Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО» Факультет программной инженерии и компьютерной техники

Отчет

# по лабораторной работе №5

**«Асинхронный обмен данными с ВУ»**

по дисциплине «Основы профессиональной деятельности» вариант 86453

Выполнил: Кобелев Р.П.,

группа Р3112

Преподаватель: Осипов С.В.

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.

1. Программа осуществляет асинхронный ввод данных с ВУ-2
2. Программа начинается с адреса 14416. Размещаемая строка находится по адресу 58416.
3. Строка должна быть представлена в кодировке КОИ-8.
4. Формат представления строки в памят и: 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Программа** | | | |
| **Адрес** | **Код команды** | **Мнемоника** | **Описание кода** |
| 143 | 583 | LEN: WORD $A\_0 | Адрес длины строки |
| 144 | 584 | AD: WORD $A\_1 | Адрес текущей ячейки |
| 145 | 0200 | START: CLA | Чистим аккумулятор |
| 146 | 1205 | W1: IN 5 | Ждём сигнала "ГОТОВ" с ВУ-2 |
| 147 | 2F40 | AND #0x40 |
| 148 | F0FD | BEQ W1 |
| 149 | 1204 | IN 4 | Берём из ВУ-2 длину строки |
| 14A | F012 | BEQ STOP | Если длина нулевая -> STOP |
| 14B | E8F7 | ST (LEN) | Загружаем длину строки в ячейку |
| 14C | 200 | S: CLA | Чистим аккумулятор |
| 14D | 0C00 | PUSH | Загружаем в ячейку один символ |
| 14E | D15E | CALL $I |
| 14F | E8F4 | ST (AD) |
| 150 | A8F2 | LD (LEN) | Проверка на то, являлся ли символ последним, причём длина строки - нечётная |
| 151 | 6F01 | SUB #0x1 |
| 152 | E8F0 | ST (LEN) |
| 153 | F101 | BNE T |
| 154 | CE06 | JUMP L |
| 155 | A8EE | T: LD (AD) | Добавляем в ячейку второй символ и сохраняем, при том адрес ячейки инкрементируем |
| 156 | 0680 | SWAB |
| 157 | 0C00 | PUSH |
| 158 | D15E | CALL $I |
| 159 | 0680 | SWAB |
| 15A | EAE9 | ST (AD)+ |
| 15B | 88E7 | L: LOOP (LEN) | Проверка оставшийся длины строки |
| 15C | CEEF | JUMP S | Повтор цикла |
| 15D | 0100 | STOP: HLT | Стоп программа |
|  |  | Подпрограмма |  |
| 15E | 1205 | I: IN 5 | Ждём сигнала "ГОТОВ" с ВУ-2 |
| 15F | 2F40 | AND #0x40 |
| 160 | F0FD | BEQ I |
| 161 | 1204 | IN 4 | Берём из ВУ-2 символ |
| 162 | 4C01 | ADD &1 | Суммируем с &1 |
| 163 | 0E00 | SWAP | Эпилог: проводим манипуляции со стеком, чтобы выйти из подпрограммы, удалить аргументы в стеке и сохранить результат в аккумуляторе |
| 164 | EC01 | ST &1 |
| 165 | 0E00 | SWAP |
| 166 | 0E00 | SWAP |
| 167 | 0800 | POP |
| 168 | 0A00 | RET | Выход из подпрограммы |
|  |  |  |  |
| 583 | 1111 | A\_0:WORD 0x1111 | Длина строки |
| 584 | 2222 | A\_1: WORD 0x2222 | Представление строки в памяти |
| 585 | АДР\_2 |  |  |

# Описание программы

* Ввод данных в память из ВУ-2
* Подпрограмма получает в виде аргумента число с пустым младшим байтом и прибавляет к этому числу символ из ВУ-2
* Программа находится по адресам [145-15D]
* Подпрограмма находится по адресам [15E-168]
* Константы находятся по адресам [143-144]
* Результат программы находится по адресу [583-…]

# Область представления

Длина строки – 8-разрядное беззнаковое число

Ячейка с двумя символами – 16-разрядное беззнаковое число

# Область допустимых значений

# *Программа на ассемблере: исправить длину*

ORG 0x143

LEN: WORD $A\_0

AD: WORD $A\_1

START: CLA

W1: IN 5

AND #0x40

BEQ W1

IN 4

BEQ STOP

ST (LEN)

S: CLA

PUSH

CALL $I

ST(AD)

LD (LEN)

SUB #0x1

ST (LEN)

BNE T

JUMP L

T:LD (AD)

SWAB

PUSH

CALL $I

SWAB

ST (AD)+

L: LOOP (LEN)

JUMP S

STOP: HLT

I: IN 5

AND #0x40

BEQ I

IN 4

ADD &1

SWAP

ST &1

SWAP

SWAP

POP

RET

ORG 0x583

A\_0: WORD 0x1111

A\_1: WORD 0x2222

# Трассировка

Введённая строка – Кв3

Перевод: К – EB16 = 111010112

в – D716 = 110101112

3 – 0316 = 000000112

Длина строки – 3

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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* |
| 145 | 200 | 146 | 200 | 145 | 200 | 0000 | 145 | 0000 | 0100 |  |  |
| 146 | 1205 | 147 | 1205 | 146 | 1205 | 0000 | 146 | 0040 | 0100 |  |  |
| 147 | 2F40 | 148 | 2F40 | 147 | 40 | 0000 | 40 | 0040 | 0000 |  |  |
| 148 | F0FD | 149 | F0FD | 148 | F0FD | 0000 | 148 | 0040 | 0000 |  |  |
| 149 | 1204 | 14A | 1204 | 149 | 1204 | 0000 | 149 | 0003 | 0000 |  |  |
| 14A | F012 | 14B | F012 | 14A | F012 | 0000 | 014A | 0003 | 0000 |  |  |
| 14B | E8F7 | 14C | E8F7 | 583 | 0003 | 0000 | FFF7 | 0003 | 0000 | 583 | 3 |
| 14C | 200 | 14D | 200 | 14C | 0200 | 0000 | 014C | 0000 | 0100 |  |  |
| 14D | 0C00 | 14E | 0C00 | 7FF | 0000 | 7FF | 014D | 0000 | 0100 | 7FF | 0 |
| 14E | D15E | 15E | D15E | 7FE | 014F | 7FE | D15E | 0000 | 0100 | 7FE | 014F |
| 15E | 1205 | 15F | 1205 | 15E | 1205 | 7FE | 015E | 0040 | 0100 |  |  |
| 15F | 2F40 | 160 | 2F40 | 15F | 0040 | 7FE | 40 | 0040 | 0000 |  |  |
| 160 | F0FD | 161 | F0FD | 160 | F0FD | 7FE | 160 | 0040 | 0000 |  |  |
| 161 | 1204 | 162 | 1204 | 161 | 1204 | 7FE | 161 | 00EB | 0000 |  |  |
| 162 | 4C01 | 163 | 4C01 | 7FF | 0000 | 7FE | 0001 | 00EB | 0000 |  |  |
| 163 | 0E00 | 164 | 0E00 | 7FE | 00EB | 7FE | 014F | 014F | 0000 | 7FE | 00EB |
| 164 | EC01 | 165 | EC01 | 7FF | 014F | 7FE | 0001 | 014F | 0000 | 7FF | 014F |
| 165 | 0E00 | 166 | 0E00 | 7FE | 014F | 7FE | 00EB | 00EB | 0000 | 7FE | 014F |
| 166 | 0E00 | 167 | 0E00 | 7FE | 00EB | 7FE | 014F | 014F | 0000 | 7FE | 00EB |
| 167 | 800 | 168 | 800 | 7FE | 00EB | 7FF | 167 | 00EB | 0000 |  |  |
| 168 | 0A00 | 14F | 0A00 | 7FF | 014F | 0000 | 168 | 00EB | 0000 |  |  |
| 14F | E8F4 | 150 | E8F4 | 584 | 00EB | 0000 | FFF4 | 00EB | 0000 | 584 | 00EB |
| 150 | A8F2 | 151 | A8F2 | 583 | 0003 | 0000 | FFF2 | 0003 | 0000 |  |  |
| 151 | 6F01 | 152 | 6F01 | 151 | 0001 | 0000 | 0001 | 0002 | 0001 |  |  |
| 152 | E8F0 | 153 | E8F0 | 583 | 0002 | 0000 | FFF0 | 0002 | 0001 | 583 | 2 |
| 153 | F101 | 155 | F101 | 153 | F101 | 0000 | 0001 | 0002 | 0001 |  |  |
| 155 | A8EE | 156 | A8EE | 584 | 00EB | 0000 | FFEE | 00EB | 0001 |  |  |
| 156 | 680 | 157 | 680 | 156 | 680 | 0000 | 156 | EB00 | 1001 |  |  |
| 157 | 0C00 | 158 | 0C00 | 7FF | EB00 | 7FF | 157 | EB00 | 1001 | 7FF | EB00 |
| 158 | D15E | 15E | D15E | 7FE | 159 | 7FE | D15E | EB00 | 1001 | 7FE | 159 |
| 15E | 1205 | 15F | 1205 | 15E | 1205 | 7FE | 015E | EB40 | 1001 |  |  |
| 15F | 2F40 | 160 | 2F40 | 15F | 40 | 7FE | 40 | 40 | 1 |  |  |
| 160 | F0FD | 161 | F0FD | 160 | F0FD | 7FE | 160 | 40 | 1 |  |  |
| 161 | 1204 | 162 | 1204 | 161 | 1204 | 7FE | 161 | 00D7 | 1 |  |  |
| 162 | 4C01 | 163 | 4C01 | 7FF | EB00 | 7FE | 1 | EBD7 | 1000 |  |  |
| 163 | 0E00 | 164 | 0E00 | 7FE | EBD7 | 7FE | 159 | 159 | 0 | 7FE | EBD7 |
| 164 | EC01 | 165 | EC01 | 7FF | 159 | 7FE | 1 | 159 | 0 | 7FF | 159 |
| 165 | 0E00 | 166 | 0E00 | 7FE | 159 | 7FE | EBD7 | EBD7 | 1000 | 7FE | 159 |
| 166 | 0E00 | 167 | 0E00 | 7FE | EBD7 | 7FE | 159 | 159 | 0 | 7FE | EBD7 |
| 167 | 800 | 168 | 800 | 7FE | EBD7 | 7FF | 167 | EBD7 | 1000 |  |  |
| 168 | 0A00 | 159 | 0A00 | 7FF | 159 | 0 | 168 | EBD7 | 1000 |  |  |
| 159 | 680 | 15A | 680 | 159 | 680 | 0 | 159 | D7EB | 1000 |  |  |
| 15A | EAE9 | 15B | EAE9 | 584 | D7EB | 0 | FFE9 | D7EB | 1000 | 144, 584 | 585, D7EB |
| 15B | 88E7 | 15C | 88E7 | 583 | 1 | 0 | 0 | D7EB | 1000 | 583 | 1 |
| 15C | CEEF | 14C | CEEF | 15C | 014C | 0 | FFEF | D7EB | 1000 |  |  |
| 14C | 200 | 14D | 200 | 14C | 200 | 0 | 014C | 0 | 100 |  |  |
| 14D | 0C00 | 14E | 0C00 | 7FF | 0 | 7FF | 014D | 0 | 100 | 7FF | 0 |
| 14E | D15E | 15E | D15E | 7FE | 014F | 7FE | D15E | 0 | 100 | 7FE | 014F |
| 15E | 1205 | 15F | 1205 | 15E | 1205 | 7FE | 015E | 40 | 100 |  |  |
| 15F | 2F40 | 160 | 2F40 | 15F | 40 | 7FE | 40 | 40 | 0 |  |  |
| 160 | F0FD | 161 | F0FD | 160 | F0FD | 7FE | 160 | 40 | 0 |  |  |
| 161 | 1204 | 162 | 1204 | 161 | 1204 | 7FE | 161 | 3 | 0 |  |  |
| 162 | 4C01 | 163 | 4C01 | 7FF | 0 | 7FE | 1 | 3 | 0 |  |  |
| 163 | 0E00 | 164 | 0E00 | 7FE | 3 | 7FE | 014F | 014F | 0 | 7FE | 3 |
| 164 | EC01 | 165 | EC01 | 7FF | 014F | 7FE | 1 | 014F | 0 | 7FF | 014F |
| 165 | 0E00 | 166 | 0E00 | 7FE | 014F | 7FE | 3 | 3 | 0 | 7FE | 014F |
| 166 | 0E00 | 167 | 0E00 | 7FE | 3 | 7FE | 014F | 014F | 0 | 7FE | 3 |
| 167 | 800 | 168 | 800 | 7FE | 3 | 7FF | 167 | 3 | 0 |  |  |
| 168 | 0A00 | 14F | 0A00 | 7FF | 014F | 0 | 168 | 3 | 0 |  |  |
| 14F | E8F4 | 150 | E8F4 | 585 | 3 | 0 | FFF4 | 3 | 0 | 585 | 3 |
| 150 | A8F2 | 151 | A8F2 | 583 | 1 | 0 | FFF2 | 1 | 0 |  |  |
| 151 | 6F01 | 152 | 6F01 | 151 | 1 | 0 | 1 | 0 | 101 |  |  |
| 152 | E8F0 | 153 | E8F0 | 583 | 0 | 0 | FFF0 | 0 | 101 | 583 | 0 |
| 153 | F101 | 154 | F101 | 153 | F101 | 0 | 153 | 0 | 101 |  |  |
| 154 | CE06 | 15B | CE06 | 154 | 015B | 0 | 6 | 0 | 101 |  |  |
| 15B | 88E7 | 15D | 88E7 | 583 | FFFF | 0 | FFFE | 0 | 101 | 583 | FFFF |
| 15D | 100 | 15E | 100 | 15D | 100 | 0 | 015D | 0 | 101 |  |  |

Доп

ORG 0x0

no: WORD $part2

RES: WORD $B\_0

START: CLA

CALL $input

LD part2

BNE STARTB

CALL $output

JUMP START

STARTB: CLA

CALL $inputB

LD EQ

BNE RESULT

CALL $outputB

JUMP STARTB

RESULT: LD minOrPlus

;-2 - плюс -3- минус -1 - равно

STOP: HLT

;1 число

A\_0: WORD 0x0

A\_1: WORD 0x0

A\_2: WORD 0x0

A\_3: WORD 0x0

A\_4: WORD 0x0

A\_5: WORD 0x0

A\_6: WORD 0x0

A\_7: WORD 0x0

revExpand: LD A\_1

ST A\_0

LD A\_2

ST A\_1

LD A\_3

ST A\_2

LD A\_4

ST A\_3

LD A\_5

ST A\_4

LD A\_6

ST A\_5

LD A\_7

ST A\_6

CLA

ST A\_7

RET

expand: LD A\_6

ST A\_7

LD A\_5

ST A\_6

LD A\_4

ST A\_5

LD A\_3

ST A\_4

LD A\_2

ST A\_3

LD A\_1

ST A\_2

LD A\_0

ST A\_1

CLA

ST A\_0

RET

input: IN 0x1D

AND #0x40

BEQ input

CALL $expand

IN 0x1C

PUSH

CALL $checkInput

BNC das

BEQ das

ADD #0x1

BEQ equals1

ADD #0x1

BEQ plus1

ADD #0x1

JUMP minus1

equals1: CALL $revExpand

LD #0x1

ST EQ

JUMP enddd

plus1: CALL $revExpand

CALL $clearOutput

LD #0x1

ST minOrPlus

ST part2

JUMP enddd

minus1: CALL $revExpand

CALL $clearOutput

LD #0x1

ST part2

JUMP enddd

das:

ST A\_0

enddd: RET

minOrPlus: WORD 0x0

part2: WORD 0x0

EQ: WORD 0x0

; 0 - minus, 1 - plus

;2 число

B\_0: WORD 0x0

B\_1: WORD 0x0

B\_2: WORD 0x0

B\_3: WORD 0x0

B\_4: WORD 0x0

B\_5: WORD 0x0

B\_6: WORD 0x0

B\_7: WORD 0x0

revExpandB: LD B\_1

ST B\_0

LD B\_2

ST B\_1

LD B\_3

ST B\_2

LD B\_4

ST B\_3

LD B\_5

ST B\_4

LD B\_6

ST B\_5

LD B\_7

ST B\_6

CLA

ST B\_7

RET

expandB: LD B\_6

ST B\_7

LD B\_5

ST B\_6

LD B\_4

ST B\_5

LD B\_3

ST B\_4

LD B\_2

ST B\_3

LD B\_1

ST B\_2

LD B\_0

ST B\_1

CLA

ST B\_0

RET

inputB: IN 0x1D

AND #0x40

BEQ inputB

CALL $expandB

IN 0x1C

PUSH

CALL $checkInput

BNC dasB

BEQ dasB

ADD #0x1

BEQ equals1B

ADD #0x1

BEQ plus1

ADD #0x1

JUMP minus1B

equals1B: CALL $revExpandB

ADD #0x1

ST EQ

JUMP endddB

plus1B: CALL $revExpandB

ADD #0x1

CALL $clearOutput

ST minOrPlus

ST part2

JUMP endddB

minus1B: CALL $revExpandB

CALL $clearOutput

ADD #0x1

ST part2

JUMP endddB

dasB:

ST B\_0

endddB: RET

ORG 0x200

tmp: WORD 0x0

en: WORD $A\_7

bool\_out: WORD 0x0

LEN: WORD 0x70

output: CLA

PUSH

PUSH

PUSH

LD bool\_out

ST &2

LD LEN

ST &0

LD en

ST &1

spinloop: IN 0x15

AND #0x40

BEQ spinloop

;проверка на ноль

LD &1

ST tmp

LD (tmp)

BNE true

JUMP check

true: LD #0x1

ST &2

check: LD &2

BNE continue

LD #0xB

JUMP ou

continue: LD &1

ST tmp

LD (tmp)

ou: ADD &0

OUT 0x14

LD &1

SUB #0x1

ST &1

LD &0

SUB #0x10

ST &0

BPL spinloop

SWAP

POP

SWAP

POP

SWAP

POP

RET

tmpB: WORD 0x0

enB: WORD $B\_7

bool\_outB: WORD 0x0

LENB: WORD 0x70

outputB: CLA

PUSH

PUSH

PUSH

LD bool\_outB

ST &2

LD LENB

ST &0

LD enB

ST &1

spinloopB: IN 0x15

AND #0x40

BEQ spinloopB

;проверка на ноль

LD &1

ST tmpB

LD (tmpB)

BNE trueB

JUMP checkB

trueB: LD #0x1

ST &2

checkB: LD &2

BNE continueB

LD #0xB

JUMP ouB

continueB: LD &1

ST tmpB

LD (tmpB)

ouB: ADD &0

OUT 0x14

LD &1

SUB #0x1

ST &1

LD &0

SUB #0x10

ST &0

BPL spinloopB

SWAP

POP

SWAP

POP

SWAP

POP

RET

clearNum: WORD 0x8B

clearOutput: CLA

PUSH

LD #0x9

ST &0

out1: IN 0x15

AND #0x40

BEQ out1

LD clearNum

OUT 0x14

SUB #0x10

ST clearNum

LOOP &0

JUMP out1

SWAP

POP

LD #0x8B

ST clearNum

RET

checkInput: LD &1

SUB #0xA

BEQ minus

ADD #0xA

SUB #0xB

BEQ plus

ADD #0xB

SUB #0xF

BEQ equals

ADD #0xF

JUMP endd

minus: SUB #0x3

JUMP endd

plus: SUB #0x2

JUMP endd

equals: SUB #0x1

endd: SWAP

ST &1

SWAP

SWAP

POP

RET

# Вывод

В данной лабораторной работе я изучил организацию системы ввода-вывода базовой ЭВМ, команды ввода-вывода и исследовал процесс функционирования ЭВМ при обмене данными по сигналам готовности внешних устройств