

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 №5

по дисциплине

«Основы профессиональной деятельности»

Асинхронный обмен данными с ВУ

Вариант № 14228

Выполнил:

Студент группы Р3107

Валиев Руслан Новруз оглы

Приняла:

Остапенко Ольга Денисовна

Содержание

| | |
|--------------------------|---|
| Задание | 3 |
| Описание программы | 4 |
| Текст исходной программы | 5 |
| Описание программы | 6 |
| Таблица трассировки | 7 |
| Заключение | 9 |

Задание

Лабораторная работа №5

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.

Введите номер варианта

1. Программа осуществляет асинхронный ввод данных с ВУ-3
2. Программа начинается с адреса 068_{16} . Размещаемая строка находится по адресу $5B9_{16}$.
3. Строка должна быть представлена в кодировке ISO-8859-5.
4. Формат представления строки в памяти: АДР1: СИМВ2 СИМВ1 АДР2: СИМВ4 СИМВ3 ... СТОП_СИМВ.
5. Ввод или вывод строки должен быть завершен по символу с кодом 0D (CR). Стоп символ является обычным символом строки и подчиняется тем же правилам расположения в памяти что и другие символы строки.

Описание программы

```
1  ORG 0x010
2  RES_ADDRESS: WORD $RES          ; current string address
3  EOF: WORD 0x0d                  ; EOF char
4  TMP: WORD ?                     ; tmp
5
6  ORG 0x068
7  START:      CLA                  ; clear
8
9  √ FIRST_SYMBOL: IN 7              ; \
10              AND #0x40           ; -> spin-loop while not ready
11              BEQ FIRST_SYMBOL    ; /
12
13              IN 6                 ; \
14              SWAB                 ; -> read char, save to first byte of the cell
15              ST (RES_ADDRESS)    ; /
16
17              SWAB                 ; \
18              CMP EOF              ; -> if eof then stop
19              BEQ STOP            ; /
20
21 √ SECOND_SYMBOL: IN 7             ; \
22              AND #0x40           ; -> spin-loop while not ready
23              BEQ SECOND_SYMBOL    ; /
24
25              IN 6                 ; \
26              ST TMP               ; | -> read char, save to second byte of the cell
27              ADD (RES_ADDRESS)    ; |
28              ST (RES_ADDRESS)+    ; /
29
30              LD TMP               ; \
31              CMP EOF              ; -> if eof then stop
32              BEQ STOP            ; /
33
34              JUMP FIRST_SYMBOL    ; while (true)
35
36 STOP:      HLT                    ; exit programm
37
38 ORG 0x5B9
39 RES: WORD ?                       ; string beginning
```

Текст исходной программы

| Адрес | Код команды | Мнемоника | Комментарии |
|-------|----------------|-------------------|--|
| 010 | - | RES_ADDRESS | Адрес текущей ячейки |
| 011 | - | EOF | Символ конца ввода |
| 012 | - | TMP | переменная для хранения временных данных |
| 528 | 0200 | CLA | Очистить аккумулятор |
| 529 | 1207 | IN #7 | Ожидание ввода |
| 52A | 2F40 | AND #0x40 | Проверка статуса |
| 52B | F0FD | BEQ IP-3 | Спин-луп в ожидании готовности |
| 52C | 1206 | IN #6 | Считывание символа |
| 52D | 0680 | SWAB | Меняем старший и младший байты |
| 52E | E8A1 | ST (RES_ADDRESS) | Сохраняем в текущей ячейке |
| 52F | 0680 | SWAB | Меняем обратно для проверки EOF |
| 530 | 7EA0 | CMP EOF | Проверяем на наличие EOF |
| 531 | F00B | BEQ IP+11 | Если EOF, то завершаем |
| 532 | 1207 | IN #7 | Ожидание ввода |
| 533 | 2F40 | AND #0x40 | Проверка статуса |
| 534 | F0FD | BEQ IP-3 | Спин-луп в ожидании готовности |
| 535 | 1206 | IN #6 | Считывание символа |
| 536 | EE9B | ST TMP | Сохраним во временной переменной |
| 537 | 4898 | ADD (RES_ADDRESS) | Добавим значение предыдущего символа |
| 538 | EA97 | ST (RES_ADDRESS)+ | Сохраним 2 символа в текущей ячейке |

| | | | |
|-----|------|------------|--|
| 539 | AE98 | LD TMP | Загружаем в аккумулятор временную переменную |
| 53A | 7E96 | CMP EOF | Проверяем на наличие EOF |
| 53B | F001 | BEQ IP+1 | Если EOF, то завершаем |
| 53C | CEEC | JUMP IP-20 | Прыжок на ячейку 069 |
| 53D | 0100 | HLT | Завершение программы |
| 9 | - | RES | Строка результата |

10100001

01011111

Описание программы

Программа осуществляет посимвольный асинхронный ввод данных с ВУ-3. Программа будет получать символы до тех пор, пока на ВУ-3 не будет введен стоп-символ (EOF) с кодировкой 0x0D, который она запишет в память и прекратит свое выполнение.

Область представления:

RES -? - 16-разрядные ячейки, хранящие в себе по два символа в кодировке Windows-1251
 RES_ADDRESS - 11 разрядная ячейка, хранящая адрес текущей ячейки
 EOF - 16-разрядная константа.

TMP - 16-разрядная ячейка для временного хранения введенных символов.

Область определения:

- RES_ADDRESS (указатель на ячейки массива, хранящий результат ввода) ∈ [5B9;2047]
- EOF: [00; FF]

Таблица трассировки

Строка для трассировки: «тетрагидроканнабинол».

ISO-8859-5: E2 D5 E2 E0 D0 D3 D8 D4 E0 DE DA D0 DD DD D0 D1 D8 DD DE DB

UTF-8: D1 82 D0 B5 D1 82 D1 80 D0 B0 D0 B3 D0 B8 D0 B4 D1 80 D0 BE D0 BA D0 B0 D0 BD D0 BD D0 B0 D0 B1 D0 B8 D0 BD D0 BE D0 BB

UTF-16: FF FE 42 04 35 04 42 04 40 04 30 04 33 04 38 04 34 04 40 04 3E 04 3A 04 30 04 3D 04 3D 04 30 04 31 04 38 04 3D 04 3E 04 3B 04

| 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 |
| 528 | 0200 | 529 | 0200 | 528 | 0200 | 000 | 0068 | 0000 | 004 | 0100 | | |
| 529 | 1207 | 52A | 1207 | 529 | 1207 | 000 | 0069 | 0040 | 004 | 0100 | | |
| 52A | 2F40 | 52B | 2F40 | 52A | 0040 | 000 | 0040 | 0040 | 000 | 0000 | | |
| 52B | F0FD | 52C | F0FD | 52B | F0FD | 000 | 006B | 0040 | 000 | 0000 | | |
| 52C | 1206 | 52D | 1206 | 52C | 1206 | 000 | 006C | 00E2 | 000 | 0000 | | |
| 52D | 0680 | 52E | 0680 | 52D | 0680 | 000 | 006D | E200 | 008 | 1000 | | |
| 52E | E8E1 | 52F | E8A1 | 5A3 | E200 | 000 | FFA1 | E200 | 008 | 1000 | 5A3 | E200 |
| 52F | 0680 | 530 | 0680 | 52F | 0680 | 000 | 006F | 00E2 | 000 | 0000 | | |
| 530 | 7EE0 | 531 | 7EA0 | 511 | 000D | 000 | FFA0 | 00E2 | 001 | 0001 | | |
| 531 | F00B | 532 | F00B | 531 | F00B | 000 | 0071 | 00E2 | 001 | 0001 | | |
| 532 | 1207 | 533 | 1207 | 532 | 1207 | 000 | 0072 | 0040 | 005 | 0101 | | |
| 533 | 2F40 | 534 | 2F40 | 533 | 0040 | 000 | 0040 | 0040 | 001 | 0001 | | |

| | | | | | | | | | | | | |
|-----|------|-----|------|-----|------|-----|------|------|-----|------|-------------|---------------|
| 534 | F0FD | 535 | F0FD | 534 | F0FD | 000 | 0074 | 0040 | 001 | 0001 | | |
| 535 | 1206 | 536 | 1206 | 535 | 1206 | 000 | 0075 | 00D5 | 001 | 0001 | | |
| 536 | EEDB | 537 | EE9B | 512 | 00D5 | 000 | FF9B | 00D5 | 001 | 0001 | 512 | 00D5 |
| 537 | 48D8 | 538 | 4898 | 5A3 | E200 | 000 | FF98 | E2D5 | 008 | 1000 | | |
| 538 | EAD7 | 539 | EA97 | 5A3 | E2D5 | 000 | FF97 | E2D5 | 008 | 1000 | 510, 5A3 | 05A4, E2D5 |
| 539 | AED8 | 53A | AE98 | 512 | 00D5 | 000 | FF98 | 00D5 | 000 | 0000 | | |
| 53A | 7ED6 | 53B | 7E96 | 511 | 000D | 000 | FF96 | 00D5 | 001 | 0001 | | |
| 53B | F001 | 53C | F001 | 53B | F001 | 000 | 007B | 00D5 | 001 | 0001 | | |
| 53C | CEEC | 529 | CEEC | 53C | 0069 | 000 | FFEC | 00D5 | 001 | 0001 | | |
| 529 | 1207 | 52A | 1207 | 529 | 1207 | 000 | 0069 | 0000 | 001 | 0001 | | |
| 52A | 2F40 | 52B | 2F40 | 52A | 0040 | 000 | 0040 | 0000 | 005 | 0101 | | |
| 52B | F0FD | 529 | F0FD | 52B | F0FD | 000 | FFFD | 0000 | 005 | 0101 | | |
| 529 | 1207 | 52A | 1207 | 529 | 1207 | 000 | 0069 | 0000 | 005 | 0101 | | |
| 52A | 2F40 | 52B | 2F40 | 52A | 0040 | 000 | 0040 | 0000 | 005 | 0101 | | |
| 52B | F0FD | 529 | F0FD | 52B | F0FD | 000 | FFFD | 0000 | 005 | 0101 | | |
| 529 | 1207 | 52A | 1207 | 529 | 1207 | 000 | 0069 | 0000 | 005 | 0101 | | |
| 52A | 2F40 | 52B | 2F40 | 52A | 0040 | 000 | 0040 | 0000 | 005 | 0101 | | |
| 52B | F0FD | 529 | F0FD | 52B | F0FD | 000 | FFFD | 0000 | 005 | 0101 | | |
| 529 | 1207 | 52A | 1207 | 529 | 1207 | 000 | 0069 | 0040 | 005 | 0101 | | |
| 52A | 2F40 | 52B | 2F40 | 52A | 0040 | 000 | 0040 | 0040 | 001 | 0001 | | |
| 52B | F0FD | 52C | F0FD | 52B | F0FD | 000 | 006B | 0040 | 001 | 0001 | | |
| 52C | 1206 | 52D | 1206 | 52C | 1206 | 000 | 006C | 000D | 001 | 0001 | | |
| 52D | 0680 | 52E | 0680 | 52D | 0680 | 000 | 006D | 0D00 | 001 | 0001 | | |
| 52E | E8E1 | 52F | E8A1 | 5A4 | 0D00 | 000 | FFA1 | 0D00 | 001 | 0001 | 5A4 | 0D00 |
| 52F | 0680 | 530 | 0680 | 52F | 0680 | 000 | 006F | 000D | 001 | 0001 | | |

| | | | | | | | | | | | | |
|-----|------|-----|------|-----|------|-----|------|------|-----|------|--|--|
| 530 | 7EE0 | 531 | 7EA0 | 011 | 000D | 000 | FFA0 | 000D | 005 | 0101 | | |
| 531 | F00B | 53D | F00B | 531 | F00B | 000 | 000B | 000D | 005 | 0101 | | |
| 53D | 0100 | 53E | 0100 | 53D | 0100 | 000 | 007D | 000D | 005 | 0101 | | |

Заключение

В ходе проделанной лабораторной работы, я познакомился 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 написания кода на Ассемблере БЭВМ.