МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №4**

по дисциплине

«ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ»

Вариант № 19856

***Выполнил:***

Студент группы P3118

Рамеев Тимур

Ильгизович

***Преподаватель:***

Осипов Святослав

Владимирович

**Содержание**

[Задание 3](#_Toc135170357)

[Выполнение работы 4](#_Toc135170358)

[Определение текста программы 4](#_Toc135170359)

[Описание программы 6](#_Toc135170360)

[Трассировка 7](#_Toc135170361)

[Вывод 8](#_Toc135170362)

# 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 ввод данных с ВУ-3
2. Программа начинается с адреса 1E616. Размещаемая строка находится по адресу 56116.
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 0D (CR). Стоп символ является обычным символом строки и подчиняется тем же правилам расположения в памяти что и другие символы строки.

# Выполнение работы

## Определение текста программы

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды/Данные** | **Мнемоника** | **Комментарий** |
| 1E6 | 0200 | START: CLA | Очистка аккумулятора |
| 1E7 | 1207 | S1: IN 7 | Значение SR ВУ-3 вносится в аккумулятор в бит №6 |
| 1E8 | 2F40 | AND #0x40 | Логическое умножение аккумулятора и числа 40­­­­16 |
| 1E9 | F0FD | BEQ S1 | Если в аккумуляторе 0 переход к S1, иначе переход к следующей команде |
| 1EA | 1206 | IN 6 | Значение DR ВУ-3 вносится в аккумулятор в младшие 8 битов |
| 1EB | 6F0D | SUB #0x0D | Отнимаем из аккумулятора значение 0D16 |
| 1EC | F00D | BEQ STOP1 | Если в аккумуляторе 0 переход к STOP1, иначе переход к следующей команде |
| 1ED | 4F0D | ADD #0x0D | Логическое умножение аккумулятора и числа 0D16 |
| 1EE | EE14 | ST RES | Сохранение аккумулятора в ячейке RES |
| 1EF | 1207 | S2: IN 7 | Значение SR ВУ-3 вносится в аккумулятор в бит №6 |
| 1F0 | 2F40 | AND #0x40 | Логическое умножение аккумулятора и числа 40­­­­16 |
| 1F1 | F0FD | BEQ S2 | Если в аккумуляторе 0 переход к S2, иначе переход к следующей команде |
| 1F2 | 1206 | IN 6 | Значение DR ВУ-3 вносится в аккумулятор в младшие 8 битов |
| 1F3 | 6F0D | SUB #0x0D | Отнимаем из аккумулятора значение 0D16 |
| 1F4 | F008 | BEQ STOP2 | Если в аккумуляторе 0 переход к STOP2, иначе переход к следующей команде |
| 1F5 | 4F0D | ADD #0x0D | Логическое умножение аккумулятора и числа 0D16 |
| 1F6 | 0680 | SWAB | Младшие 8 бит аккумулятора меняются местами со старшими 8 битами |
| 1F7 | 4E0B | ADD RES | Добавляем значение ячейки RES к аккумулятору |
| 1F8 | EA09 | ST (ADR)+ | Сохраняем аккумулятор по адресу, находящемуся в ячейке ADR. Прибавляем 1 к адресу, находящемуся в ячейке ADR |
| 1F9 | CEEC | JUMP START | Переходим к ячейке START |
| 1FA | 4F0D | STOP1: ADD #0x0D | Добавляем значение 0x0D к аккумулятору |
| 1FB | E806 | ST (ADR) | Сохраняем аккумулятор по адресу, находящемуся в ячейке ADR |
| 1FC | 0100 | HLT | Завершение программы |
| 1FD | 4F0D | STOP2: ADD #0x0D | Добавляем значение 0x0D к аккумулятору |
| 1FE | 0680 | SWAB | Младшие 8 бит аккумулятора меняются местами со старшими 8 битами |
| 1FF | 4E03 | ADD RES | Добавляем значение ячейки RES к аккумулятору |
| 200 | E801 | ST (ADR) | Сохраняем аккумулятор по адресу, находящемуся в ячейке ADR |
| 201 | 0100 | HLT | Завершаем программу |
| 202 | 0561 | ADR: WORD 0x561 | Ячейка, содержащая адрес начала строки |
| 203 | ? | RES: WORD ? | Ячейка, содержащая промежуточный результат  (нечетные символы строки) |

## Описание программы

1. Исходные данные:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес в памяти | Назначение, описание | Присвоенное обозначение | Область представления | Область допустимых значений |
| 202 | Адрес начала строки | ADR = 561  (по условию) | Адрес памяти  БЭВМ | 0x000 ≤ ADR ≤ 0x7FF |
| 203 | Промежуточный результат, содержащий один нечетный символ | RES | - | 0x0000 ≤ RES ≤ 0x00FF |
| 567 - … | Результат (вводимая строка) | STR | - | 0x0000 ≤ STR ≤ 0xFFFF |

1. Назначение программы и функция:

программа осуществляет ввод или вывод информации, используя в качестве подтверждения данных сигнал готовности ВУ–3.

1. Адрес первой исполняемой команды – 1E6, адрес последней исполняемой команды – 201.

## Трассировка

СЛОВО -

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Выполняемая команда** | | **Содержимое регистров после выполнения программы** | | | | | | | | | **Изменения в памяти** | |
| Адрес | Значение | IP | CR | AR | DR | SP | BR | AC | PS | NZVC | Адр | Знчн |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

# Вывод

При выполнении лабораторной работы мною был изучен асинхронный обмен с устройствами ввода-вывода в БЭВМ. Я написал программу для записи символов из устройства в память.