Федеральное государственное автономное образовательное учреждение высшего образования

**Университет ИТМО**

**Факультет программной инженерии и компьютерной техники**

**Дисциплина: Основы профессиональной деятельности**

**Лабораторная работа №5**

**«Исследование работы БЭВМ»**

Вариант 967

**Выполнил:** Кривоносов Егор Дмитриевич

**Группа:** Р3111

**Преподаватель:** Николаев Владимир Вячеславович

Санкт-Петербург, 2020г

**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 41016. Размещаемая строка находится по адресу 61B16.
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 должна быть завершен по символу c кодом 0D (CR)

**Выполнение работы:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Адрес** | **Код команды** | **Мнемоника** | **Комментарий** |
| 1 | 410 | 061B | ADDR | Адрес начала строки |
| 2 | 411 | 0000 | I | Хранит адрес текущей ячейки с символами |
| 3 | 412 | 0005 | M | Количество символов в строке (длина) |
| 4 | 413 | 0000 | BUFF | Служебная ячейка для хранения 2-х символов |
| 5 | 414 | 0200 | CLA | Очистка аккумулятора |
| 6 | 415 | AEFA | LD (IP-6) | Загрузка в AC переменной ADDR (ячейка 410) |
| 7 | 416 | EEFA | ST (IP-6) | Сохранение в ячейку 411 |
| 8 | 417 | AEFA | LD (IP-6) | Загрузка в AC переменной M (ячейка 412) |
| 9 | 418 | 0700 | INC | Инкремент AC+1 -> AC |
| 10 | 419 | EEF8 | ST (IP-8) | Сохранение в ячейку 412 |
| 11 | 41A | 1203 | IN 3 | Ожидание готовности 1 символа (SR (#3) -> AC) |
| 12 | 41B | 2F40 | AND #40 | Бит 6 SR == 0 (“Готов” нажата?) (M & AC -> AC) |
| 13 | 41C | F0FD | BEQ (IP-3) | Нет – “Спин-луп” (IF Z == 1 THEN IP-3+1 -> IP) |
| 14 | 41D | AAF3 | LD (IP-13) + | Загрузка в AC значение по адресу ячейки 411, к значению ячейки 411 плюс 1 |
| 15 | 41E | EEF4 | ST (IP-12) | Сохранение в ячейку 413 |
| 16 | 41F | 0600 | SXTB | Расширение знака байта |
| 17 | 420 | 8EF1 | LOOP M | Декремент и пропуск если М - 1 -> M; если М <0 |
| 18 | 421 | 7F0D | CMP #0D | Установка флагов по результату AC – 000D |
| 19 | 422 | F00D | BEQ (IP+13) | IF Z == 1 THEN IP+13+1 -> IP |
| 20 | 423 | 1302 | OUT 2 | Вывод 1 символа (младшие 8 разрядов) в ВУ-1 |
| 21 | 424 | 1203 | IN 3 | Ожидание готовности 1 символа (SR (#3) -> AC) |
| 22 | 425 | 2F40 | AND #40 | Бит 6 SR == 0 (“Готов” нажата?) (M & AC -> AC) |
| 23 | 426 | F0FD | BEQ (IP-3) | Нет – “Спин-луп” (IF Z == 1 THEN IP-3+1 -> IP) |
| 24 | 427 | AEEB | LD (IP-21) | Загрузка в AC значение по адресу ячейки 413 |
| 25 | 428 | 0680 | SWAB | Обмен старших и младших байтов |
| 26 | 429 | 0600 | SXTB | Расширение знака байта |
| 27 | 42A | 7F0D | CMP #0D | Установка флагов по результату AC – 000D |
| 28 | 42B | F003 | BEQ (IP+3) | IF Z == 1 THEN IP+3+1 -> IP |
| 29 | 42C | 1302 | OUT 2 | Вывод 2 символа (младшие 8 разрядов) в ВУ-1 |
| 30 | 42D | 8EE4 | LOOP M | Декремент и пропуск если М - 1 -> M; если М <0 |
| 31 | 42E | CEEB | JUMP (IP-21) | Переход в ячейку 41A |
| 32 | 42F | 0100 | HLT | Остановка |
| 33 | 430 | CEF3 | JUMP (IP-13) | Переход в ячейку 424 |
| Массив символов строки: | | | | |
| 34 | 61B |  | СИМВ2 СИМВ1 | |
| 35 | 61C |  | СИМВ4 СИМВ3 | |
| 36 | 61D |  | СИМВ6 СИМВ5 | |
| … | … | … | … | |
| … | … | 000D | СТОП\_СИМВ | |

**Текст программы на языке Ассемблера БЭВМ:**

ORG 0x410  
ADDR: WORD $ARRAY  
I: WORD 0  
M: WORD 5  
BUFF: WORD 0  
START: CLA  
 LD ADDR  
 ST I  
 LD M  
 INC  
 ST M

S1: IN 3  
AND #0x40  
BEQ S1  
LD (I)+  
ST BUFF  
SXTB  
LOOP M  
CMP #0x0D  
BEQ HELP  
OUT 2

S2: IN 3  
AND #0x40  
BEQ S2  
LD BUFF  
SWAB  
SXTB  
CMP #0x0D  
BEQ STOP  
OUT 2

NEXT: LOOP M  
 BR S1

STOP: HLT  
HELP: BR S2

ORG 0x61B  
ARRAY: ?

WORD 0x0D

**Описание программы:**

1. **Назначение программы**:  
   Асинхронный вывод данных на ВУ-1. Программа получает строку в кодировке ISO-8859-5. Полученная строка посимвольно выводится на ВУ-1. Признаком окончания вывода является символ 0D (CR).

Строка представления в кодировке ISO-8859-5, формат представления в памяти: АДР1: СИМВ2 СИМВ1, АДР2: СИМВ4 СИМВ3 … СТОП\_СИМВ.

1. **Область представления и ОДЗ**:

**Область представления:**

* СИМВ – 8-ми битное, целое, беззнаковое число.
* I – 11-ти битное, целое, беззнаковое число.

**Область допустимых значений:**

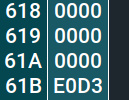
* СИМВ: [0, 28-1]

1. **Расположение данных и программы в памяти ЭВМ**:  
   Переменные: 410-413,  
   Программа: 414-430,   
   Адрес начала строки: 61B
2. **Адреса первой и последней выполняемых команд программы**:   
   Адрес первой выполняемой команды - 414,   
   Адрес последней выполняемой команды - 429.

**Исходная строка:**греча

|  |  |  |  |
| --- | --- | --- | --- |
| Символ | ISO-8859-5 | UTF-8 | UTF-16BE |
| г | D3 | D0 B3 | 04 33 |
| р | E0 | D1 80 | 04 40 |
| е | D5 | D0 B5 | 04 35 |
| ч | E7 | D1 87 | 04 47 |
| а | D0 | D0 B0 | 04 30 |

**Первые 2 символа слова после трассировки в памяти БЭВМ:**

****

**Трассировка программы:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Выполняемая команда** | | **Содержимое регистров процессора после выполнения команды** | | | | | | | | **Ячейка, содержимое которой изменилось после выполнения команды** | |
| Адрес | Код | IP | CR | AR | DR | SP | BR | AC | NZVC | Адрес | Новый код |
| 414 | 0200 | 415 | 0200 | 414 | 0200 | 000 | 0414 | 0000 | 0100 |  |  |
| 415 | AEFA | 416 | AEFA | 410 | 061B | 000 | FFFA | 061B | 0000 |  |  |
| 416 | EEFA | 417 | EEFA | 411 | 061B | 000 | FFFA | 061B | 0000 | 411 | 061B |
| 417 | AEFA | 418 | AEFA | 412 | 0005 | 000 | FFFA | 0005 | 0000 |  |  |
| 418 | 0700 | 419 | 0700 | 418 | 0700 | 000 | 0418 | 0006 | 0000 |  |  |
| 419 | EEF8 | 41A | EEF8 | 412 | 0006 | 000 | FFF8 | 0006 | 0000 | 412 | 0006 |
| 41A | 1203 | 41B | 1203 | 41A | 1203 | 000 | 041A | 0040 | 0000 |  |  |
| 41B | 2F40 | 41C | 2F40 | 41B | 0040 | 000 | 0040 | 0040 | 0000 |  |  |
| 41C | F0FD | 41D | F0FD | 41C | F0FD | 000 | 041C | 0040 | 0000 |  |  |
| 41D | AAF3 | 41E | AAF3 | 61B | E0D3 | 000 | FFF3 | E0D3 | 1000 |  |  |
| 41E | EEF4 | 41F | EEF4 | 413 | E0D3 | 000 | FFF4 | E0D3 | 1000 | 413 | E0D3 |
| 41F | 0600 | 420 | 0600 | 41F | 0600 | 000 | 041F | FFD3 | 1000 |  |  |
| 420 | 8EF1 | 421 | 8EF1 | 412 | 0005 | 000 | 0004 | FFD3 | 1000 | 412 | 0005 |
| 421 | 7F0D | 422 | 7F0D | 421 | 000D | 000 | 000D | FFD3 | 1001 |  |  |
| 422 | F00D | 423 | F00D | 422 | F00D | 000 | 0422 | FFD3 | 1001 |  |  |
| 423 | 1302 | 424 | 1302 | 423 | 1302 | 000 | 0423 | FFD3 | 1001 | ВУ-1 | D3 |
| 424 | 1203 | 425 | 1203 | 424 | 1203 | 000 | 0424 | FF40 | 1001 |  |  |
| 425 | 2F40 | 426 | 2F40 | 425 | 0040 | 000 | 0040 | 0040 | 0001 |  |  |
| 426 | F0FD | 427 | F0FD | 426 | F0FD | 000 | 0426 | 0040 | 0001 |  |  |
| 427 | AEEB | 428 | AEEB | 413 | E0D3 | 000 | FFEB | E0D3 | 1001 |  |  |
| 428 | 0680 | 429 | 0680 | 428 | 0680 | 000 | 0428 | D3E0 | 1001 |  |  |
| 429 | 0600 | 42A | 0600 | 429 | 0600 | 000 | 0429 | FFE0 | 1001 |  |  |
| 42A | 7F0D | 42B | 7F0D | 42A | 000D | 000 | 000D | FFE0 | 1001 |  |  |
| 42B | F003 | 42C | F003 | 42B | F003 | 000 | 042B | FFE0 | 1001 |  |  |
| 42C | 1302 | 42D | 1302 | 42C | 1302 | 000 | 042C | FFE0 | 1001 | ВУ-1 | E0 |
| 42D | 8EE4 | 42E | 8EE4 | 412 | 0004 | 000 | 0003 | FFE0 | 1001 | 412 | 0004 |
| 42E | CEEB | 41A | CEEB | 42E | 041A | 000 | FFEB | FFE0 | 1001 |  |  |

**Вывод:**

Во время выполнения данной лабораторной работы я изучил работу БЭВМ с ВУ в асинхронном режиме.