входными и выходными данными; таблица, содержащая время обработки входного массива каждой из программ.

4. ВАРИАНТЫ

Таблица 2 - Варианты заданий для веб-страницы.

№	Форма	Поля ввода	Правила ПО
1	Монстр	Здоровье, сила	Суммарное значение здоровья и силы должно лежать в диапазоне от 2 до 150, каждое свойство должно лежать в диапазоне от 1 до 100. Если задаваемое значение не будет соответствовать условию, то автоматически присвоить ближайшее разрешенное.
2	Почтовый адрес	улица, дом, квартира	Первые символы до пробела свойства «улица» указывают на тип. «ул» - улица, «пр» - проспект, «пер» - переулок. На переулке номера домов лежат в диапазоне от 1 до 30, на улице от 1 до 100, на проспекте от 1 до 1000. Если дом частный, то квартира не указывается.
3	Результат Зачетной недели	Студент, семестр предметы,	Семестр может быть от 1 до 9. На 3, 7 семестрах студенты сдают по 3 зачета, на 5 и 6 по 4 зачета, на

		зачеты	остальных по 5 зачетов. Написать функцию, определяющую можно ли студенту поставить штамп «зачтено», который ставится только при всех сданных зачетах.
4	Сессия ВУЗа (простое)	Семестр, Количество зачетов, количество экзаменов	Суммарное количество зачетов и экзаменов должно быть от 7 до 9.
5	Человек (сложное)	Голова, туловище, руки, ноги	Голова задается окружностью с заданными координатами центра и радиусом. Туловище задается прямоугольником: координатами верхнего левого угла, высотой и шириной. Руки и ноги задаются длинами и углами положения. Все члены не должны быть отделены друг от друга. Руки растут из верхних углов прямоугольника, ноги из нижних. Написать функцию определения роста человека.
6	Система прямоугольников	Прямоугольник1, Прямоугольник2	Прямоугольники задаются координатами. Площади прямоугольников не должны пересекаться и должны лежать в координатах от (0, 0) до (1000, 1000). Размеры прямоугольников не должны превышать 500.
7	Система фигур	Прямоугольник, окружность.	Прямоугольник задается координатами. Окружность задается координатами центра и радиусом. Площади фигур должны пересекаться. Написать функцию определения общей площади занимаемой фигурами.
8	Дата рождения пациента больницы	День, месяц, год	Год должен лежать в диапазоне от 1900 до 2014. Месяц может принимать значения от «январь» до «декабрь», день в зависимости от месяца и года (учитывая високосные года). Написать функцию для определения возраста пациента.
9	Табурет	Высота седушки, количество ножек, материал	Высота седушки: от 20 до 60 см. Количество ножек: 1, 3 или 4. материал: металл, дерево, пластик.

1.6. Создание Web-приложений

1. ЦЕЛЬ

Получить навыки программирования web-приложений.

2. ЗАДАНИЕ

При обращении к серверу, он отправляет клиенту данные из своего хранилища и отображает их в окне браузера. Клиент обеспечивает ввод данных пользователем и отправку их на сервер для сохранения в хранилище.

Задание рекомендуется сделать с использованием языка программирования PHP или Python. На стороне клиента используются языки: HTML, Javascript.

СОДЕРЖАНИЕ ОТЧЕТА

В результатах работы продемонстрировать подключение к серверу с различных узлов, возможность ввода данных и вывода актуальных данных, хранящихся на сервере.

3. ВАРИАНТЫ

1. Форум

Сервер в хранилище должен сохранять список сообщений (от кого сообщение, текст сообщения, дата-время приема сообщения). Клиент отправляет на сервер текст сообщения и подписывается (от кого сообщение).

2. Форум

Сервер в хранилище должен сохранять список сообщений (от кого сообщение, кому сообщение (может быть пустым), текст сообщения). Клиент отправляет на сервер текст сообщения, подписывается (от кого сообщение), может указать имя пользователя, кому обращено сообщение.

3. Интернет-магазин

Сервер хранит список товаров и их количество на складе. Клиент отправляет на сервер товар, который пользователь хочет приобрести, и в каком количестве. Сервер уменьшает количество купленного товара на указанную величину. Сервер не должен допускать уменьшения количества товара меньше нуля.

4. Журнал преподавателя

Сервер хранит оценки учеников по различным предметам. Клиент отправляет оценку, полученную указанным учеником

по указанному предмету. Если оценка по указанному предмету у данного ученика уже есть, то заменить на новую. Отображать хранилище в виде таблицы: строки - ученики, столбцы — предметы.

1.7. Массивы и векторные операции

1. ЦЕЛЬ

Научиться обрабатывать массивы данных на языке Ассемблер. Познакомиться с векторными операциями процессора.

2. ЗАДАНИЕ

Задание состоит из трёх частей. В первой части нужно выполнить задание на языке с++. Во второй части требуется написать программу обработки массива согласно заданию, используя скалярные команды обработки данных. Во второй части требуется написать такую же программу, но используя векторные операции. Для каждой программы засечь время выполнения, провести не менее 100 замеров. Вычислить среднее время выполнения для каждой программы и сделать вывод об эффективности векторных операций. На выполнение всего задания выделяется 8 часов аудиторного времени.

Пояснения к выполнению работы.

Формат файла BMP (сокращенно от BitMaP) - это формат растровой графики близкий Windows, поскольку он лучше всего соответствует внутреннему формату Windows, в котором эта система хранит свои растровые массивы.

В файлах BMP информация о цвете каждого пикселя кодируется 1, 4, 8, 16, 24 или 32 битами (бит/пиксель). Числом бит/пиксель, называемым также глубиной представления цвета, определяется максимальное число цветов в изображении. Изображение при глубине 1 бит/пиксель может иметь всего два цвета, а при глубине 24 бит/пиксель - более 16 млн. различных цветов.

В таблице 3 показана структура BMP-файла без палитры. Файл разбит на 3 основных раздела: заголовок файла растровой графики, информационный заголовок растрового массива и данные растрового массива. Заголовок файла растровой графики содержит информацию о файле, в том числе адрес, с которого начинается область данных растрового массива. В информационном заголовке растрового массива содержатся сведения об изображении, хранящемся в файле, например, его высоте и ширине в пикселях.

Структура пикселя для 32-битной картинки:

X R G B или A R G B,

здесь X - зарезервированное место, A - прозрачность, R G B - компоненты цветности.

Таблица 3 — Структура BMP-файла без палитры.

Заголовок файла растровой графики (14 байт)	Сигнатура файла BMP (2 байт) – Символы “BM” Размер файла (4 байт) Не используется (2 байт) Не используется (2 байт) Местонахождение данных растрового массива (4 байт)
Информационный заголовок растрового массива (40 байт)	Длина этого заголовка (4 байт) Ширина изображения (4 байт) Высота изображения (4 байт) Число цветовых плоскостей (2 байт) Бит/пиксел (2 байт) Метод сжатия (4 байт) Длина растрового массива (4 байт) Горизонтальное разрешение (4 байт) Вертикальное разрешение (4 байт) Число цветов изображения (4 байт) Число основных цветов (4 байт)
Данные растрового массива	RGBRGBRGB... R=1 байт, G=1 байт, B=1 байт

Пример программы копирования 32-битной картинки .bmp

```
#include "stdafx.h"

#include <stdlib.h>
#include <conio.h>

int _tmain(int argc, _TCHAR* argv[])
{
    FILE *in, *out;

    unsigned __int8 *buffer;
    unsigned __int32 wight, height;

    fopen_s(&in, "D:\\twilight.bmp", "rb");
    if (in == NULL)
    {
        printf("Error: Can't open twilight.bmp for reading\n");
        return 0;
    }

    buffer = (unsigned __int8*)malloc(54);
    fread(buffer, 1, 54, in);

    for (int i = 0; i <= 3; i++)
    {
        wight = buffer[21 - i] | wight << 8;
        height = buffer[25 - i] | height << 8;
    }

    printf("wight = %d\n", wight);
    printf("height = %d\n", height);
}
```

```

fopen_s(&out, "D:\\out.bmp", "wb");
if (out == NULL)
{
    printf("Error: Can't open file %s for writing\n", "D:\\out.bmp");
    return 0;
}

fwrite(buffer, 1, 54, out);
free(buffer);
buffer = (unsigned __int8*)malloc(wight*height*4);
fread(buffer, 1, wight*height * 4, in);
fclose(in);
fwrite(buffer, 1, wight*height * 4, out);
fclose(out);
free(buffer);

_getch();

return 0;
}

```

Список функций mmx

Таблица 4 - Операции для векторной обработки данных.

emms	Подготовка FPU контекста для выполнения FPU операций после MMX
movd	Запись младшего двойного слова из регистра MMX в регистр общего назначения или ячейку памяти. При записи в регистр MMX старшее двойное слово обнуляется.
movq	Запись qword из регистра MMX в память и из памяти в регистр MMX
pack[ssdw,sswb,uswb]	Преобразование (с насыщением) размеров единиц данных из первого операнда в младшее двойное слово результата, из второго - в старшее. С.м пример ниже.
padd[b, w, d, s[b,w], us[b,w]]	Сложение элементов данных регистра. возможно насыщение.
pand[n]	Операция AND/ AND и XOR над данными регистра
pcmpreq[b, w, d]	сравнить элементы данных, если равны установить биты элемента в единицы, если нет то в нули.
pcmpgt[b, w, d]	сравнить элементы данных, если соответствующий элемент данных первого операнда больше соответствующего второго, то установить его в единицы, если нет то в нули.
pmaddwd	Перемножить каждое соответствующее слово операндов, сложить результат умножения старших двух слов и записать в старшее двойное слово ре-

	зультата, младших - в младшее.
pmul[h,l]w	Знаковое умножение упакованных слов. Возврат старших/младших слов результата.
por	Операция OR
psll[w, d, q]	Логический сдвиг влево упакованных единиц данных
psra[w, d]	Арифметический сдвиг вправо
psrl[w, d, q]	Логический сдвиг вправо
psub[b, w, d, s[b, w], us[b,w]]	Вычитание упакованных элементов. Возможно насыщение.
punpck[h,l][bw,wd,dq]	Распаковка старших/младших двойных слов операндов. С.м ниже.
pxor	Операция XOR

СОДЕРЖАНИЕ ОТЧЕТА

В результатах работы продемонстрировать эффект от использования векторных операций процессора при обработки раstra изображения.

3. ВАРИАНТЫ

1. Увеличить яркость красной составляющей верхней половины картинки.

2. Увеличить яркость зелёной составляющей картинки у правой стороны картинки

3. Увеличить яркость синей составляющей над главной диагональю матрицы пикселей.

4. Увеличить яркость красной и зелёной составляющих у чётных строк.

5. Увеличить яркость горизонтальными полосами по 64 пикселя.

6. Увеличить яркость вертикальными полосами по 64 пикселя.

7. Уменьшить яркость зелёной составляющей нижней половины картинки.
8. Уменьшить яркость синей составляющей картинки у правой стороны картинки
9. Уменьшить яркость красной составляющей под главной диагональю матрицы пикселей.
10. Уменьшить яркость синей и красной составляющих у нечётных строк.
11. Уменьшить яркость красной и зелёной составляющих у строк, номера которых не являются числами ряда Фибоначчи.
12. Уменьшить яркость у прямоугольника в центре с высотой 64 пикселя и шириной 128 пикселей.

2. Список рекомендуемой литературы

2.1. Основная литература

1. Пятибратов, Александр Петрович. Вычислительные системы, сети и телекоммуникации: учебник для вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко; ред. А. П. Пятибратов. - 3-е изд., перераб. и доп. - М.: Финансы и статистика, 2006. - 558 с. (90 экз. библиотека ТУСУР)
2. Павловская, Татьяна Александровна. С/С++. Программирование на языке высокого уровня: учебник для вузов / Т. А. Павловская. - СПб.: Питер, 2007. - 464 с. (47 экз. библиотека ТУСУР)

2.2. Дополнительная литература

1. Юров, Виктор Иванович. Assembler: Учебное пособие для вузов / В. И. Юров. - 2-е изд. - СПб.: Питер, 2005. - 636с. (20 экз. библиотека ТУСУР)
2. Абель, Питер. Язык Ассемблера для IBM PC и программирования: Пер. с англ. / П. Абель; пер. Ю. В. Сальников. - М.: Высшая школа, 1992. - 447 с. (24 экз. библиотека ТУСУР)

2.3. Перечень пособий, методических указаний и материалов, используемых в учебном процессе

1. Фефелов, Николай Петрович. Организация ЭВМ и систем. Введение в ассемблер: учебное пособие к лабораторным работам для студентов специальности 230105 - Программное обеспечение вычислительной техники и автоматизированных систем / Н. П. Фефелов; Федеральное агентство по образованию, Томский государственный университет систем управления и радиоэлектроники, Кафедра автоматизированных систем управления. - Томск: ТУСУР, 2006. - 51 с. (85 экз. библиотека ТУСУР)