Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №5

Асинхронных обмен данными с ВУ

Вариант 16503

Выполнил:

Шмунк Андрей Александрович

Группа P3108

Преподаватели:

Ткешелашвили Нино Мерабиевна

Клименков Сергей Викторович

Оглавление

[Задание 3](#_Toc162387341)

[Описание программы 3](#_Toc162387342)

[Текст исходной программы 4](#_Toc162387343)

[Описание программы 4](#_Toc162387344)

[Трассировка программы 5](#_Toc162387345)

[Дополнительное задание 5](#_Toc162387346)

[Вывод 8](#_Toc162387347)

# 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 1F016. Размещаемая строка находится по адресу 5AD16.
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.

# Описание программы

**Передаваемое сообщение: «ФИЛИН»**

В кодировке КОИ-8: E6E9 ECE9 EE00

В кодировке UTF-8: D0 A4 D0 98 D0 9B D0 98 D0 9D

В кодировке UTF-16: 04 24 04 18 04 1B 04 18 04 1D

ORG 0x1F0 ;

BEGIN: WORD 0x5AD ; Ссылка на строку

CUR: WORD ? ; Адрес текущей ячейки

FIRST: WORD ? ; Первый символ

SECOND: WORD ? ; Второй символ

MASK: WORD 0xFF ; Маска

START: CLA ; Очистка аккумулятора

LD BEGIN ; Загрузка в аккумулятор адреса первой ячейки

ST CUR ; Сохранение адреса в ячейку счетчика

S0: LD (CUR)+ ; Загрузка первых двух символов

ST SECOND ; Запись первого символа

SWAB ; Обмен старшего и младшего байтов

ST FIRST ; Запись первого символа

W1: IN 7 ; Чтение регистра состояния ВУ-3

AND #0x40 ; Проверка готовности

BEQ W1 ; «Спин-луп»

LD FIRST ; Загрузка второго символа

AND MASK ; Наложение маски

OUT 6 ; Вывод данных на ВУ-3

BEQ STOP ; Проверка на стоп-символ

W2: IN 7 ; Чтение регистра состояния ВУ-3

AND #0x40 ; Проверка готовности

BEQ W2 ; «Спин-луп»

LD SECOND ; Загрузка второго символа

AND MASK ; Наложение маски

OUT 6 ; Вывод данных на ВУ-3

BEQ STOP ; Проверка на стоп-символ

JUMP S0 ; Переход на загрузку символов

STOP: HLT

ORG 0x1F0

WORD 0xE6E9

WORD 0xECE9

WORD 0xEE00

# Текст исходной программы

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код команды | Мнемоника | Комментарии |
| 1F0 | **05AD** | BEGIN | Ссылка на строку |
| 1F1 | **0000** | CUR | Адрес текущей ячейки |
| 1F2 | **0000** | FIRST | Первый символ |
| 1F3 | **0000** | SECOND | Второй символ |
| 1F4 | **00FF** | MASK | Маска |
| 1F5 | **0200** | CLA | Очистка аккумулятора |
| 1F6 | **AEF9** | LD IP-7 | Загрузка в аккумулятор адреса первой ячейки |
| 1F7 | **EEF9** | ST IP-7 | Сохранение адреса в ячейку счетчика |
| 1F8 | **AAF8** | LD (IP-8)+ | Загрузка первых двух символов |
| 1F9 | **EEF9** | ST IP-7 | Запись второго символа |
| 1FA | **0680** | SWAB | Обмен старшего и младшего байтов |
| 1FB | **EEF6** | ST IP-10 | Запись первого символа |
| 1FC | **1207** | IN 7 | Чтение регистра состояния ВУ-3 |
| 1FD | **2F40** | AND #40 | Проверка готовности |
| 1FE | **F0FD** | BEQ IP-2 | «Спин-луп» |
| 1FF | **AEF2** | LD IP-14 | Загрузка второго символа |
| 200 | **2EF3** | AND IP-13 | Наложение маски |
| 201 | **F008** | BEQ IP+8 | Проверка на стоп-символ |
| 202 | **1306** | OUT 6 | Вывод данных на ВУ-3 |
| 203 | **1207** | IN 7 | Чтение регистра состояния ВУ-3 |
| 204 | **2F40** | AND #40 | Проверка готовности |
| 205 | **F0FD** | BEQ IP-2 | «Спин-луп» |
| 206 | **AEEC** | LD IP-20 | Загрузка второго символа |
| 207 | **2EEC** | AND IP-20 | Наложение маски |
| 208 | **F001** | BEQ IP+1 | Проверка на стоп-символ |
| 209 | **1306** | OUT 6 | Вывод данных на ВУ-3 |
| 20A | **CEED** | JUMP IP-19 | Переход на загрузку символов |
| 20B | **0100** | HLT | ОСТАНОВ |

# Описание программы

Программа осуществляет асинхронный вывод данных на ВУ-3. Вывод осуществляется до тех пор, пока не встретится стоп-символ 0x00. Данные из ячейки распределяются по ячейкам для разделения на символы, после чего на них накладывается маска для отделения друг от друга и происходит вывод данных при готовности устройства.

# Трассировка программы

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Выполняемая  команда | | Содержимое регистров 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 |
| 1F0 | E6E9 | 1F1 | E6E9 | 6E9 | 0000 | 000 | 01F0 | 0000 | 004 | 0100 | 6E9 | 0000 |
| 1F1 | 0000 | 1F2 | 0000 | 1F1 | 0000 | 000 | 01F1 | 0000 | 004 | 0100 |  |  |
| 1F2 | 0000 | 1F3 | 0000 | 1F2 | 0000 | 000 | 01F2 | 0000 | 004 | 0100 |  |  |
| 1F3 | 0000 | 1F4 | 0000 | 1F3 | 0000 | 000 | 01F3 | 0000 | 004 | 0100 |  |  |
| 1F4 | 00FF | 1F5 | 00FF | 1F4 | 00FF | 000 | 01F4 | 0000 | 004 | 0100 |  |  |
| 1F5 | 0200 | 1F6 | 0200 | 1F5 | 0200 | 000 | 01F5 | 0000 | 004 | 0100 |  |  |
| 1F6 | AEF9 | 1F7 | AEF9 | 1F0 | E6E9 | 000 | FFF9 | E6E9 | 008 | 1000 |  |  |
| 1F7 | EEF9 | 1F8 | EEF9 | 1F1 | E6E9 | 000 | FFF9 | E6E9 | 008 | 1000 | 1F1 | E6E9 |
| 1F8 | AAF8 | 1F9 | AAF8 | 6E9 | 0000 | 000 | FFF8 | 0000 | 004 | 0100 | 1F1 | E6EA |
| 1F9 | EEF9 | 1FA | EEF9 | 1F3 | 0000 | 000 | FFF9 | 0000 | 004 | 0100 | 1F3 | 0000 |
| 1FA | 0680 | 1FB | 0680 | 1FA | 0680 | 000 | 01FA | 0000 | 004 | 0100 |  |  |
| 1FB | EEF6 | 1FC | EEF6 | 1F2 | 0000 | 000 | FFF6 | 0000 | 004 | 0100 | 1F2 | 0000 |
| 1FC | 1207 | 1FD | 1207 | 1FC | 1207 | 000 | 01FC | 0040 | 004 | 0100 |  |  |
| 1FD | 2F40 | 1FE | 2F40 | 1FD | 0040 | 000 | 0040 | 0040 | 000 | 0000 |  |  |
| 1FE | F0FD | 1FF | F0FD | 1FE | F0FD | 000 | 01FE | 0040 | 000 | 0000 |  |  |
| 1FF | AEF2 | 200 | AEF2 | 1F2 | 0000 | 000 | FFF2 | 0000 | 004 | 0100 |  |  |
| 200 | 2EF3 | 201 | 2EF3 | 1F4 | 00FF | 000 | FFF3 | 0000 | 004 | 0100 |  |  |
| 201 | 1306 | 202 | 1306 | 201 | 1306 | 000 | 0201 | 0000 | 004 | 0100 |  |  |
| 202 | F008 | 20B | F008 | 202 | F008 | 000 | 0008 | 0000 | 004 | 0100 |  |  |
| 20B | 0100 | 20C | 0100 | 20B | 0100 | 000 | 020B | 0000 | 004 | 0100 |  |  |

# Дополнительное задание

На ВУ-6(бегущая строка) рисуется прямой угол (фигура из двух линий). Если значение на ВУ-2 положительное, угол имеет размеры 3х3 клетки, если отрицательное – 5х5. В зависимости от значения последних двух бит отрисовать поворот угла на 90, 180, 270 градусов.

ORG 0x50

NUM: WORD ?

I: IN 5

AND #0x40

BEQ I

IN 4

SXTB

BNS T

BNC F

T:CALL $FOR\_FIVE

JUMP I

F: CALL $FOR\_THREE

JUMP I

HLT

FOR\_THREE:

AND #3

BZC STATE11

CALL $TR

JUMP RT1

STATE11: CMP #3

BZC STATE12

CALL $TF

JUMP RT1

STATE12: CMP #2

BZC STATE13

T\_pos: CALL $TS

JUMP RT1

STATE13: CMP #1

BZC RT1

FF\_pos: CALL $TT

JUMP RT1

RT1: RET

FOR\_FIVE:

AND #3

BZC STATE21

CALL $FR

JUMP RT2

STATE21: CMP #3

BZC STATE22

CALL $FF

JUMP RT2

STATE22: CMP #2

BZC STATE23

CALL $FS

JUMP RT2

STATE23: CMP #1

BZC RT2

CALL $FT

JUMP RT2

RT2: RET

TR:

LD #7

OUT 0x10

LD #1

OUT 0x10

OUT 0x10

CLA

OUT 0x10

RET

FR:

LD #31

OUT 0x10

LD #1

OUT 0x10

OUT 0x10

OUT 0x10

OUT 0x10

CLA

OUT 0x10

RET

TS:

LD #4

OUT 0x10

OUT 0x10

LD #7

OUT 0x10

CLA

OUT 0x10

RET

FS:

LD #16

OUT 0x10

OUT 0x10

OUT 0x10

OUT 0x10

LD #31

OUT 0x10

CLA

OUT 0x10

RET

TT:

LD #1

OUT 0x10

OUT 0x10

LD #7

OUT 0x10

CLA

OUT 0x10

RET

FT:

LD #1

OUT 0x10

OUT 0x10

OUT 0x10

OUT 0x10

LD #31

OUT 0x10

CLA

OUT 0x10

RET

TF:

LD #7

OUT 0x10

LD #4

OUT 0x10

OUT 0x10

CLA

OUT 0x10

RET

FF:

LD #31

OUT 0x10

LD #16

OUT 0x10

OUT 0x10

OUT 0x10

OUT 0x10

CLA

OUT 0x10

RET

# Вывод

При выполнении данной работы я познакомился с асинхронным вводом-выводом данных в БЭВМ, узнал о внешних устройствах, их регистрах и принципе работы. Познакомился с представлением данных в различных кодировках и попрактиковался в выводе данных на ВУ-3.