МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №4**

по дисциплине

«Основы профессиональной деятельности»

Вариант № 3102

***Выполнил:***

Студент группы P3131

Дворкин Борис Александрович

***Преподаватель:***

Клименков Сергей

Викторович

Санкт-Петербург, 2023 г.

Содержание

[Текст задания 3](#_Toc127641422)

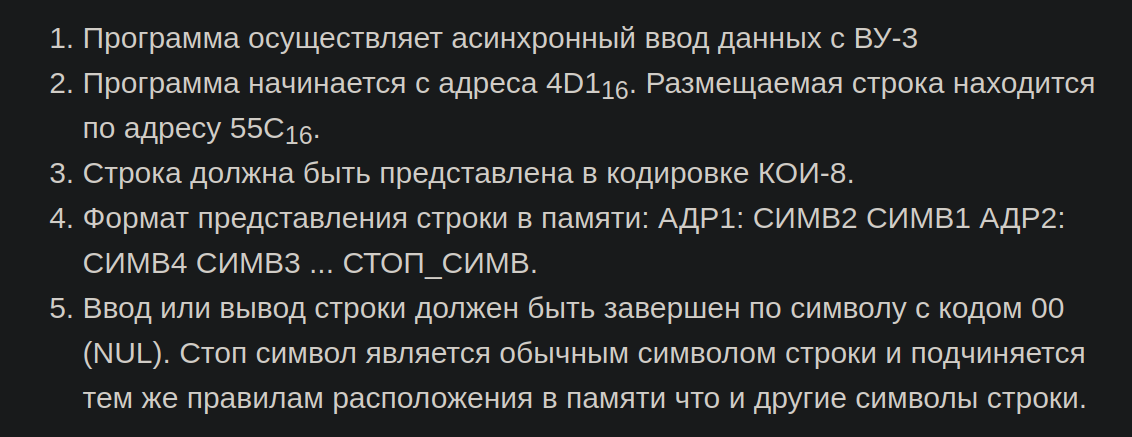
[Описание программы 3](#_Toc127641423)

[Вывод 6](#_Toc127641424)

[Таблица трассировки 7](#_Toc127641425)

Текст задания

Описание программы



**Передаваемое сообщение**: «Колодец»

В кодировке КОИ-8: EB D3 CC D3 D4 D5 C3

В кодировке UTF-8: D09A D0BE D0BB D0BE D0B4 D0B5 D186

В кодировке UTF-16: 041A 043E 043B 043E 0434 0435 0446

В кодировке WINDOWS-1251: CA EE EB EE E4 E5 F6

В кодировке ISO-8859-5: BA BE C1 BE D4 D5 DE

**Текст программы на ассемблере:**

ORG 0x4D1 ; Адрес начала программы

res: WORD 0x55C ; Ссылка на результат

finish: WORD 0x00 ; Стоп-символ

temp: WORD ? ; Ячейка для записи нечётных символов

START: CLA ; Очистка аккумулятора

s1: IN 7 ; Ожидание ввода нечетного символа

AND #0x40; Проверка на наличие введенного символа

BEQ s1 ; Нет - "Спин-луп"

IN 6 ; Ввод байта в AC

ST (res) ; Сохраняем символ в результат

ST temp ; Сохраняем символ во временную переменную

CMP finish ; Проверяем на стоп-символ

BEQ exit ; Если стоп-символ - выход

CLA ; Очистка аккумулятора

s2: IN 7 ; Ожидание ввода четного символа

AND #0x40 ; Проверка на наличие введенного символа

BEQ s2 ; Нет - "Спин-луп"

IN 6 ; Вывод байта в AC

SWAB ; Перемещаем четный символ в старший байт

OR temp ; Совмещаем с 1-м символом

ST (res) ; Сохраняем в память по ссылки

SUB temp ; Вычитаем 1-й символ

SWAB ; Перемещаем четный символ в младший байт

CMP finish ; Проверяем на стоп-символ

BEQ exit ; Если стоп-символ - выход

LD (res)+ ; Инкрементируем ссылку на результат

CLA ; Очистка аккумулятора

JUMP s1 ; Возвращаемся в начало цикла

exit: LD (res)+ ; Инкрементируем ссылку на результат

HLT ; Остановка программы

**Текст исходной программы:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарии** |
| 06F | 055С | res | Ссылка на результат |
| 070 | 0000 | finish | Стоп-символ |
| 071 | 0000 | temp | Ячейка для записи нечетных символов |
| 072 | + 0200 | CLA | Очистка аккумулятора |
| 073 | 1207 | IN 7 | Чтение регистра состояния ВУ-3 |
| 074 | 2F40 | AND #0x40 | Проверка на наличие введенного символа |
| 075 | F0FD | BEQ IP-3 | Нет - "Спин-луп" |
| 076 | 1206 | IN 6 | Чтение регистра данных ВУ-3 |
| 077 | E8F7 | ST (IP-9) | Сохраняем символ в результат |
| 078 | EEF8 | ST IP-8 | Сохраняем символ во временную переменную |
| 079 | 7EF6 | CMP IP-10 | Проверяем на стоп-символ |
| 07A | F00F | BEQ IP+15 | Если стоп-символ - выход |
| 07B | 0200 | CLA | Очистка аккумулятора |
| 07C | 1207 | IN 7 | Чтение регистра состояния ВУ-3 |
| 07D | 2F40 | AND #0x40 | Проверка на наличие введенного символа |
| 07E | F0FD | BEQ IP-3 | Нет - "Спин-луп" |
| 07F | 1206 | IN 6 | Чтение регистра данных ВУ-3 |
| 080 | 0680 | SWAB | Обмен старшего и младшего байтов |
| 081 | 3EEF | OR IP-17 | Логическое или ^(^MEM & ^AC) → AC |
| 082 | E8EC | ST (IP-20) | Сохраняем в память по ссылки |
| 083 | 6EED | SUB IP-19 | Вычитание AC – MEM → AC |
| 084 | 0680 | SWAB | Обмен старшего и младшего байтов |
| 085 | 7EEA | CMP IP-22 | Проверяем на стоп-символ |
| 086 | F003 | BEQ IP+3 | Если стоп-символ - выход |
| 087 | AAE7 | LD (IP-25)+ | Инкрементируем ссылку на результат |
| 088 | 0200 | CLA | Очистка аккумулятора |
| 089 | CEE9 | JUMP IP-23 | Возвращаемся в начало цикла |
| 08A | AAE4 | LD (IP-30)+ | Инкрементируем ссылку на результат |
| 08B | 0100 | HLT | Остановка программы |
| … | | | |
| 562 | 0000 | 0000 | Строка результата |

**Описание программы**

Программа осуществляет посимвольный асинхронный ввод данных с ВУ-3, посимвольно записывает их в память. Программа будет получать символы до тех пор, пока на ВУ-3 не будет введен стоп-символ с кодировкой 0x00, который она запишет в память и прекратит свое выполнение.

**Область представления**

* res – 11-разрядная ячейка со ссылкой на результат.
* finish – 16-разрядная константа.
* temp – 16-разрядная ячейка для временного хранения введенных символов.
* XXX - ? – 16-разрядные ячейки, хранящие в себе по два символа в кодировке КОИ-8.

**Расположение данных в памяти**

* – команды;
* – исходные данные;
* XXX - ? – результат.

**Адреса первой и последней выполняемой команды**

* Адрес первой команды:
* Адрес последней команды:

**Область допустимых значений**

* res (указатель на ячейки массива, хранящий результат ввода) ∈ [55С;7FF]
* temp (ячейка для записи нечетных символов) ∈ [0;255], т.к. в нее записывается только 1 символ из 8 бит.
* Введенный символ: [00; FF]

Адрес первого элемента массива равен 55C по условию. Т.к. 7FF – 55С (16 CC) = 675(10 CC) – кол-во ячеек, которые могут использоваться для записи результата => 675\*2 = 1350 – максимально возможное кол-во введенных символов (т.к. в данной кодировке символ занимает 1 байт), включая обязательный стоп-символ => Кол-во введенных символов ∈ [1;1350].

Вывод

При выполнении данной лабораторной работы я познакомился с асинхронным вводом-выводом данных в БЭВМ, узнал о внешних устройствах, их регистрах и принципах работы. Также, я познакомился с представлением данных в различных кодировках и попрактиковался с вводом данных на ВУ-3.

Таблица трассировки

Строка для трассировки: «Колодец», кодировка КОИ-8:

EB D3 CC D3 D4 D5 C3

*Трассировка производится для первых двух символов.*

**Таблица трассировки:**

Адр Знчн IP CR AR DR SP BR AC PS NZVC Адр Знчн