Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №5

Вариант 6611

Выполнил:

Козаченко Данил Александрович

Группа P3112

Проверил:

Абузов Ярослав Александрович

Содержание

[Задание 3](#_Toc195175524)

[Ход выполнения 3](#_Toc195175525)

[1) Передаваемое сообщение в различных кодировках 3](#_Toc195175526)

[2) Исходной код программы на Ассемблере 3](#_Toc195175527)

[3) Текст исходной программы 4](#_Toc195175528)

[4) Описание программы 4](#_Toc195175529)

[5) Таблица трассировки 5](#_Toc195175530)

[Заключение 6](#_Toc195175531)

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 ввод данных с ВУ-2
2. Программа начинается с адреса 10116. Размещаемая строка находится по адресу 5CF16.
3. Строка должна быть представлена в кодировке Windows-1251.
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.

Ход выполнения

1. Передаваемое сообщение в различных кодировках

«Привет, мир!» - исходное сообщение

CF F0 E8 E2 E5 F2 2C 20 EC E8 F0 21 – в кодировке Windows-1251

D0 9F D1 80 D0 B8 D0 B2 D0 B5 D1 82 2C 20 D0 BC D0 B8 D1 80 21 – в кодировке UTF-8

1F 04 40 04 38 04 32 04 35 04 42 04 2C 00 20 00 3C 04 38 04 40 04 21 00 – в кодировке UTF-16

1. Исходной код программы на Ассемблере

ORG 0x101 ; Начало программы

RES\_ADDRESS: WORD 0x5CF ; Адрес введённой строки

EOF: WORD 0x00 ; Стоп-символ

START: CLA

FIRST: IN 5 ; Проверка регистра состояния ВУ-2

AND #0x40

BEQ FIRST ; "Спин-луп"

IN 4 ; Чтение с ВУ-2

SWAB

ST (RES\_ADDRESS)

CMP EOF ; Проверка на стоп-символ

BEQ STOP

CLA

SECOND: IN 5 ; Проверка регистра состояния ВУ-2

AND #0x40

BEQ SECOND ; "Спин-луп"

IN 4 ; Чтение с ВУ-2

CMP EOF ; Проверка на стоп-символ

BEQ STOP

ADD (RES\_ADDRESS)

ST (RES\_ADDRESS)+

CLA

JUMP FIRST

STOP: HLT

1. Текст исходной программы

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарии** |
| 101 | 05CF | RES\_ADDRESS | Адрес начала строки |
| 102 | 0000 | EOF | Стоп-символ |
| 103 | 0200 | CLA | Очистка аккумулятора  0 ⇒ AC |
| 104 | 1205 | IN 5 | Чтение регистра состояния ВУ-2 |
| 105 | 2F40 | AND #0x40 | Проверка регистра |
| 106 | F0FD | BEQ FIRST | "Спин-луп" |
| 107 | 1204 | IN 4 | Чтение с ВУ-2 |
| 108 | 0680 | SWAB | Обмен младшего и старшего байта |
| 109 | E8F7 | ST (RES\_ADDRESS) | Сохранение введённого символа |
| 10A | 7EF7 | CMP EOF | Проверка на стоп-символ |
| 10B | F00B | BEQ STOP | Если EOF – выход из программы |
| 10C | 0200 | CLA | Очистка аккумулятора  0 ⇒ AC |
| 10D | 1205 | IN 5 | Чтение регистра состояния ВУ-2 |
| 10E | 2F40 | AND #0x40 | Проверка регистра |
| 10F | F0FD | BEQ SECOND | "Спин-луп" |
| 110 | 1204 | IN 4 | Чтение с ВУ-2 |
| 111 | 7EF0 | CMP EOF | Проверка на стоп-символ |
| 112 | F004 | BEQ STOP | Если EOF – выход из программы |
| 113 | 48ED | ADD(RES\_ADDRESS) | Добавление предыдущего символа |
| 114 | EAEC | ST(RES\_ADDRESS)+ | Сохранение 2-х символов в результат |
| 115 | 0200 | CLA | Очистка аккумулятора  0 ⇒ AC |
| 116 | CEED | JUMP FIRST | Переход в начало программы |
| 117 | 0100 | HLT | Отключение тактового генератора, остановка программы |
| … | | | |
| 5CF | 0000 | RESULT | Начало строки |
| … | | | |

1. Описание программы

***Программа реализуют следующую функцию:***

Посимвольный ввод данных с ВУ-2. Программа записывает символы до введения стоп-символа с кодировкой 0x00.

***Область представления:***

* + RES\_ADDRESS – беззнаковое, 11-ти разрядное число с адресом результата
  + EOF – 16-ти разрядная константа
  + RESULT – 16-ти разрядные ячейки, хранящие по два символа в кодировке Windows-1251

***Область допустимых значений:***

* RES\_ADDRESS ∈ [0, 100] ∪ [118, 7FF]
* EOF ∈ [00, FF]
* **Расположение в памяти ЭВМ программы, исходных данных и результатов**

103-117 – основная программа;

101, 102 – исходные данные;

5CF – итоговый результат

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

103 – адрес первой команды

117 – адрес последней команды

1. Таблица трассировки

Для трассировки были введены значения CF F0. «Привет, мир!» в кодировке Windows-1251

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |
| 103 | 0200 | 104 | 0200 | 103 | 0200 | 000 | 0103 | 0000 | 0100 |  |  |
| 104 | 1205 | 105 | 1205 | 104 | 1205 | 000 | 0104 | 0040 | 0100 |  |  |
| 105 | 2F40 | 106 | 2F40 | 105 | 0040 | 000 | 0040 | 0040 | 0000 |  |  |
| 106 | F0FD | 107 | F0FD | 106 | F0FD | 000 | 0106 | 0040 | 0000 |  |  |
| 107 | 1204 | 108 | 1204 | 107 | 1204 | 000 | 0107 | 00CF | 0000 |  |  |
| 108 | 0680 | 109 | 0680 | 108 | 0680 | 000 | 0108 | CF00 | 1000 |  |  |
| 109 | E8F7 | 10A | E8F7 | 5CF | CF00 | 000 | FFF7 | CF00 | 1000 | 5CF | CF00 |
| 10A | 7EF7 | 10B | 7EF7 | 102 | 0000 | 000 | FFF7 | CF00 | 1001 |  |  |
| 10B | F00B | 10C | F00B | 10B | F00B | 000 | 010B | CF00 | 1001 |  |  |
| 10C | 0200 | 10D | 0200 | 10C | 0200 | 000 | 010C | 0000 | 0101 |  |  |
| 10D | 1205 | 10E | 1205 | 10D | 1205 | 000 | 010D | 0000 | 0101 |  |  |
| 10E | 2F40 | 10F | 2F40 | 10E | 0040 | 000 | 0040 | 0000 | 0101 |  |  |
| 10F | F0FD | 10D | F0FD | 10F | F0FD | 000 | FFFD | 0000 | 0101 |  |  |
| 10D | 1205 | 10E | 1205 | 10D | 1205 | 000 | 010D | 0040 | 0101 |  |  |
| 10E | 2F40 | 10F | 2F40 | 10E | 0040 | 000 | 0040 | 0040 | 0001 |  |  |
| 10F | F0FD | 110 | F0FD | 10F | F0FD | 000 | 010F | 0040 | 0001 |  |  |
| 110 | 1204 | 111 | 1204 | 110 | 1204 | 000 | 0110 | 00F0 | 0001 |  |  |
| 111 | 7EF0 | 112 | 7EF0 | 102 | 0000 | 000 | FFF0 | 00F0 | 0001 |  |  |
| 112 | F004 | 113 | F004 | 112 | F004 | 000 | 0112 | 00F0 | 0001 |  |  |
| 113 | 48ED | 114 | 48ED | 5CF | CF00 | 000 | FFED | CFF0 | 1000 |  |  |
| 114 | EAEC | 115 | EAEC | 5CF | CFF0 | 000 | FFEC | CFF0 | 1000 | 101 | 05D0 |
| 5CF | CFF0 |
| 115 | 0200 | 116 | 0200 | 115 | 0200 | 000 | 0115 | 0000 | 0100 |  |  |
| 116 | CEED | 104 | CEED | 116 | 0104 | 000 | FFED | 0000 | 0100 |  |  |
| 104 | 1205 | 105 | 1205 | 104 | 1205 | 000 | 0104 | 0000 | 0100 |  |  |
| 105 | 2F40 | 106 | 2F40 | 105 | 0040 | 000 | 0040 | 0000 | 0100 |  |  |
| 106 | F0FD | 104 | F0FD | 106 | F0FD | 000 | FFFD | 0000 | 0100 |  |  |
| 104 | 1205 | 105 | 1205 | 104 | 1205 | 000 | 0104 | 0040 | 0100 |  |  |
| 105 | 2F40 | 106 | 2F40 | 105 | 0040 | 000 | 0040 | 0040 | 0000 |  |  |
| 106 | F0FD | 107 | F0FD | 106 | F0FD | 000 | 0106 | 0040 | 0000 |  |  |
| 107 | 1204 | 108 | 1204 | 107 | 1204 | 000 | 0107 | 0000 | 0000 |  |  |
| 108 | 0680 | 109 | 0680 | 108 | 0680 | 000 | 0108 | 0000 | 0100 |  |  |
| 109 | E8F7 | 10A | E8F7 | 5D0 | 0000 | 000 | FFF7 | 0000 | 0100 | 5D0 | 0000 |
| 10A | 7EF7 | 10B | 7EF7 | 102 | 0000 | 000 | FFF7 | 0000 | 0101 |  |  |
| 10B | F00B | 117 | F00B | 10B | F00B | 000 | 000B | 0000 | 0101 |  |  |
| 117 | 0100 | 118 | 0100 | 117 | 0100 | 000 | 0117 | 0000 | 0101 |  |  |

Заключение

Я научился выполнять асинхронный ввод-вывод данных в БЭВМ, изучил различные внешние устройства и их регистры. Познакомился с различными командами для ввода-вывода данных в БЭВМ.