Федеральное государственное автономное образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский

**университет информационных технологий, механики и оптики»**

**Факультет: Программной инженерии**

**Дисциплина: Основы Профессиональной Деятельности**

**Лабораторная работа №5**

**Вариант 14145.**

**Выполнил: Нуруллаев Даниил**

**Группа: Р3114**

**Преподаватель:**

**Блохина Елена Николаевна**

**Г.Санкт-Петербург**

**2021г.**

**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-3
2. Программа начинается с адреса 23F16. Размещаемая строка находится по адресу 59016.
3. Строка должна быть представлена в кодировке КОИ-8.
4. Формат представления строки в памяти: АДР1: СИМВ1 СИМВ2 АДР2: СИМВ3 СИМВ4 ... 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.

**Код исходной программы:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарий** |
| 23F | +0200 | CLA | Очищает аккумулятор |
| 240 | AAFB | LD (IP-4)+ | Загружаем адрес элемента массива |
| 241 | EEFC | ST IP-3 | Сохраняет адрес элемента массива |
| 242 | 1207 | IN 7 | Подаем REG в AC из ВУ-3 |
| 243 | 2F40 | AND #40 | Проводим логическое умножение AC и 40 |
| 244 | F0FD | BEQ FD | Проверяем если результат равен 40 то продолжаем, иначе возвращаемся в ячейку по адресу 242 |
| 245 | AEF8 | LD IP-7 | Загружаем значение из массива |
| 246 | 0680 | SWAB | Меняем старшие байты с младшими |
| 247 | 7EF5 | CMP IP-A | Выставляем флаги по AC-Стоп Слово |
| 248 | F009 | BEQ 09 | Если Z-флаг равен единице выходим из программы |
| 249 | 1306 | OUT 6 | Выводим букву в ВУ-3 |
| 24A | 1207 | IN 7 | Подаем REG в AC из ВУ-3 |
| 24B | 2F40 | AND #40 | Проводим логическое умножение AC и 40 |
| 24C | F0FD | BEQ FD | Проверяем если результат равен 40 то продолжаем, иначе возвращаемся в ячейку по адресу 24C |
| 24D | AEF0 | LD IP-F | Загружаем значение из массива |
| 24E | 7EEE | CMP IP-11 | Выставляем флаги по AC-Стоп Слово |
| 24F | F002 | BEQ 02 | Если Z-флаг равен единице выходим из программы |
| 250 | 1306 | OUT 6 | Выводим букву в ВУ-3 |
| 251 | CEED | JUMP IP-12 | Переходи в начало программы |
| 252 | 0100 | HLT | Остановка программы |

**Текст исходной программы:**

ORG 0x23C

ADRESS: WORD $STRING

NL: WORD 0xD8

NOW: WORD 0

START: CLA

LD (ADRESS)+

ST NOW

POINT1: IN 0x11

AND #0x40

BEQ POINT1

LD NOW

SWAB

CMP NL

BEQ STOPPOINT

OUT 0x10

POINT2: IN 0x11

AND #0x40

BEQ POINT2

LD NOW

CMP NL

BEQ STOPPOINT

OUT 0x10

JUMP START

STOPPOINT: OUT 0x10

HLT

ORG 0x590

STRING: WORD 0xEDC1

WORD 0xCA2E

WORD 0x20E1

WORD 0xD0D2

WORD 0xC5CC

WORD 0xD800

**Описание программы:**

Программа реализует посимвольный асинхронный вывод в ВУ-3 в кодировке КОИ-8-R. В 16 битной ячейке памяти БЭВМ размещаются два 8-битных символа, в старшем байте символ с меньшим индексом. Строка начинается с ячейки 0x590. Цикл вывода программы продолжается до тех пор, пока не дойдет до символа NL (0x00)

**Область представления и область допустимых значений исходных данных и результата:**

**Область представления:**

Ячейки NL,NOW:16-разрядные беззнаковые числа

Ячейки с строкой: 16-разрядные беззнаковые числа

**Область допустимых значений:**

**NL-Const=0x00**

Длина строки: от 0 до 2393 символов

**Расположение в памяти ЭВМ программы, исходных данных и результатов:**

**Программа:**

Адрес ячейки первого символа строки: 23C (ADDRESS)

Код символа окончания строки: 23D (NL)

Адрес текущей ячейки записи символов: 23E (NOW)

Введенная строка: 590…590 + (𝑁16+1)/2(без остатка)

(где N – длина строки в 16-ричной СС)

**Адреса первой и последней выполняемой команд программы:**

Адрес первой команды программы: 23F

Адрес последней команды программы: 252

**Закодированная строка в KOI8-R в UTF8 и UTF 16:**

KOI8-R: ed c1 ca 2e 20 e1 d0 d2 c5 cc d8

UTF8: d09c d0b0 d0b9 2e 20 d090 d0bf d180 d0b5 d0bb d18c

UTF16: 041c 0430 0439 002e 0020 0410 043f 0440 0435 043b 044c

**Вывод:**

В ходе выполнения данной лабораторной работы я познакомился с ВУ. Узнал о новых командах IN и OUT. Начал изучение ассемблера, который позволяет гораздо проще ввести программу.

**Трассировка:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Адрес** | **Код** | **IP** | **CR** | **AR** | **DR** | **SP** | **BR** | **AC** | **NZVC** | **Адрес** | **Знач** |
| 23F | 0200 | 240 | 0200 | 23F | 0200 | 000 | 023F | 0000 | 0100 |  |  |
| 240 | AAFB | 241 | AAFB | 590 | EDC1 | 000 | FFFB | EDC1 | 1000 | 23C | 591 |
| 241 | EEFC | 242 | EEFC | 23E | EDC1 | 000 | FFFC | EDC1 | 1000 | 23E | EDC1 |
| 242 | 1211 | 243 | 1211 | 242 | 1211 | 000 | 0242 | ED40 | 1000 |  |  |
| 243 | 2F40 | 244 | 2F40 | 243 | 0040 | 000 | 0040 | 0040 | 0000 |  |  |
| 244 | F0FD | 245 | F0FD | 244 | F0FD | 000 | 0244 | 0040 | 0000 |  |  |
| 245 | AEF8 | 246 | AEF8 | 23E | EDC1 | 000 | FFF8 | EDC1 | 1000 |  |  |
| 246 | 0680 | 247 | 0680 | 246 | 0680 | 000 | 0246 | C1ED | 1000 |  |  |
| 247 | 7EF5 | 248 | 7EF5 | 23D | 00D8 | 000 | FFF5 | C1ED | 1001 |  |  |
| 248 | F009 | 249 | F009 | 248 | F009 | 000 | 0248 | C1ED | 1001 |  |  |
| 249 | 1310 | 24A | 1310 | 249 | 1310 | 000 | 0249 | C1ED | 1001 |  |  |
| 24A | 1211 | 24B | 1211 | 24A | 1211 | 000 | 024A | C140 | 1001 |  |  |
| 24B | 2F40 | 24C | 2F40 | 24B | 0040 | 000 | 0040 | 0040 | 0001 |  |  |
| 24C | F0FD | 24D | F0FD | 24C | F0FD | 000 | 024C | 0040 | 0001 |  |  |
| 24D | AEF0 | 24E | AEF0 | 23E | EDC1 | 000 | FFF0 | EDC1 | 1001 |  |  |
| 24E | 7EEE | 24F | 7EEE | 23D | 00D8 | 000 | FFEE | EDC1 | 1001 |  |  |
| 24F | F002 | 250 | F002 | 24F | F002 | 000 | 024F | EDC1 | 1001 |  |  |
| 250 | 1310 | 251 | 1310 | 250 | 1310 | 000 | 0250 | EDC1 | 1001 |  |  |
| 251 | CEED | 23F | CEED | 251 | 023F | 000 | FFED | EDC1 | 1001 |  |  |