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**

“Асинхронный обмен данными с ВУ”

Вариант № 52308

Выполнил:

Сандов Кирилл Алекссевич

Группа:

P3113

Проверила:

преподаватель Блохина Елена Николаевна

Санкт-Петербург

2023

# 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.

* Программа осуществляет асинхронный ввод данных с ВУ-3.
* Программа начинается с адреса . Размещаемая строка находится по адресу .
* Строка должна быть представлена в кодировке КОИ-8.
* Формат представления строки в памяти: АДР1: СИМВ2 СИМВ1 АДР2: СИМВ4 СИМВ3 ... СТОП\_СИМВ.
* 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.
* Заданное слово: Сметана-.

Последовательность шестнадцатеричных кодов символов слова в кодировке КОИ-8:

F3, CD, C5, D4, C1, CE, C1, 2D

В кодировке UTF-8:

D0A1, D0BC, D0B5, D182, D0B0, D0BD, D0B0, 2D

В кодировке UTF-16:

0421, 043C, 0435, 0442, 0430, 043D, 0430, 002D

# Пункт 1

Запишем разработанную программу в виде таблицы (Таблица 1).

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарии** |
| 244 | 0200 | CLA | Обнулить аккумулятор |
| 245 | 1207 | IN 0x07 | Считать данные с регистра по адресу 0x07 устройства ВУ-3 (будет считан SR устройства) |
| 246 | 2F40 | AND #0x40 | Выполнить операцию побитового И аккумулятора и значения 0x40 и записать результат в аккумулятор |
| 247 | F0FD | BEQ IP-3 | Переход в ячейку 247+1-3=245, если Z = 1 (то есть на предыдущем шаге 6-ой бит не равен 1, значит, сигнал «Готов» не установлен) |
| 248 | 1206 | IN 0x06 | Считать данные с регистра по адресу 0x06 устройства ВУ-3 (будет считан DR устройства) |
| 249 | 7F0D | CMP #0x0D | Установить признаки NZVC по операции разности аккумулятора и значения 0x0D |
| 24A | F00B | BEQ IP+1 | Переход в ячейку 24A+1+B=256, если Z = 1 (с учётом предыдущего шага, полученное значение из ВУ-3 равно 0x0D) |
| 24B | EE24 | ST (IP+24) | Сохранить значение аккумулятора в ячейку 24B+1+24=270 |
| 24C | 0200 | CLA | Обнулить аккумулятор |
| 24D | 1207 | IN 0x07 | Считать данные с регистра по адресу 0x07 устройства ВУ-3 (будет считан SR устройства) |
| 24E | 2F40 | AND #0x40 | Выполнить операцию побитового И аккумулятора и значения 0x40 и записать результат в аккумулятор |
| 24F | F0FD | BEQ IP-3 | Переход в ячейку 247+1-3=245, если Z = 1 (то есть на предыдущем шаге 6-ой бит не равен 1, значит, сигнал «Готов» не установлен) |
| 250 | 1206 | IN 0x06 | Считать данные с регистра по адресу 0x06 устройства ВУ-3 (будет считан DR устройства) |
| 251 | 7F0D | CMP #0x0D | Установить признаки NZVC по операции разности аккумулятора и значения 0x0D |
| 252 | 0D00 | PUSHF | Сохранить PS в стек (нужно для сохранения признаков NZVC) |
| 253 | 0680 | SWAB | Обменять старший и младший байты аккумулятора местами |
| 254 | 4E1B | ADD (IP+1B) | Прибавить к аккумулятору значение ячейки 254+1+1B=270 |
| 255 | 0900 | POPF | Загрузить значение с вершины стека в PS, уменьшив стек (вернуть сохранённые признаки NZVC) |
| 256 | EA49 | ST (IP+49)+ | Сохранить значение аккумулятора в ячейку, адрес которой лежит в ячейке 256+1+49=2A0, увеличив после значение этого адреса на 1 |
| 257 | F001 | BEQ IP+1 | Переход в ячейку 257+1+1=259, если Z = 1 (был считано значение 0x0D) |
| 258 | CEEC | JUMP (IP-13) | Переход в ячейку 259-1-13=245 |
| 259 | 0100 | HLT | Останов |

Таблица 1

Также запишем эту же программу на языке Ассемблера БЭВМ (Листинг 1).

ORG 0x244

**START**: CLA

S1: **IN** 0x007 *; Спин-луп для чтения состояния готовности с ВУ-3*

**AND** #0x40

BEQ S1 *; Конец*

**IN** 0x006 *; Прочитать первый символ*

**CMP** #0x0D *; Проверка на стоп-символ, если он найден, то Z=1*

BEQ SAVE *; При обнаружении стоп-символа не читаем второй символ*

**ST** TEMP *; Временно выгружаем первый символ (он перезапишется в спин-лупе чтения второго символа)*

CLA *; Очищаем аккумулятор для спин-лупа*

S2: **IN** 0x007 *; Спин-луп для чтения состояния готовности с ВУ-3*

**AND** #0x40

BEQ S2 *; Конец*

**IN** 0x006 *; Прочитать второй символ*

**CMP** #0x0D *; Проверка на стоп-символ, если он найден, то Z=1*

**PUSHF** *; Сохраняем PS в стек, чтобы NZVC не были утеряны после ADD*

SWAB *; Второй символ в левый байт*

**ADD** TEMP *; Возвращаем первый символ в правый байт*

**POPF** *; Возвращаем состояние NZVC перед командой ADD*

SAVE: **ST** **(**ADDR**)**+ *; Сохранить данные, переместив указатель на следующую ячейку*

BEQ STOP *; Стоит Z=1 - значит, был стоп-символ, нужен останов*

JUMP S1 *; Продолжение чтения*

STOP: **HLT**

ORG 0x270

TEMP: **WORD** ? *; Ячейка для временного сохранения первого символа*

ORG 0x2A0

ADDR: **WORD** 0x05AD *; Адрес сохранения символов*

Листинг

# Пункт 2

Описание программы:

* **Назначение программы:**

Программа осуществляет ввод символов в кодировке КОИ-8 с устройства ВУ-3, записывая их в память подряд в формате:

АДР1: СИМВ2 СИМВ1

АДР2: СИМВ4 СИМВ3

...

СТОП\_СИМВ.

Ввод оканчивается на символе 0D.

* **Описание и назначение исходных данных, область представления и область допустимых значений исходных данных и результата:**

C – вводимый символ в устройство ВУ-3

ADDR – указатель на текущую ячейку массива для сохранения символов

Область представления:

* + C – однобайтовое значение (набор из 8 логических значений),

, где – номер бита;

* + ADDR – адрес ячейки памяти,

Область допустимых значений:

;

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

Основная программа: в ячейках 244-259.

Исходные данные основной программы:

- C: в регистре DR устройства ВУ-3 (адрес: 0x06);

Результат: в ячейках 5AD-5B1

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

первая – 244, последняя – 259

# Пункт 3

Запишем таблицу трассировки программы при вводе первых двух символов (Таблица 2):

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |
| 244 | 0200 | 245 | 0200 | 244 | 0200 | 000 | 0244 | 0000 | 0100 | - | - |
| 245 | 1207 | 246 | 1207 | 245 | 1207 | 000 | 0245 | 0040 | 0100 | - | - |
| 246 | 2F40 | 247 | 2F40 | 246 | 0040 | 000 | 0040 | 0040 | 0000 | - | - |
| 247 | F0FD | 248 | F0FD | 247 | F0FD | 000 | 0247 | 0040 | 0000 | - | - |
| 248 | 1206 | 249 | 1206 | 248 | 1206 | 000 | 0248 | 00F3 | 0000 | - | - |
| 249 | 7F0D | 24A | 7F0D | 249 | 000D | 000 | 000D | 00F3 | 0001 | - | - |
| 24A | F00B | 24B | F00B | 24A | F00B | 000 | 024A | 00F3 | 0001 | - | - |
| 24B | EE24 | 24C | EE24 | 270 | 00F3 | 000 | 0024 | 00F3 | 0001 | 270 | 00F3 |
| 24C | 0200 | 24D | 0200 | 24C | 0200 | 000 | 024C | 0000 | 0101 | - | - |
| 24D | 1207 | 24E | 1207 | 24D | 1207 | 000 | 024D | 0040 | 0101 | - | - |
| 24E | 2F40 | 24F | 2F40 | 24E | 0040 | 000 | 0040 | 0040 | 0001 | - | - |
| 24F | F0FD | 250 | F0FD | 24F | 0040 | 000 | 024F | 0040 | 0001 | - | - |
| 250 | 1206 | 251 | 1206 | 250 | 1206 | 000 | 0250 | 00CD | 0001 | - | - |
| 251 | 7F0D | 252 | 7F0D | 252 | 000D | 000 | 000D | 00CD | 0001 | - | - |
| 252 | 0D00 | 253 | 0D00 | 7FF | 0101 | 7FF | 0252 | 00CD | 0001 | 7FF | 0101 |
| 253 | 0680 | 254 | 0680 | 253 | 0680 | 7FF | 0253 | CD00 | 0001 | - | - |
| 254 | 4E1B | 255 | 4E1B | 270 | 00F3 | 7FF | 001B | CDF3 | 1000 | - | - |
| 255 | 0900 | 256 | 0900 | 7FF | 0101 | 000 | 0255 | CDF3 | 1000 | - | - |
| 256 | EA49 | 257 | EA49 | 5AD | CDF3 | 000 | 0049 | CDF3 | 1000 | 2A0, 5AD | 05AE, CDF3 |
| 257 | F001 | 258 | F001 | 257 | CDF3 | 000 | 0257 | CDF3 | 1000 | - | - |
| 258 | CEEC | 245 | CEEC | 258 | 0245 | 000 | FFEC | CEEC | 1000 | - | - |
| … | | | | | | | | | | | |

Таблица

# Заключение

В результате выполнения данной лабораторной работы изучен ввод-вывод данных во внешние устройства в БЭВМ, а именно асинхронный способ ввода-вывода. Рассмотрены основные принципы асинхронного ввода-вывода (состояние готовности устройства, спин-лупы), внешние устройства в БЭВМ и способы обращения к ним. В итоге была создана программа для ввода символьных данных с ВУ-3 и их сохранения в памяти.

Дополнительное задание

1. Программа осуществляет асинхронный ввод данных с ВУ-3.

2. Программа начинается с адреса 〖244〗\_((16)). Размещаемая строка находится по адресу 5AD\_((16)).

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.

6. После окончания ввода строки осуществляется вывод английских букв в обратном порядке на ВУ-5 (текстовый принтер)