МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра вычислительные системы и технологии

Лабораторная работа № 1

Программная модель процессора

Вариант №13

ОТЧЕТ

по лабораторной работе

по дисциплине

Принципы и методы  
организации системных программных средств

РУКОВОДИТЕЛЬ:

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

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сапожников В.О.

19-ИВТ-3

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

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

Нижний Новгород 2021

1. **Цель работы**
2. Освоить систему меню отладчика Turbo Debugger (td.exe)
3. Изучить рамки окна CPU и локальные меню.
4. Научиться управлять отображением и редактированием информации в рамках окна CPU
5. Научится сохранять информацию, отображаемую в окне CPu в виде текстовых (в журнале отладчика Log) и бинарных файлов.
6. Составить в кодах, ввести и выполнить в отладчике несколько команд перессылки с различными метолами адрессации, команды условного перехода и цикла.
7. Написать программу. Ввести программу в отладчике, записать в память необходимые данные, выполнить программу в отладчике. Получить распечатки программы, дампов до и после выполнения программы.
8. **Задание**

Подготовить средствами отладчика в памяти данные для задачи: исходный массив кодов {a(1),a(2),...,a(n)}. Адрес массива и число элементов выбрать самостоятельно. Написать программу, выполняющую пересылку элементов исходного массива (или преобразование и пересылку), удовлетворяющих некоторому условию в другую область памяти.

**13 вариант:** переслать слова, в которых число сброшенных битов больше половины.

1. **Алгоритм.**

НАЧАЛО

Ввести данные (массив)

Цикл ДЛЯ каждого элемента массива:

Подсчитать кол-во установленных бит (кол-во единиц в двоичном коде)

ЕСЛИ кол-во единиц ≧ 8:

ТО пропустить данный элемент

ИНАЧЕ переслать данный элемент

Всё – цикл

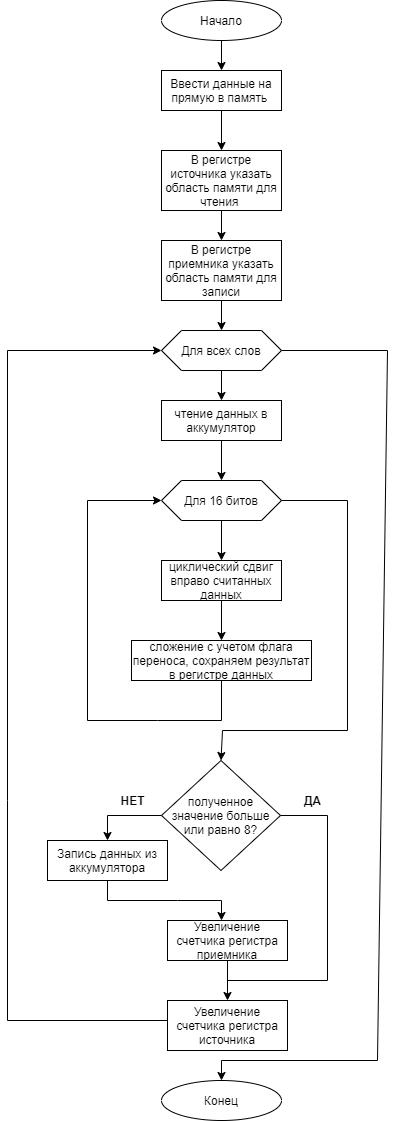
КОНЕЦ

Особенности:

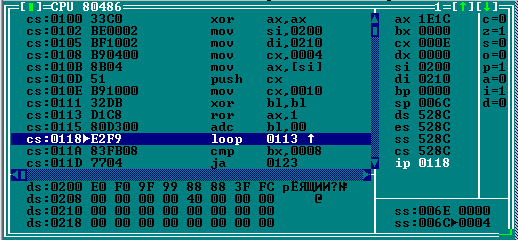
Так как, согласно заданию, мы работаем со словами (2 байта = 16 бит) необходимо это учитывать при увеличении счетчиков регистров источника и приемника данных.

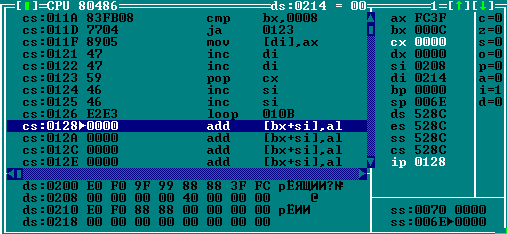
Для подсчета установленных бит используется операция **adc** при сложении с 0 (данная реализация является лишь одним из вариантов решения данной задачи). Данная команда выполняет сложение с учетом флага переноса.

**Реализация в отладчике**



1. **Дампы памяти**





Входные данные:

1111 0000 1110 0000b → E0 F0h

1001 1001 1001 1111b → 9F 99h

1000 1000 1000 1000b → 88 88h

1111 1100 0011 1111b → 3F FCh

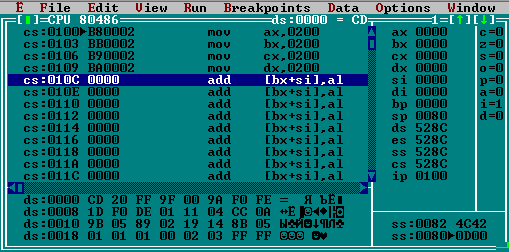
Выходные данные:

1111 0000 1110 0000b → E0 F0h

1000 1000 1000 1000b → 88 88h

1. **Способы адресации**

**Непосредственная адресация** – операнд располагается в последних байтах команды.



**B80002 mov ax,0200**

**B8 = 1011 1000**

**КОП** команда пересылки непосредственных данных в регистр (сокращённая команда)

**w = 1** операнд слово

**reg** регистр **ax**

**BB0002 mov bx,0200**

**BB= 1011 1011**

**КОП** команда пересылки непосредственных данных в регистр (сокращённая команда)

**w = 1** операнд слово

**reg** регистр **bx**

**B90002 mov cx,0200**

**B8 = 1011 1001**

**КОП** команда пересылки непосредственных данных в регистр (сокращённая команда)

**w = 1** операнд слово

**reg** регистр **cx**

**BA0002 mov dx,0200**

**B8 = 1011 1011**

**КОП** команда пересылки непосредственных данных в регистр (сокращённая команда)

**w = 1** операнд слово

**reg** регистр **dx**

**Регистровая адресация** – операнд находится в одном из общих (сегментных) регистрах.

**8CCB mov bx,cs**

**8CCA = 1000 1100 1100 1010**

**КОП** команда пересылки из сегментного регистра в регистр

**mod** операнд в регистре

**reg** сегментный регистр **CS**

**r/m** регистр **DX**

**Прямая (абсолютная) адресация** – эффективный адрес является частью команды (байты смещения в команде). Прямая адресация может быть дальней с указанием дальнего адреса сегмент:смещение, такая адресация используется только в командах межсегментных переходов и вызовов подпрограмм.

**8B1E1003 mov bx, [0310h]**

**8B1E1003 = 1000 1011 0001 1110**

**КОП** команда пересылки

**d=1** пересылка в регистр код которого в reg

**w = 1** операнд слово

**mod** операнд в памяти без смещения

**reg** регистр **BX**

**r/m** disp 16 смещение

**Косвенная адресация -** операнд находится в памяти, для вычисления эффективного адреса используются значения базовых, индексных регистров и, возможно, смещение.

**8B940002 mov dx, [si+0200]**

**8B94 = 1000 1011 1001 0100**

**КОП** команда пересылки

**d=1** пересылка в регистр код которого в reg

**w = 1** операнд слово

**mod** операнд в памяти со смещением в 2 байта

**reg** регистр **DX**

**r/m** SI+disp16

**Неявная адресация -** адресуемый объект указывается с помощью одного байта команды вместе с кодом операции без выделения специального поля.

**51 push cx**

**51 = 0101 0001**

**КОП** команда загрузки в стек из регистра

**reg** регистр **CX**

1. **Вывод**

При выполнении данной работы были получены навыки работы с отладчиком ***Turbo Debugger***, были изучены структуры и форматы команд. Разработана простейшая программа на ассемблере и выполнена в пошаговом режиме.