

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО».

Факультет программной инженерии и компьютерной техники

Основы профессиональной деятельности

Лабораторная работа №5

Асинхронный обмен данными с ВУ

Вариант 8634

Выполнил: Свечников Константин Денисович, Р3130

Проверил: Саржевский И. А., преподаватель практики

Оглавление

Задание.....	2
Ход работы	2
Описание программы:	Ошибка! Закладка не определена.
Исходные данные:	4
Область представления:	4
Область допустимых значений	4
Трассировка программы	5
Заключение.....	6

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 148_{16} . Размещаемая строка находится по адресу 595_{16} .
3. Строка должна быть представлена в кодировке КОИ-8.
4. Формат представления строки в памяти: АДР0: ДЛИНА АДР1: СИМВ2 СИМВ1 АДР2: СИМВ4 СИМВ3 ..., где ДЛИНА - 16 разрядное слово, где значащими являются 8 младших бит.
5. Ввод строки начинается со ввода количества символов (1 байт), и должен быть завершен по вводу их необходимого количества.

Ход работы

Текст исходной программы описан ниже:

Асинхронный ввод данных (в кодировке КОИ-8-R) с ВУ-2 (по количеству символов)
ORG 0x148 ; <i>Начало кода</i> STR: WORD 0x0595 ; <i>Адрес начала данных</i> POINTER: WORD ? ; <i>Указатель на текущее слово данных</i> COUNT: WORD ? ; <i>Количество символов (1 байт)</i>
START: CLA ; <i>Очистить аккумулятор</i>
GET_LEN: IN 0x05 ; <i>Ожидание ввода количества символов</i> AND #0x40 ; <i>Бит 6 SR == 0 («Готов» нажат?)</i> BEQ GET_LEN ; <i>Нет – «Спин-луп»</i> IN 0x04 ; <i>Иначе – читаем количество с ВУ-2...</i> ST COUNT ; <i>... и сохраняем в память</i>
LD STR ; <i>Адрес начала данных</i> ST POINTER ; <i>Инициализируем указатель</i> LD COUNT ; <i>Загружаем длину слова</i> BEQ PHLT ; <i>Если длина 0 – стоп, ...</i> ST (POINTER)+ ; <i>иначе – сохраняем в память</i>
S1: IN 0x05 ; <i>Ожидание ввода первого символа</i> AND #0x40 ; <i>Бит 6 SR == 0 («Готов» нажат?)</i> BEQ S1 ; <i>Нет – «Спин-луп»</i> IN 0x04 ; <i>Иначе – читаем символ с ВУ-2 ...</i> ST (POINTER) ; <i>... и сохраняем его в память</i> LD COUNT ; <i>Проверяем количество символов</i> DEC ; <i>Уменьшаем на 1</i> BEQ PHLT ; <i>Если 0 – закончить программу</i> ST COUNT ; <i>Сохраняем в переменную</i>
S2: IN 0x05 ; <i>Ожидание ввода второго символа</i> AND #0x40 ; <i>Бит 6 SR == 0 («Готов» нажат?)</i> BEQ S2 ; <i>Нет – «Спин-луп»</i> LD (POINTER) ; <i>Загружаем из памяти первый символ ...</i> SWAB ; <i>... и меняем порядок байтов (СИМВ1 0)</i> IN 0x04 ; <i>Читаем символ с ВУ-2...</i> SWAB ; <i>... и снова меняем порядок байтов (СИМВ2 СИМВ1)</i> ST (POINTER)+ ; <i>Сохраняем окончательный элемент строки</i> LD COUNT ; <i>Проверяем количество символов</i> DEC ; <i>Уменьшаем на 1</i> ST COUNT ; <i>Сохраняем в переменную</i> BEQ PHLT ; <i>Если 0 – закончить программу</i> JUMP S1 ; <i>И по новой</i>
PHLT: HLT ; <i>Выход из программы</i>

ORG 0x0595 ; Тут содержится длина строки, а далее – сама строка
WORD ?

Назначение программы:

Посимвольный асинхронный ввод строки, закодированной в КОИ-8-R с ВУ-2.

Исходные данные:

- Исходные данные: 148. В этой ячейке содержится адрес начала строки
- Переменные: 149 – 14A. Ячейка 149 используется для указателя текущего адреса (текущих 2 символа строки). Ячейка 14A используется для счетчика символов строки
- Программа: 14B – 169 (14B – адрес первой команды, 169 – последней)
- Результат: 595 – ... В ячейке 595 содержится длина вводимой строки, в последующих – сама строка.

Область представления:

- STR – беззнаковое 11-разрядное число
- COUNT – беззнаковое 11-разрядное число
- POINTER – беззнаковое 12-разрядное число
- Ячейка 595 – беззнаковое 11-разрядное число
- Ячейки с 596 – до 2-х восьмиразрядных символов в кодировке КОИ-8. Младший байт – 1-й символ элемента, старший байт – 2-й.

Область допустимых значений:

- $$\begin{cases} 0x16A \leq STR \leq 2^{11} - 2 - \text{ОКРУГ.ВВЕРХ}(\frac{COUNT}{2}) \\ 0 \leq COUNT \leq 2^{12} \end{cases}$$
- $$\begin{cases} 0x000 \leq STR \leq 0x148 - 1 - \text{ОКРУГ.ВВЕРХ}(\frac{COUNT}{2}) \\ 0 \leq COUNT \leq 0x290 \end{cases}$$
- Символ строки:

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
8.	— 2500	 2502	┐ 250C	┌ 2510	└ 2514	┘ 2518	└┐ 251C	┐┌ 2524	┘┘ 252C	└┘ 2534	┘└ 253C	■ 2580	■ 2584	■ 2588	■ 258C	■ 2590
9.	░ 2591	▒ 2592	▓ 2593	┌ 2320	■ 25A0	· 2219	√ 221A	≈ 2248	≤ 2264	≥ 2265	NBSP 00A0	┘ 2321	° 00B0	² 00B2	· 00B7	÷ 00F7
A.	= 2550	 2551	ƒ 2552	ё 0451	п 2553	р 2554	я 2555	т 2556	х 2557	л 2558	л 2559	л 255A	л 255B	л 255C	л 255D	л 255E
B.	 255F	 2560	 2561	Ё 0401	 2562	 2563	т 2564	т 2565	т 2566	т 2567	л 2568	л 2569	л 256A	л 256B	л 256C	© 00A9
C.	ю 044E	а 0430	б 0431	ц 0446	д 0434	е 0435	ф 0444	г 0433	х 0445	н 0438	й 0439	к 043A	л 043B	м 043C	н 043D	о 043E
D.	п 043F	я 044F	р 0440	с 0441	т 0442	у 0443	ж 0436	в 0432	ь 044C	ы 044B	з 0437	ш 0448	э 044D	щ 0449	ч 0447	ъ 044A
E.	Ю 042E	А 0410	Б 0411	Ц 0426	Д 0414	Е 0415	Ф 0424	Г 0413	Х 0425	И 0418	Й 0419	К 041A	Л 041B	М 041C	Н 041D	О 041E
F.	П 041F	Я 042F	Р 0420	С 0421	Т 0422	У 0423	Ж 0416	В 0412	Ь 042C	Ы 042B	З 0417	Ш 0428	Э 042D	Щ 0429	Ч 0427	Ъ 042A

Трассировка программы

Заданная строка: Волосатый кот

ВУ-8 (клавиатура) – ввод, ВУ-5 (принтер) – вывод

```

ORG 0x148; Начало кода
STR: WORD 0x0595 ; Адрес начала данных
POINTER: WORD ? ; Указатель на текущее слово данных

START: CLA ; Очистить аккумулятор
LD STR ; Инициализация указателя
ST POINTER

S1: ; Ввод первого символа, остановка, если стоп-символ (Enter)
IN 0x19
AND #0x40
BEQ S1
IN 0x18
ST (POINTER)
CMP #0x0A
BEQ PHLT

S1P: ; Вывод первого символа в принтер
IN 0x0D
AND #0x40
BEQ S1P
LD (POINTER)
OUT 0xC

```

S2: Ввод второго символа, остановка, если стоп-символ (Enter)

```
IN 0x19
AND #0x40
BEQ S2
LD (POINTER)
SWAB
IN 0x18
SWAB
ST (POINTER)
```

SWAB

SXTB

CMP #0x0A

BEQ PHLT

S2P: ; Вывод второго символа в принтер

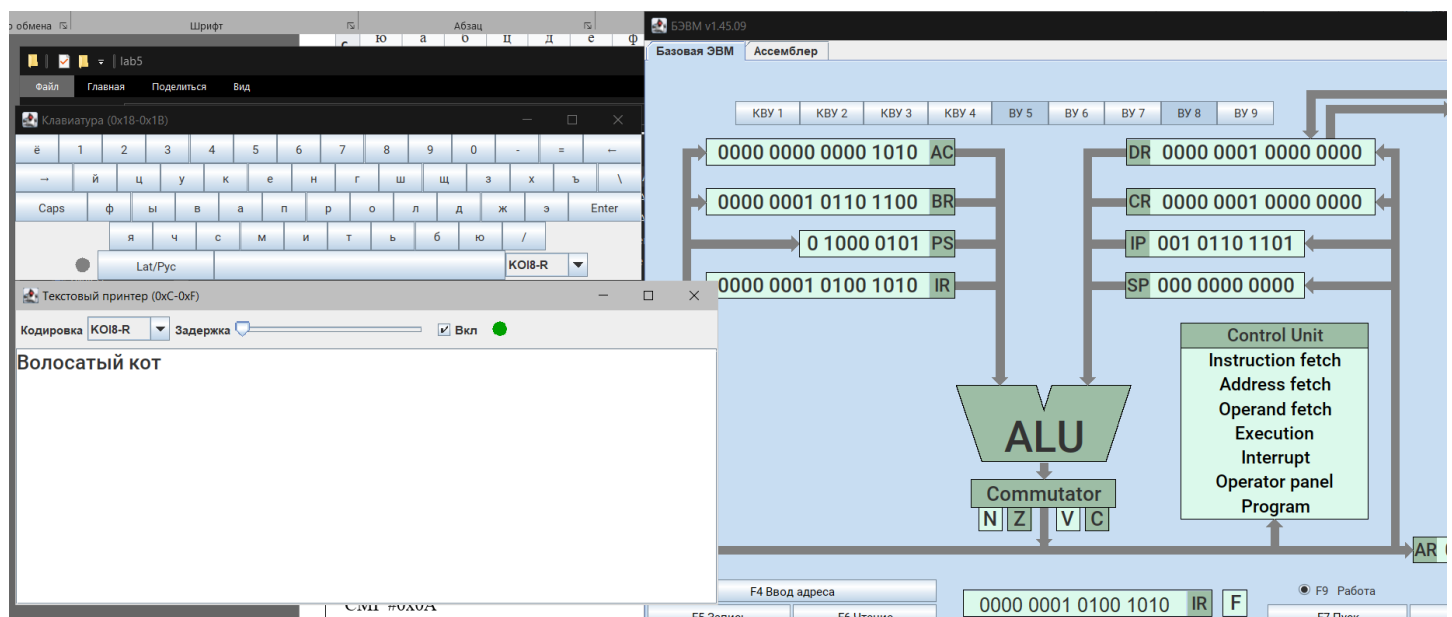
```
IN 0x0D
AND #0x40
BEQ S2P
LD (POINTER)+
SWAB
OUT 0xC
```

JUMP S1; По новой

PHLT: HLT ; Выход из программы

ORG 0x0595 ; Тут содержится сама строка
WORD ?

Пример вывода программы:



Заключение

Прикольная работа.