Федеральное государственное автономное образовательное учреждение высшего образования

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Дисциплина:

Основы профессиональной деятельности

Лабораторная работа №5

" Асинхронный обмен данными с ВУ"

Вариант: 51213

Выполнил:

Кузнецов Максим Александрович

Группа: Р3111

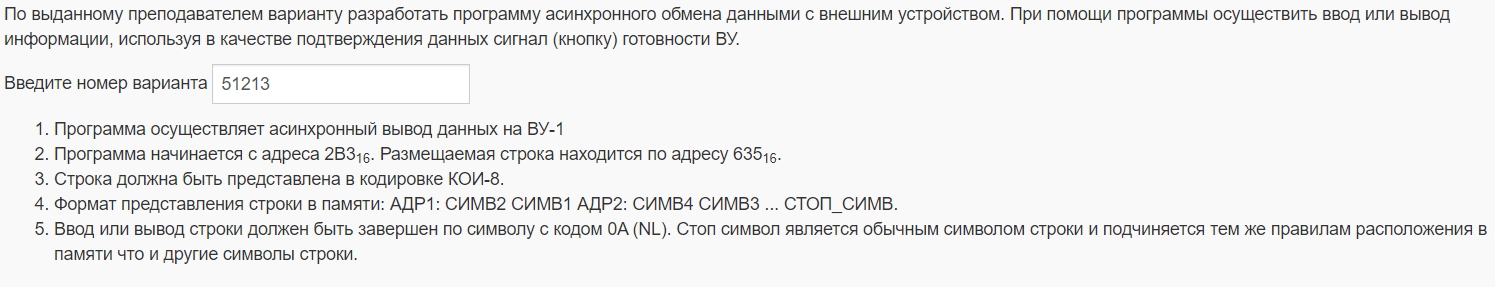
Преподаватель:

Блохина Елена Николаевна

Санкт-Петербург

2021

# Задание:



# Программа:

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код команды | Мнемоника | Комментарий |
| 2B0 | 0635 |  | Указатель на адрес текущей строки |
| 2B1 | 000A |  | Стоп символ |
| 2B2 | 00FF |  | Нужно, чтобы оставить только 1-ый символ в строке |
| 2B3 | +0200 | CLA | Очистка аккумулятора |
| 2B4 | 1203 | IN 3 | Ожидание ввода сигнала готовности |
| 2B5 | 2F40 | AND #0x40 | Проверки кнопки |
| 2B6 | F0FD | BEQ (IP-3) | Возврат в ожидание, если кнопка не нажата |
| 2B7 | A8F8 | LD (IP-8) | Загрузка двух символов |
| 2B8 | 1302 | OUT 2 | Вывод только 1-го символа |
| 2B9 | 2EF8 | AND (IP-8) | Оставляем ТОЛЬКО 1-ый символ |
| 2BA | 7EF6 | CMP (IP-10) | Сравниваем данный символ со стоп-символом – выставляем по итогу флаги |
| 2BB | F00A | BEQ (IP+10) | Если Z==1, то есть это было стоп-слово, тогда прекращаем исполнение программы. |
| 2BC | 1203 | IN 3 | Иначе же, продолжаем исполнение программы, ожидаем ввода сигнала готовности |
| 2BD | 2F40 | AND #0x40 | Проверка кнопки |
| 2BE | F0FD | BEQ (IP-3) | Возврат в ожидание, если кнопка не нажата |
| 2BF | AAF0 | LD(IP-10)+ | Загружаем два символа с последующим автоинкрементным увеличением номера ячейки |
| 2C0 | 0680 | SWAB | Меняем местами эти символы, чтобы теперь проверить 2-ой символ |
| 2C1 | 1302 | OUT 2 | Выводим 2-ой символ |
| 2C2 | 2EEF | AND (IP-17) | Оставляем ТОЛЬКО 2-ой символ |
| 2C3 | 7EED | CMP (IP-19) | Сравниваем данный символ со стоп-символом – выставляем по итогу флаги |
| 2C4 | F001 | BEQ (IP+1) | Если Z==1, то есть это было стоп-слово, тогда прекращаем исполнение программы. |
| 2C5 | CEEE | JUMP (IP-18) | Иначе же - переходим к началу нашего цикла |
| 2C6 | 0100 | HLT | Остановка |

# Описание программы

Задается строка в кодировке КОИ-8 и асинхронно выводится на ВУ-1. Цикл вывода заканчивается при стоп-символе 0А, каждая пара символов находится в разных ячейках памяти, начиная с 0x635. Сама программа с 0x2B3. По нажатии на кнопку “готов” программа выводит следующие два символа из строки.

# Область представления:

*Исходных данных:*

2B0, 2B1 → знаковые числа в 16-раздрядном двоичном формате

# Область допустимых значений:

Заданная строка начинается в ячейке 63516, строка может находиться с ячейки 63516 по ячейку 2AF, так как память циклична.

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

2B3 – 2C6 – команды

2B0 – 2B2, 635 – 2AF – исходные данные

# Адреса первой и последней выполняемых команд программы:

Адрес первой выполняемой команды: 2B3

Адрес последней выполняемой команды: 2C6

# Трассировка программы:

1. 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** | **PS** | **NZVC** | **Адр** | **Знач** |
| **Загрузка и вывод первого символа** | | | | | | | | | | | | | |
| **2B3** | **0200** | **2B4** | **0200** | | **2B4** | **0200** | **000** | **02B3** | **0000** | **004** | **0100** |  |  |
| **2B4** | **1203** | **2B5** | **1203** | | **2B4** | **1203** | **000** | **02B4** | **0000** | **004** | **0100** |  |  |
| **2B5** | **2F40** | **2B6** | **2F40** | | **2B5** | **0040** | **000** | **0040** | **0000** | **004** | **0100** |  |  |
| **2B6** | **F0FD** | **2B7** | **F0FD** | | **2B6** | **F0FD** | **000** | **02B6** | **0040** | **000** | **0000** |  |  |
| **2B7** | **A8F8** | **2B8** | **A8F8** | | **635** | **F0EF** | **000** | **FFF8** | **F0EF** | **008** | **1000** |  |  |
| **2B8** | **1302** | **2B9** | **1302** | | **2B8** | **1302** | **000** | **02B8** | **F0EF** | **008** | **1000** | REG | EF |
| **2B9** | **2EF8** | **2BA** | **2EF8** | | **2B2** | **00FF** | **000** | **FFF8** | **00EF** | **000** | **0000** |  |  |
| **2BA** | **7EF6** | **2BB** | **7EF6** | | **2BB** | **000A** | **000** | **FFF6** | **00EF** | **001** | **0001** |  |  |
| **2BB** | **F00A** | **2BC** | **F00A** | | **2BC** | **F00A** | **000** | **02BB** | **00EF** | **001** | **0001** |  |  |
|  | | | | | | | | | | | | | |
| **Загрузка второго символа** | | | | | | | | | | | | | |
| **2BC** | **1203** | **2BD** | **1203** | | **2BC** | **1203** | **000** | **02B4** | **0000** | **004** | **0100** |  |  |
| **2BD** | **2F40** | **2BE** | **2F40** | | **2BD** | **0040** | **000** | **0040** | **0000** | **004** | **0100** |  |  |
| **2BE** | **F0FD** | **2BF** | **F0FD** | | **2BE** | **F0FD** | **000** | **02B6** | **0040** | **000** | **0000** |  |  |
| **2BF** | **AAF0** | **2C0** | **AAF0** | | **2BF** | **F0EF** | **000** | **FFF0** | **F0EF** | **009** | **1001** |  |  |
| **2C0** | **0680** | **2C1** | **0680** | | **2C0** | **0680** | **000** | **02C0** | **EFF0** | **009** | **1001** |  |  |
| **2C1** | **1302** | **2C2** | **1302** | | **2C1** | **1302** | **000** | **02C1** | **EFF0** | **009** | **1001** | **REG** | **F0** |
| **2C2** | **2EEF** | **2C3** | **2EEF** | | **2B2** | **00FF** | **000** | **FFEF** | **00F0** | **001** | **0001** |  |  |
| **2C3** | **7EED** | **2C4** | **7EED** | | **2B1** | **000A** | **000** | **FFED** | **000A** | **005** | **0101** |  |  |
| **2C4** | **F001** | **2C6** | **F001** | | **2C4** | **F001** | **000** | **0001** | **000A** | **005** | **0101** |  |  |
| **2C6** | **0100** | **2C7** | **0100** | | **2C6** | **0100** | **000** | **02C6** | **000A** | **005** | **0101** |  |  |

# Код на ассемблере:

ORG 0x2B0

ADDRESS: WORD 0x635

STOPWORD: WORD 0x0A

MASK: WORD 0xFF

START: CLA

INPUT1LETTER: IN 3

AND #0x40

BEQ INPUT1LETTER

LD (ADDRESS)

OUT 2

AND MASK

CMP STOPWORD

BEQ EXIT

INPUT2LETTER: IN 3

AND #0x40

BEQ INPUT2LETTER

LD (ADDRESS)+

SWAB

OUT 2

AND MASK

CMP STOPWORD

BEQ EXIT

JUMP INPUT1LETTER

EXIT : HLT

ORG 0x635

WORD 0xF0EF

WORD 0x0AE4

# Кодировка

В кодировке КОИ-8 здесь записано слово «ОПД»

EF – О(U+041E F0 – П(U+041F E4 – Д(U+0414, 0A – стоп-символ

# Доп. Задание с выводом надписи на ВУ-6 (Бегущая строка)

Код на ассемблере:

ORG 0xF5

WORD $symbU

ORG 0xEE

WORD $symbN

ORG 0xE9

WORD $symbI

ORG 0xF7

WORD $symbV

ORG 0xE5

WORD $symbE

ORG 0xF2

WORD $symbR

ORG 0xF3

WORD $symbS

ORG 0xF4

WORD $symbT

ORG 0xED

WORD $symbM

ORG 0xEF

WORD $symbO

ORG 0x9A

WORD $whitespace

ORG 0x0A

WORD $stopSymbol

ORG 0x100

symbU: LD #0x00

OUT 0x10

LD #0x81

OUT 0x10

LD #0x41

OUT 0x10

LD #0x22

OUT 0x10

LD #0x14

OUT 0x10

LD #0x08

OUT 0x10

LD #0x10

OUT 0x10

LD #0x20

OUT 0x10

LD #0x40

OUT 0x10

LD #0x80

OUT 0x10

RET

symbN: LD #0x00

OUT 0x10

LD #0xFF

OUT 0x10

LD #0x18

OUT 0x10

OUT 0x10

OUT 0x10

LD #0xFF

OUT 0x10

RET

symbI: LD #0x00

OUT 0x10

LD #0xFF

OUT 0x10

LD #0x02

OUT 0x10

LD #0x0C

OUT 0x10

LD #0x18

OUT 0x10

LD #0x30

OUT 0x10

LD #0x40

OUT 0x10

LD #0xFF

OUT 0x10

RET

symbV: LD #0x00

OUT 0x10

LD #0xFF

OUT 0x10

LD #0x99

OUT 0x10

OUT 0x10

LD #0x69

OUT 0x10

LD #0x06

OUT 0x10

RET

symbE: LD #0x00

OUT 0x10

LD #0xFF

OUT 0x10

LD #0x99

OUT 0x10

OUT 0x10

LD #0x81

OUT 0x10

OUT 0x10

RET

symbR: LD #0x00

OUT 0x10

LD #0xFF

OUT 0x10

LD #0x90

OUT 0x10

OUT 0x10

OUT 0x10

LD #0x60

OUT 0x10

RET

symbS: LD#0x00

OUT 0x10

LD #0x7E

OUT 0x10

LD #0x81

OUT 0x10

OUT 0x10

OUT 0x10

LD #0x66

OUT 0x10

RET

symbT: LD #0x00

OUT 0x10

LD #0x80

OUT 0x10

OUT 0x10

LD #0xFF

OUT 0x10

LD #0x80

OUT 0x10

OUT 0x10

RET

symbM: LD #0x00

OUT 0x10

LD #0xFF

OUT 0x10

LD #0x60

OUT 0x10

LD #0x30

OUT 0x10

LD #0x60

OUT 0x10

LD #0xFF

OUT 0x10

RET

symbO: LD #0x00

OUT 0x10

LD #0x7E

OUT 0x10

LD #0x81

OUT 0x10

OUT 0x10

OUT 0x10

LD #0x7E

OUT 0x10

RET

whitespace: LD #0x00

OUT 0x10

OUT 0x10

OUT 0x10

OUT 0x10

RET

stopSymbol: LD #0x00

OUT 0x10

RET

ORG 0x2AF

ADDRESS: WORD 0x635

STOPWORD: WORD 0x0A

MASK: WORD 0xFF

CURRENT: WORD 0x00

START: CLA

INPUT1LETTER: LD (ADDRESS)

AND MASK

CMP STOPWORD

BEQ EXIT

ST CURRENT

LD (CURRENT)

ST CURRENT

CALL (CURRENT)

INPUT2LETTER: LD (ADDRESS)+

SWAB

AND MASK

CMP STOPWORD

BEQ EXIT

ST CURRENT

LD (CURRENT)

ST CURRENT

CALL (CURRENT)

JUMP INPUT1LETTER

EXIT: HLT

ORG 0x635

WORD 0xEEF5

WORD 0xF7E9

WORD 0xF2E5

WORD 0xE9F3

WORD 0xE5F4

WORD 0x9AF4

WORD 0xF4E9

WORD 0xEFED

WORD 0x000A

Принцип работы: в начале у нас с ячеек 00 по 100 хранятся в ячейках с номером, равным кодировке буквы, хранится ссылка на адрес ячейки для отрисовки данного элемента. Затем, идут блоки реализации, а затем сам запуск программы, и отрисовка надписи «Университет ИТМО».