*Факультет программной инженерии и компьютерной техники*

*Направление подготовки: 09.03.04 – Программная инженерия,*

*Системное и прикладное программное обеспечение*

*Дисциплина «Программирование»*

**Лабораторная работа по ОПД №5**

**Вариант №6394**

Выполнил:

Ткачев Денис Владимирович

Группа P3111

Преподаватели:

Остапенко Ольга Денисовна

Оглавление

[Задание 2](#_Toc192711942)

[Текст исходной программы 3](#_Toc192711943)

[Описание программы 3](#_Toc192711944)

[Текст программы на ассемблере: 3](#_Toc192711945)

[Текст исходной программы 3](#_Toc192711946)

[Назначение программы: 5](#_Toc192711947)

[Расположение в памяти БЭВМ программы, исходных данных и результатов 5](#_Toc192711948)

[Область представления 5](#_Toc192711949)

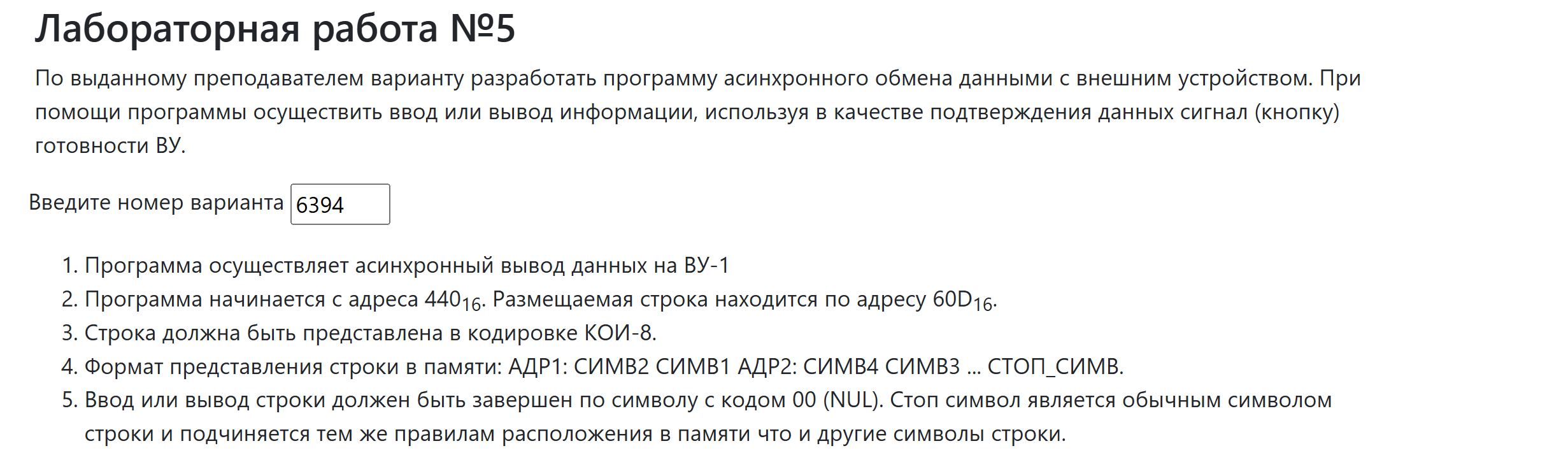
[Область допустимых значений 5](#_Toc192711950)

[Таблица трассировки 6](#_Toc192711951)

[Вывод 7](#_Toc192711952)

## 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 на ВУ-1
2. Программа начинается с адреса 44016. Размещаемая строка находится по адресу 60D16.
3. Строка должна быть представлена в кодировке КОИ-8.
4. Формат представления строки в памяти: АДР1: СИМВ2 СИМВ1 АДР2: СИМВ4 СИМВ3 ... СТОП\_СИМВ.
5. Ввод или вывод строки должен быть завершен по символу c кодом 00 (NUL). Стоп символ является обычным символом строки и подчиняется тем же правилам расположения в памяти что и другие символы строки.

### Текст исходной программы на ассемблере:

**Асинхронный вывод данных (в кодировке КОИ-8) на ВУ-1 (до NUL-символа 0x00)**

ORG 0x440 ; Адрес начала программы

STR: WORD 0x60D ; Размещаемая строка находится по адресу 60D

STOP: WORD 0x00 ; Стоп-символ

MASK\_FIRST\_SYMBOL: WORD 0x00FF ; Маска для того, чтобы остался младший байт

TEMPORARY: WORD ? ; хранится копия аккумулятора

START: CLA ; Очистка аккумулятора

S1: IN 3 ; Ожидание начала записи нечетного символа в ВУ-1

AND #0x40 ; Бит 6 SR == 0 (кнопка "ГОТОВ" нажата?)

BEQ S1 ; Нет - "Спин-луп"

LD (STR)+ ; Загрузка двух символов в порядке СИМ\_ЧЁТ-СИМ\_НЕЧ, инкрементация ссылки на следующие два символа

ST TEMPORARY ; Сохранение копии акуммулятора во временную переменную

OUT 2 ; Загрузка нечётного символа в УВ-1

AND MASK\_FIRST\_SYMBOL; Остаётся младший байт, нечётный символ, 0-СИМ\_Нечётный

CMP STOP ; Проверяем на стоп-символ

BEQ EXIT ; Если стоп-символ, то выход

CLA ; Очистка аккумулятора

S2: IN 3 ; Ожидание начала записи четного символа в ВУ-1

AND #0x40 ; Бит 6 SR == 0 (кнопка "ГОТОВ" нажата?)

BEQ S2 ; Нет - "Спин-луп"

OR TEMPORARY ; Берём копию двух символов из временной переменной СИМ\_ЧЁТ-СИМ\_НЕЧ

SWAB ; Перемещаем четный символ в младший байт, нечётный символ в старший байт, СИМ\_НЕЧ-СИМ\_ЧЁТ

OUT 2 ; Загрузка чётного символа в УВ-1

AND MASK\_FIRST\_SYMBOL ; Остаётся младший байт, чётный символ, 0-СИМ\_Чётный

CMP STOP ; Проверяем на стоп-символ

BEQ EXIT ; Если стоп-символ, то выход

CLA

JUMP S1 ; Возвращаемся в начало цикла для считывания следующей пары символов

EXIT: CLA ; Очистка аккумулятора

ST TEMPORARY ; чистка временной переменной

HLT ; Остановка программы

### Текст исходной программы

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарий** |
| 440 | 060D | STR | Адрес для считывания строк |
| 441 | 0x00 | STOP | Стоп-символ |
| 442 | 0x00FF | MASK\_FIRST\_SYMBOL | Маска для того, чтобы остался младший байт |
| 443 | 0x00 | TEMPORARY | Хранение копии аккумулятора |
| 444 | 0200 | CLA | Очистка аккумулятора |
| 445 | 1203 | IN 0x03 | Ожидание начала записи четного символа в ВУ-1 |
| 446 | 2F40 | AND #0x40 | Бит 6 SR == 0 (кнопка "ГОТОВ" нажата?) |
| 447 | F0FD | BEQ IP-3 | Нет - "Спин-луп" |
| 448 | AAF7 | LD (IP-9)+ | Загрузка двух символов в порядке СИМ\_ЧЁТ-СИМ\_НЕЧ, инкрементация ссылки на следующие два символа |
| 449 | EEF9 | ST IP – 7 | Сохранение копии акуммулятора во временную переменную |
| 44A | 1302 | OUT 0x02 | Загрузка нечётного символа в УВ-1 |
| 44B | 2EF6 | AND IP-10 | Остаётся младший байт, нечётный символ, 0-СИМ\_Нечётный |
| 44C | 7EF4 | CMP IP-12 | Проверяем на стоп-символ |
| 44D | F00C | BEQ IP+12 | Если стоп-символ, то выход |
| 44E | 0200 | CLA | Очистка аккумулятора |
| 44F | 1203 | IN 0x03 | Ожидание начала записи четного символа в ВУ-1 |
| 450 | 2F40 | AND #0x40 | Бит 6 SR == 0 (кнопка "ГОТОВ" нажата?) |
| 451 | F0FD | BEQ IP-3 | Нет - "Спин-луп" |
| 452 | 3EF0 | OR IP-16 | Берём копию двух символов из временной переменной СИМ\_ЧЁТ-СИМ\_НЕЧ |
| 453 | 0680 | SWAB | Перемещаем четный символ в младший байт, нечётный символ в старший байт, СИМ\_НЕЧ-СИМ\_ЧЁТ |
| 454 | 1302 | OUT 0x02 | Загрузка чётного символа в УВ-1 |
| 455 | 2EEC | AND IP-20 | Остаётся младший байт, чётный символ, 0-СИМ\_Чётный |
| 456 | 7EEA | CMP IP-22 | Проверяем на стоп-символ |
| 457 | F002 | BEQ IP+2 | Если стоп-символ, то выход |
| 458 | 0200 | CLA | Очистка аккумулятора |
| 459 | CEEB | JUMP-21 | Возвращаемся в начало цикла для считывания следующей пары символов |
| 45A | 0200 | CLA | Очистка аккумулятора |
| 45B | EEE7 | ST IP-25 | Очистка временной переменной |
| 45С | 0100 | HLT | Остановка программы | |
| … | | | | |
| 60D | 0000 | 0000 |  | |

Назначение программы:

Посимвольный асинхронный вывод строки, закодированной в КОИ-8(R) на ВУ-1. Программа будет отправлять символы на ВУ-1 до тех пор, пока не будет встречен стоп-символ с кодировкой 0x00, который она отправит на ВУ-1 и прекратит своё выполнение.

## Расположение в памяти БЭВМ программы, исходных данных и результатов

0x440 - Адрес для считывания 16-разрядных ячеек, хранящие в себе по два символа в кодировке КОИ-8.

0x441- 16-разрядная константа, стоп-символ для окончания вывода на ВУ-1.

0x442 – 16-разрядное число, маска для того, чтобы остался младший байт.

0x443 – Хранение копии аккумулятора.

0x444 – 0x45C – программа

0x60D – XXX? – строки для считывания символов

## Область представления

STR – 11-разрядная ячейка для обращения к памяти

TEMPORARY – 16-разрядная ячейка

STOP – 16-разрядная ячейка

MASK\_FIRST\_SYMBOL – 16-разрядная ячейка

## Область допустимых значений

STR (указатель на ячейки памяти, хранящие символы строки) ∈ [60D; 7FF]

TEMPORARY (копия аккумулятора) ∈ [0; 2^16-1], т.к. в неё записывается 2 символа – 16 битов.

Символ для вывода – 8 бит [00;FF]

Адрес первого элемента массива равен 60D по условию. Т.к. 7FF – 60D (16 CC) = 498(10 CC) – кол-во ячеек, которые могут использоваться для вывода на ВУ-1 => 498\*2 = 996 – максимально возможное кол-во символов, которые можно использовать для вывода на ВУ-1 (т.к. в данной кодировке символ занимает 1 байт), включая обязательный стоп-символ => Кол-во введенных символов ∈ [1;996].

## Таблица трассировки

**Передаваемое сообщение**: «Барбисайзгерлз», кодировка КОИ-8:

В кодировке КОИ-8: E2 C1 D2 C2 C9 D3 C1 CA DA C7 C5 D2 CC DA

Трассировка выполнения команд только для двух символов в кодировке КОИ-8: **C1 E2**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Адр | Знчн | IP | CR | AR | DR | SP | BR | AC | NZVC | Адр\_1 | Знчн\_2 |
| 444 | 0200 | 445 | 0200 | 444 | 0200 | 0 | 0444 | 0000 | 0100 |  |  |
| 445 | 1203 | 446 | 1203 | 445 | 1203 | 0 | 0445 | 0040 | 0100 |  |  |
| 446 | 2F40 | 447 | 2F40 | 446 | 0040 | 0 | 0040 | 0040 | 0000 |  |  |
| 447 | F0FD | 448 | F0FD | 447 | F0FD | 0 | 0447 | 0040 | 0000 |  |  |
| 448 | AAF7 | 449 | AAF7 | 60D | E2C1 | 0 | FFF7 | E2C1 | 1000 | 440 | 060E |
| 449 | EEF9 | 44A | EEF9 | 443 | E2C1 | 0 | FFF9 | E2C1 | 1000 | 443 | E2C1 |
| 44A | 1302 | 44B | 1302 | 44A | 1302 | 0 | 044A | E2C1 | 1000 |  |  |
| 44B | 2EF6 | 44C | 2EF6 | 442 | 00FF | 0 | FFF6 | 00C1 | 0000 |  |  |
| 44C | 7EF4 | 44D | 7EF4 | 441 | 0000 | 0 | FFF4 | 00C1 | 0001 |  |  |
| 44D | F00C | 44E | F00C | 44D | F00C | 0 | 044D | 00C1 | 0001 |  |  |
| 44E | 0200 | 44F | 0200 | 44E | 0200 | 0 | 044E | 0000 | 0101 |  |  |
| 44F | 1203 | 450 | 1203 | 44F | 1203 | 0 | 044F | 0040 | 0101 |  |  |
| 450 | 2F40 | 451 | 2F40 | 450 | 0040 | 0 | 0040 | 0040 | 0001 |  |  |
| 451 | F0FD | 452 | F0FD | 451 | F0FD | 0 | 0451 | 0040 | 0001 |  |  |
| 452 | 3EF0 | 453 | 3EF0 | 443 | E2C1 | 0 | 1D3E | E2C1 | 1001 |  |  |
| 453 | 0680 | 454 | 0680 | 453 | 0680 | 0 | 0453 | C1E2 | 1001 |  |  |
| 454 | 1302 | 455 | 1302 | 454 | 1302 | 0 | 0454 | C1E2 | 1001 |  |  |
| 455 | 2EEC | 456 | 2EEC | 442 | 00FF | 0 | FFEC | 00E2 | 0001 |  |  |
| 456 | 7EEA | 457 | 7EEA | 441 | 0000 | 0 | FFEA | 00E2 | 0001 |  |  |
| 457 | F002 | 458 | F002 | 457 | F002 | 0 | 0457 | 00E2 | 0001 |  |  |
| 458 | 0200 | 459 | 0200 | 458 | 0200 | 0 | 0458 | 0000 | 0101 |  |  |
| 459 | CEEB | 445 | CEEB | 459 | 0445 | 0 | FFEB | 0000 | 0101 |  |  |
| 445 | 1203 | 446 | 1203 | 445 | 1203 | 0 | 0445 | 0040 | 0101 |  |  |
| 446 | 2F40 | 447 | 2F40 | 446 | 0040 | 0 | 0040 | 0040 | 0001 |  |  |
| 447 | F0FD | 448 | F0FD | 447 | F0FD | 0 | 0447 | 0040 | 0001 |  |  |
| 448 | AAF7 | 449 | AAF7 | 60E | 0000 | 0 | FFF7 | 0000 | 0101 | 440 | 060F |
| 449 | EEF9 | 44A | EEF9 | 443 | 0000 | 0 | FFF9 | 0000 | 0101 | 443 | 0000 |
| 44A | 1302 | 44B | 1302 | 44A | 1302 | 0 | 044A | 0000 | 0101 |  |  |
| 44B | 2EF6 | 44C | 2EF6 | 442 | 00FF | 0 | FFF6 | 0000 | 0101 |  |  |
| 44C | 7EF4 | 44D | 7EF4 | 441 | 0000 | 0 | FFF4 | 0000 | 0101 |  |  |
| 44D | F00C | 45A | F00C | 44D | F00C | 0 | 000C | 0000 | 0101 |  |  |
| 45A | 0200 | 45B | 0200 | 45A | 0200 | 0 | 045A | 0000 | 0101 |  |  |
| 45B | EEE7 | 45C | EEE7 | 443 | 0000 | 0 | FFE7 | 0000 | 0101 | 443 | 0000 |
| 45C | 0100 | 45D | 0100 | 45C | 0100 | 0 | 045C | 0000 | 0101 |  |  |
| 45D | 0000 | 45E | 0000 | 45D | 0000 | 0 | 045D | 0000 | 0101 |  |  |
| 45E | 0000 | 45F | 0000 | 45E | 0000 | 0 | 045E | 0000 | 0101 |  |  |
| 45F | 0000 | 460 | 0000 | 45F | 0000 | 0 | 045F | 0000 | 0101 |  |  |
| 460 | 0000 | 461 | 0000 | 460 | 0000 | 0 | 0460 | 0000 | 0101 |  |  |
| 461 | 0000 | 462 | 0000 | 461 | 0000 | 0 | 0461 | 0000 | 0101 |  |  |
| 462 | 0000 | 463 | 0000 | 462 | 0000 | 0 | 0462 | 0000 | 0101 |  |  |
| 463 | 0000 | 464 | 0000 | 463 | 0000 | 0 | 0463 | 0000 | 0101 |  |  |
| 464 | 0000 | 465 | 0000 | 464 | 0000 | 0 | 0464 | 0000 | 0101 |  |  |

## Вывод

Во время выполнения данной лабораторной работы я ознакомился с программой асинхронного обмена данных через внешние устройства с Базовой ЭВМ, со способами представления строк, командами ввода-вывода, а также общей организацией системы ввода-вывода в Базовой ЭВМ.