Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №5

Асинхронный обмен данными с ВУ

Вариант 16509

Выполнил:

Григорьев Даниил Александрович

Группа P3116

Преподаватель:

Ткешелашвили Нино Мерабиевна

Содержание

[Задание 3](#_Toc196316531)

[Описание программы 5](#_Toc196316532)

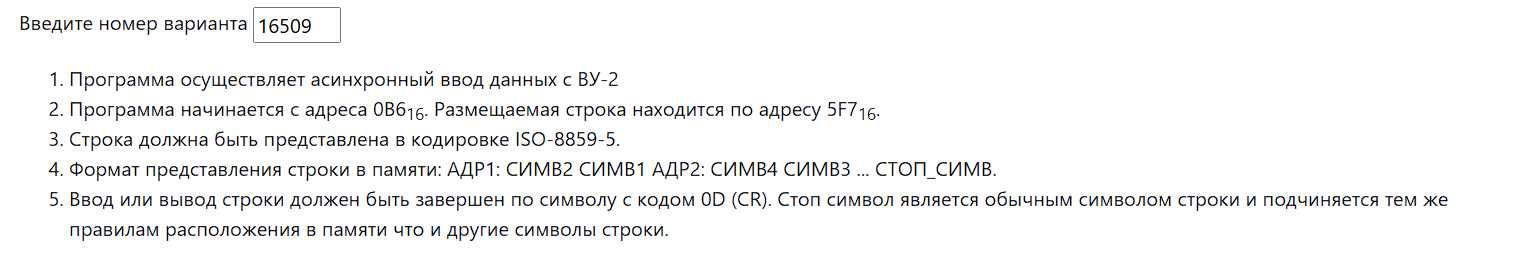
[Область представления 5](#_Toc196316533)

[Трассировка программы 9](#_Toc196316534)

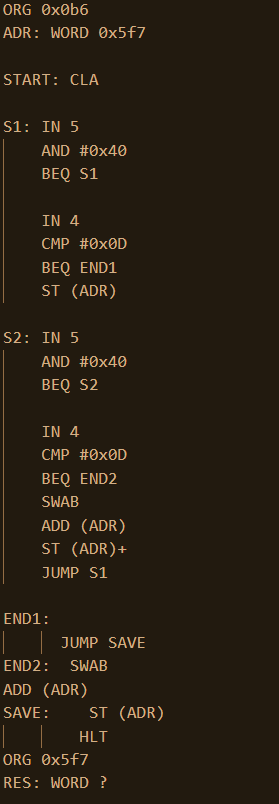
[Вывод 11](#_Toc196316535)

# Задание

По выданному преподавателем варианту разработать программу асинхронного обмена данными с внешним устройством. При помощи программы осуществить ввод или вывод информации, используя в качестве подтверждения данных сигнал (кнопку) готовности ВУ.



**Программа на ассемблере**



**Таблица команд**

# 

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарии** |
| 0B6 | - | ADR | Адрес ячейки результата на текущий момент |
| 0B7 | 200 | CLA | Очистить аккумулятор |
| 0B8 | 1205 | IN #5 | Ожидание ввода |
| 0B9 | 2F40 | AND #0x40 | Проверка статуса |
| 0BA | F0FD | BEQ IP-3 | Спин-луп в ожидании готовности |
| 0BB | 1204 | IN #4 | Считывание символа |
| 0BC | 7F0D | CMP #0x0D | Делаем проверку на стоп-слово |
| 0BD | F00B | BEQ IP+11 | Переходим к маркеру с END1 в случае стоп-слова (случай, когда ввели правое слово = 0x0D) |
| 0BE | E8F7 | ST (ADR) | Записываем правое слово по текущему адресу |
| 0BF | 1205 | IN #5 | Ожидание ввода |
| 0C0 | 2F40 | AND #0x040 | Проверка статуса |
| 0C1 | F0FD | BEQ IP-3 | Спин-луп в ожидании готовности |
| 0C2 | 1204 | IN #4 | Считывание символа |
| 0C3 | 7F0D | CMP 0x0D | Проверка на стоп-слово |
| 0C4 | F005 | BEQ IP+5 | Переходим к маркеру с END2, если вводили левое слово, содержащее 0x0D |
| 0C5 | 0680 | SWAB | Меняем старший и младший байты |
| 0C6 | 48EF | ADD (ADR) | Добавим значение правого символа |
| 0C7 | EAEE | ST (ADR)+ | Сохраним левые и правые символы в текущей ячейке и делаем постинкремент для указания на следующую ячейку |
| 0C8 | CEEF | JUMP IP-17 | Переходим к маркеру S1 (На ячейку 0x0b8) |
| 0C9 | CE02 | JUMP IP+2 | Переходим к маркеру SAVE (на ячейку 0x0CC) |
| 0CA | 0680 | SWAB | Меняем старший и младший байты |
| 0CB | 48EA | ADD (ADR) | Добавим значение правого символа |
| 0CC | E8E9 | ST (ADR) | Сохраним левые и правые символы в текущей ячейке |
| 0CD | 100 | HLT | Завершение программы |
| 5F7 | - | RES | Строка результата |

# Описание программы

Программа осуществляет посимвольный асинхронный ввод данных с ВУ-2. Программа будет получать символы до тех пор, пока на ВУ-2 не будет введен стоп-символ с кодировкой 0x0D, который программа запишет в память и прекратит свое выполнение.

# Область представления

RES -? - 16-разрядные ячейки, хранящие в себе по два символа в кодировке ISO-8859-5

ADR - 11 разрядная ячейка, хранящая адрес текущей ячейки

**Расположение данных в памяти**

ADR (указатель на ячейки массива, хранящий результат ввода) **∈** [5F7, 7FF]

**Таблица трассировки**

Строка для трассировки: ПЕРЕЦ

ISO-8859-5: BF B5 C0 B5 C6

UTF-8: D0 9F D0 95 D0 A0 D0 95 D0 A6

UTF-16: 04 1F 04 15 04 20 04 15 04 26

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Выполняемая команда** | | **Содержимое регистров процессора после выполнения команды** | | | | | | | | | **Ячейка, содержимое которой изменилось после выполнения команды** | |
| Адрес | Код команды | IP | CR | AR | DR | SP | BR | AC | PS | NZVC | Адрес | Новый  код |
| 0B7 | 0200 | 0B8 | 0200 | 0B7 | 0200 | 000 | 00B7 | 0000 | 004 | 0100 |  |  |
| 0B8 | 1205 | 0B9 | 1205 | 0B8 | 1205 | 000 | 00B8 | 0040 | 004 | 0100 |  |  |
| 0B9 | 2F40 | 0BA | 2F40 | 0B9 | 0040 | 000 | 0040 | 0040 | 000 | 0000 |  |  |
| 0BA | F0FD | 0BB | F0FD | 0BA | F0FD | 000 | 00BA | 0040 | 000 | 0000 |  |  |
| 0BB | 1204 | 0BC | 1204 | 0BB | 1204 | 000 | 00BB | 00BF | 000 | 0000 |  |  |
| 0BC | 7F0D | 0BD | 7F0D | 0BC | 000D | 000 | 000D | 00BF | 001 | 0001 |  |  |
| 0BD | F00B | 0BE | F00B | 0BD | F00B | 000 | 00BD | 00BF | 001 | 0001 |  |  |
| 0BE | E8F7 | 0BF | E8F7 | 5F7 | 00BF | 000 | FFF7 | 00BF | 001 | 0001 | 5F7 | 00BF |
| 0BF | 1205 | 0C0 | 1205 | 0BF | 1205 | 000 | 00BF | 0040 | 001 | 0001 |  |  |
| 0C0 | 2F40 | 0C1 | 2F40 | 0C0 | 0040 | 000 | 0040 | 0040 | 001 | 0001 |  |  |
| 0C1 | F0FD | 0C2 | F0FD | 0C1 | F0FD | 000 | 00C1 | 0040 | 001 | 0001 |  |  |
| 0C2 | 1204 | 0C3 | 1204 | 0C2 | 1204 | 000 | 00C2 | 00B5 | 001 | 0001 |  |  |
| 0C3 | 7F0D | 0C4 | 7F0D | 0C3 | 000D | 000 | 000D | 00B5 | 001 | 0001 |  |  |
| 0C4 | F005 | 0C5 | F005 | 0C4 | F005 | 000 | 00C4 | 00B5 | 001 | 0001 |  |  |
| 0C5 | 0680 | 0C6 | 0680 | 0C5 | 0680 | 000 | 00C5 | B500 | 009 | 1001 |  |  |
| 0C6 | 48EF | 0C7 | 48EF | 5F7 | 00BF | 000 | FFEF | B5BF | 008 | 1000 |  |  |
| 0C7 | EAEE | 0C8 | EAEE | 5F7 | B5BF | 000 | FFEE | B5BF | 008 | 1000 | 5F7,  0B6 | B5BF,  5F8 |
| 0C8 | CEEF | 0C9 | CEEF | 0C8 | 00B8 | 000 | FFEF | B5BF | 008 | 1000 |  |  |
| 0B8 | 1205 | 0B9 | 1205 | 0B8 | 1205 | 000 | 00B8 | B500 | 008 | 1000 |  |  |
| 0B9 | 2F40 | 0BA | 2F40 | 0B9 | 0040 | 000 | 0040 | 0000 | 004 | 0100 |  |  |
| 0BA | F0FD | 0B8 | F0FD | 0BA | F0FD | 000 | FFFD | 0000 | 004 | 0100 |  |  |
| 0B8 | 1205 | 0B9 | 1205 | 0B8 | 1205 | 000 | 00B8 | 0040 | 004 | 0100 |  |  |
| 0B9 | 2F40 | 0BA | 2F40 | 0B9 | 0040 | 000 | 00BA | 0040 | 000 | 0000 |  |  |
| 0BA | F0FD | 0BB | F0FD | 0BA | F0FD | 000 | 00BA | 0040 | 000 | 0000 |  |  |
| 0BB | 1204 | 0BC | 1204 | 0BB | 1204 | 000 | 00BB | 000D | 000 | 0000 |  |  |
| 0BC | 7F0D | 0BD | 7F0D | 0BC | 000D | 000 | 000D | 000D | 005 | 0101 |  |  |
| 0BD | F00B | 0C9 | F00B | 0BD | F00B | 000 | 000B | 000D | 005 | 0101 |  |  |
| 0C9 | CE02 | 0CC | CE02 | 0C9 | 00CC | 000 | 0002 | 000D | 005 | 0101 |  |  |
| 0CC | E8E9 | 0CD | E8E9 | 5F8 | 000D | 000 | FFE9 | 000D | 005 | 0101 | 5F8 | 0000D |
| 0CD | 0100 | 0CE | 0100 | 0CD | 0100 | 000 | 00CD | 000D | 005 | 0101 |  |  |

**Дополнительное задание**

Калькулятор. Ввод цифр с ВУ-9 (цифровая клавиатура), вывод результата на ВУ-7 (семисегментный индикатор), реализовать сложение и вычитание знаковых чисел от 1 до 3-х знаков.

Реализация на ассемблере:



# Вывод

В ходе лабораторной работы я разобрался с асинхронным вводом-выводом данных в БЭВМ, узнал о внешних устройствах и их регистрах. Также познакомился с представлением данных в различных кодировках и попрактиковался в написании кода на Ассемблере БЭВМ.