**Министерство науки и высшего образования Российской Федерации**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Факультет ПИиКТ**

**Дисциплина: Основы профессиональной деятельности**

**Лабораторная работа №5**

**Асинхронный обмен**

**данными с ВУ**

**Вариант 7587**

Выполнил: Михайлов Петр Сергеевич

Группа: Р3111

Преподаватель: Остапенко Ольга Денисовна

Санкт-Петербург 2025г.

Содержание

[Задание 3](#_Toc191537569)

[Определение функции, вычисляемой программой 4](#_Toc191537570)

[1. Текст исходный программы 4](#_Toc191537571)

[2. Описание программы 5](#_Toc191537572)

[3. Расположение в БЭВМ программы, исходных данных и результатов 5](#_Toc191537573)

[4. Область представления 5](#_Toc191537574)

[5. Область допустимых значений 6](#_Toc191537575)

[Трассировка программы 7](#_Toc191537576)

[Заключение 8](#_Toc191537577)

# 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.

1. Программа осуществляет асинхронный вывод данных с ВУ-3
2. Программа начинается с адреса 0BE16. Размещаемая строка находится по адресу 5E616.
3. Строка должна быть представлена в кодировке ISO-8859-5.
4. Формат представления строки в памяти: АДР0: ДЛИНА АДР1: СИМВ2 СИМВ1 АДР2: СИМВ4 СИМВ3 ..., где ДЛИНА - 16 разрядное слово, где значащими являются 8 младших бит.
5. Вывод строки начинается со вывода количества символов (1 байт), и должен быть завершен по выводу их необходимого количества.

# Определение функции, вычисляемой программой

1. Текст исходный программы

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарии** |
| 0BE | 05E6 | - | Res - Ссылка на ячейку данных |
| 0BF | 00XX | - | Count - Оставшееся количество символов |
| 0C0 | 0200 | CLA | Очистить аккумулятор:  0 ⇒ AC |
| 0C1 | AEFD | LD (IP-3) | Загружаем ячейку count |
| 0C2 | 1207 | IN 7 | Ожидание действия |
| 0C3 | 2F40 | AND #040 |  |
| 0C4 | F0FD | BEQ (IP-3) |  |
| 0C5 | 1306 | OUT 6 | Вывод длины слов на ВУ-3 |
| 0C6 | 1207 | IN 7 | Ожидание действия |
| 0C7 | 2F40 | AND #040 | Проверка сигнала готовности |
| 0C8 | F0FD | BEQ (IP-3) | Если нет сигнала, то спин-луп |
| 0C9 | AEF7 | LD (IP-9) | Загружаем младшие 8 бит первого слова |
| 0CA | 1306 | OUT 6 | Выводим их на ВУ-3 |
| 0CB | AEF6 | LD (IP-10) | Загружаем переменную count из аккумулятора |
| 0CC | 0740 | DEC | Уменьшаем count на 1 |
| 0CD | EEF4 | ST (IP-12) | Сохраняем переменную count |
| 0CE | 7F00 | CMP #0 | Сравниваем count с нулем |
| 0CF | F00C | BEQ (IP+12) | Если count = 0, переходим к остановке |
| 0D0 | 1207 | IN 7 | Ожидание действия |
| 0D1 | 2F40 | AND #040 | Проверка сигнала готовности |
| 0D2 | F0FD | BEQ (IP-3) | Если нет сигнала, то спин-луп |
| 0D3 | AAED | LD (IP-19)+ | Загружаем вторые 8 бит слова и делаем инкремент в следующую ячейку |
| 0D4 | 0680 | SWAB | Обмен байтов слова |
| 0D5 | 1306 | OUT 6 | Выводим на ВУ-3 |
| 0D6 | AEEA | LD (IP-22) | Выгружаем значение count |
| 0D7 | 0740 | DEC | Уменьшаем count на 1 |
| 0D8 | EEE8 | ST (IP-24) | Сохраняем значение count |
| 0D9 | 7F00 | CMP #0 | Сравниваем count с нулем |
| 0DA | F00C | BEQ (IP+1) | Если count = 0, переходим к остановке |
| 0DB | CEEA | JUMP (IP-22) | Возвращаемся к чтению младшего байта слова |
| 0DC | AEE1 | LD (IP-31) | Выгружаем адрес |
| 0DD | 6F02 | SUB #02 | Сдвигаем адрес в исходное положение |
| 0DE | EEDF | ST (IP-33) | Сохраняем изначальное значение адреса |
| 0DF | AEDF | LD (IP-33) | Выгружаем значение count |
| 0E0 | 4F04 | ADD #04 | Возвращаем count в исходное состояние |
| 0E1 | EEDD | ST (IP-35) | Сохраняем count |
| 0E2 | 0100 | HLT | Останов |
| … | | | |
| 5E6 | CAE8 | - | Ячейка данных |
| 5E7 | D4C0 | - | Ячейка данных |

2. Код программы на Ассемблере

ORG 0x0BE

ADDR: WORD $STRING ; Ссылка на ячейку данных

count: WORD 4 ; Оставшееся количество символов

START:

CLA

IN 7 ; Ожидание действия

AND #0x40

BEQ START

LD count

OUT 6 ; Вывод длины слов на ВУ-3

S1: IN 7 ; Ожидание действия

AND #0x40 ; Проверка сигнала готовности

BEQ S1 ; Если нет сигнала, то спин-луп

LD (ADDR) ; Косвенная адресация, загружаем младшие 8 бит первого слова

OUT 6 ; Выводим их на ВУ-3

LD count ; Загружаем переменную count из аккумулятора

DEC ; Уменьшаем count на 1

ST count ; Сохраняем переменную count

CMP #0 ; Сравниваем count с нулем

BEQ STOP\_POINT ; Если count = 0, переходим к остановке

S2: IN 7 ; Ожидание действия

AND #0x40 ; Проверка сигнала готовности

BEQ S2 ; Если нет сигнала, то спин-луп

LD (ADDR)+ ; Косвенная адресация, загружаем вторые 8 бит слова и делаем инкремент в следующую ячейку

SWAB ; Обмен байтов слова

OUT 6 ; Выводим на ВУ-3

LD count ; Выгружаем значение count

DEC ; Уменьшаем count на 1

ST count ; Сохраняем значение count

CMP #0 ; Сравниваем count с нулем

BEQ STOP\_POINT ; Если count = 0, переходим к остановке

JUMP S1 ; Возвращаемся к чтению младшего байта слова

STOP\_POINT:

LD ADDR ; Выгружаем адрес

SUB #2 ; Сдвигаем адрес в исходное положение

ST ADDR ; Сохраняем изначальное значение адреса

LD count ; Выгружаем значение count

ADD #4 ; Возвращаем count в исходное состояние

ST count ; Сохраняем count

HLT ; Останов

ORG 0x5E6

STRING:

WORD 0xCAE8 ; Слово ШКАФ в ISO-8859-5: E8 CA C0 D4

WORD 0xD4C0

3. Описание программы

Слово ШКАФ в ISO-8859-5: E8 CA C0 D4

Слово ШКАФ в UTF-8: D0 A8 D0 9A D0 90 D0 AF

Слово ШКАФ в UTF-16: A8 04 9A 04 90 04 AF 04

Программа осуществляет посимвольный асинхронный вывод данных из памяти на ВУ-3. Программа будет считывать символы из памяти до тех пор, пока не будет достигнута общая длина строк, которую мы указываем заранее. Тогда программа прекратит свое выполнение.

4. Область представления

res – 11-разрядная ячейка со ссылкой на результат.

count – 16-разрядная ячейка, у которой значащие младшие 8 бит.

5E6 - ? – 16-разрядные ячейки, хранящие в себе по два символа в кодировке ISO-8859-5.

5. Область допустимых значений

res ∈ [0x5E6; 0x7FF]

Хранимый в памяти символ: [00; FF]

Адрес первого элемента массива равен 0x5E6 по условию. Т.к. 2047 – 1510 = 537 – количество ячеек, которые могут использоваться для хранения результата => 537\*2 = 1047 – максимально возможное количество хранимых символов (т.к. в данной кодировке символ занимает 1 байт) => Кол-во хранимых символов ∈ [1;1047].

6. Расположение в памяти данных

0C0 – 0D9 – команды

5E6 - ? – исходные данные

# Трассировка программы

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| 0C0 | 0200 | 0C0 | 0000 | 000 | 0000 | 000 | 0000 | 0000 | 000 | 0100 |  |  |
| 0C1 | 1207 | 0C2 | 1207 | 0C1 | 1207 | 000 | 00C1 | 0000 | 000 | 0100 |  |  |
| 0C2 | 2F40 | 0C3 | 2F40 | 0C2 | 0040 | 000 | 0040 | 0000 | 000 | 0100 |  |  |
| 0C3 | F0FC | 0C0 | F0FC | 0C3 | F0FC | 000 | FFFC | 0000 | 000 | 0100 |  |  |
| 0C4 | AEFA | 0C5 | AEFA | 0BF | 0004 | 000 | FFFA | 0004 | 000 | 0000 |  |  |
| 0C5 | 1306 | 0C6 | 1306 | 0C5 | 1306 | 000 | 00C5 | 0004 | 000 | 0000 |  |  |
| 0C6 | 1207 | 0C7 | 1207 | 0C6 | 1207 | 000 | 00C6 | 0000 | 000 | 0000 |  |  |
| 0C7 | 2F40 | 0C8 | 2F40 | 0C7 | 0040 | 000 | 0040 | 0000 | 000 | 0100 |  |  |
| 0C8 | F0FD | 0C6 | F0FD | 0C8 | F0FD | 000 | FFFD | 0000 | 000 | 0100 |  |  |
| 0C9 | A8F4 | 0CA | A8F4 | 5E6 | D0CA | 000 | FFF4 | D0CA | 000 | 1000 |  |  |
| 0CA | 1306 | 0CB | 1306 | 0CA | 1306 | 000 | 00CA | D0CA | 000 | 1000 |  |  |
| 0CB | AEF3 | 0CC | AEF3 | 0BF | 0004 | 000 | FFF3 | 0004 | 000 | 0000 |  |  |
| 0CC | 0740 | 0CD | 0740 | 0CC | 0740 | 000 | 00CC | 0003 | 000 | 0001 |  |  |
| 0CD | EEF1 | 0CE | EEF1 | 0BF | 0003 | 000 | FFF1 | 0003 | 000 | 0001 | 0BF | 0003 |
| 0CE | 7F00 | 0CF | 7F00 | 0CE | 0000 | 000 | 0000 | 0003 | 000 | 0001 |  |  |
| 0CF | F00C | 0D0 | F00C | 0CF | F00C | 000 | 00CF | 0003 | 000 | 0001 |  |  |
| 0D0 | 1207 | 0D1 | 1207 | 0D0 | 1207 | 000 | 00D0 | 0040 | 000 | 0001 |  |  |
| 0D1 | 2F40 | 0D2 | 2F40 | 0D1 | 0040 | 000 | 0040 | 0040 | 000 | 0001 |  |  |
| 0D2 | F0FD | 0D3 | F0FD | 0D2 | F0FD | 000 | 00D2 | 0040 | 000 | 0001 |  |  |
| 0D3 | AAEA | 0D4 | AAEA | 5E6 | D0CA | 000 | FFEA | D0CA | 000 | 1001 | 0BE | 05E7 |
| 0D4 | 0680 | 0D5 | 0680 | 0D4 | 0680 | 000 | 00D4 | CAD0 | 000 | 1001 |  |  |
| 0D5 | 1306 | 0D6 | 1306 | 0D5 | 1306 | 000 | 00D5 | CAD0 | 000 | 1001 |  |  |
| 0D6 | AEE8 | 0D7 | AEE8 | 0BF | 0003 | 000 | FFE8 | 0003 | 000 | 0001 |  |  |
| 0D7 | 0740 | 0D8 | 0740 | 0D7 | 0740 | 000 | 00D7 | 0002 | 000 | 0001 |  |  |
| 0D8 | EEE6 | 0D9 | EEE6 | 0BF | 0002 | 000 | FFE6 | 0002 | 000 | 0001 | 0BF | 0002 |
| 0D9 | 7F00 | 0DA | 7F00 | 0D9 | 0000 | 000 | 0000 | 0002 | 000 | 0001 |  |  |
| 0DA | F001 | 0DB | F001 | 0DA | F001 | 000 | 00DA | 0002 | 000 | 0001 |  |  |
| 0DB | CEEA | 0C6 | CEEA | 0DB | 00C6 | 000 | FFEA | 0002 | 000 | 0001 |  |  |
| 0C6 | 1207 | 0C7 | 1207 | 0C6 | 1207 | 000 | 00C6 | 0040 | 000 | 0001 |  |  |
| 0C7 | 2F40 | 0C8 | 2F40 | 0C7 | 0040 | 000 | 0040 | 0040 | 000 | 0001 |  |  |
| 0C8 | F0FD | 0C9 | F0FD | 0C8 | F0FD | 000 | 00C8 | 0040 | 000 | 0001 |  |  |
| 0C9 | A8F4 | 0CA | A8F4 | 5E7 | C9C0 | 000 | FFF4 | C9C0 | 000 | 1001 |  |  |
| 0CA | 1306 | 0CB | 1306 | 0CA | 1306 | 000 | 00CA | C9C0 | 000 | 1001 |  |  |
| 0CB | AEF3 | 0CC | AEF3 | 0BF | 0002 | 000 | FFF3 | 0002 | 000 | 0001 |  |  |
| 0CC | 0740 | 0CD | 0740 | 0CC | 0740 | 000 | 00CC | 0001 | 000 | 0001 |  |  |
| 0CD | EEF1 | 0CE | EEF1 | 0BF | 0001 | 000 | FFF1 | 0001 | 000 | 0001 | 0BF | 0001 |
| 0CE | 7F00 | 0CF | 7F00 | 0CE | 0000 | 000 | 0000 | 0001 | 000 | 0001 |  |  |
| 0CF | F00C | 0D0 | F00C | 0CF | F00C | 000 | 00CF | 0001 | 000 | 0001 |  |  |
| 0D0 | 1207 | 0D1 | 1207 | 0D0 | 1207 | 000 | 00D0 | 0040 | 000 | 0001 |  |  |
| 0D1 | 2F40 | 0D2 | 2F40 | 0D1 | 0040 | 000 | 0040 | 0040 | 000 | 0001 |  |  |
| 0D2 | F0FD | 0D3 | F0FD | 0D2 | F0FD | 000 | 00D2 | 0040 | 000 | 0001 |  |  |
| 0D3 | AAEA | 0D4 | AAEA | 5E7 | C9C0 | 000 | FFEA | C9C0 | 000 | 1001 | 0BE | 05E8 |
| 0D4 | 0680 | 0D5 | 0680 | 0D4 | 0680 | 000 | 00D4 | C0C9 | 000 | 1001 |  |  |
| 0D5 | 1306 | 0D6 | 1306 | 0D5 | 1306 | 000 | 00D5 | C0C9 | 000 | 1001 |  |  |
| 0D6 | AEE8 | 0D7 | AEE8 | 0BF | 0001 | 000 | FFE8 | 0001 | 000 | 0001 |  |  |
| 0D7 | 0740 | 0D8 | 0740 | 0D7 | 0740 | 000 | 00D7 | 0000 | 000 | 0101 |  |  |
| 0D8 | EEE6 | 0D9 | EEE6 | 0BF | 0000 | 000 | FFE6 | 0000 | 000 | 0101 | 0BF | 0000 |
| 0D9 | 7F00 | 0DA | 7F00 | 0D9 | 0000 | 000 | 0000 | 0000 | 000 | 0101 |  |  |
| 0DA | F001 | 0DC | F001 | 0DA | F001 | 000 | 0001 | 0000 | 000 | 0101 |  |  |
| 0DC | 0100 | 0DD | 0100 | 0DC | 0100 | 000 | 00DC | 0000 | 000 | 0101 |  |  |

# Заключение

В ходе выполнения данной лабораторной работы я познакомился с асинхронным обменом данных с ВУ и ассемблером БЭВМ.