Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

**Лабораторная работа № 5**

**«Асинхронный обмен данными с ВУ»**

По дисциплине «Основы профессиональной деятельности»

Вариант 1756

Выполнила:

Студентка группы P3117

Русакова Е.Д.

Преподаватель:

Ткешелашвили Н.М.

Санкт-Петербург

2023

Оглавление

[Задание: 3](#_Toc132668161)

[Выполнение задания: 4](#_Toc132668162)

[1 - Расшифровка исходного кода: 4](#_Toc132668163)

[Основная программа: 4](#_Toc132668164)

[2 - Описание работы программы: 4](#_Toc132668165)

[Трассировка: 5](#_Toc132668166)

# 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.

Здание для варианта 1756:

1. грамма осуществляет асинхронный ввод данных с ВУ-3
2. Программа начинается с адреса 4DF16. Размещаемая строка находится по адресу 60316.
3. Строка должна быть представлена в кодировке ISO-8859-5.
4. Формат представления строки в памяти: АДР1: СИМВ2 СИМВ1 АДР2: СИМВ4 СИМВ3 ... СТОП\_СИМВ.
5. Ввод или вывод строки должен быть завершен по символу c кодом 0A (NL). Стоп символ является обычным символом строки и подчиняется тем же правилам расположения в памяти что и другие символы строки.

# Выполнение задания:

## 1 - Расшифровка исходного кода:

### Основная программа:

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарий |
| 4DC | 000A |  | Стоп символ |
| 4DD | 0603 |  | Указатель на текущую ячейку |
| 4DE | 0000 |  | Вспомогательная ячейка |
| 4DF | 0200 | CLA | 0 -> AC |
| 4E0 | 1207 | IN 7 | Чтение регистра ВУ – 3, проверка готовности |
| 4E1 | 2F40 | AND #40 | AC AND 40 |
| 4E2 | F0FD | BEQ (IP-3) | Переход к 4E0, если Z==1 |
| 4E3 | 1206 | IN 6 | Чтение данных ВУ – 3, ввод первого символа |
| 4E4 | 0680 | SWAB | Обмен мл. и ст. байт |
| 4E5 | E8F7 | ST (IP-9) | Сохранение символа в ячейку (4DD) |
| 4E6 | 0680 | SWAB | Обмен мл. и ст. байт |
| 4E7 | 74DC | CMP 4DC | Установление флагов от AC-4DC |
| 4E8 | F00C | BEQ (IP+12) | Переход к 4F5, если Z==1 |
| 4E9 | 0200 | CLA | 0 -> AC |
| 4EA | 1207 | IN 7 | Чтение регистра ВУ – 3, проверка готовности |
| 4EB | 2F40 | AND #40 | AC AND 40 |
| 4EC | FOFD | BEQ (IP-3) | Переход к 4EA, если Z==1 |
| 4ED | 1206 | IN 6 | Чтение данных ВУ – 3, ввод второго символа |
| 4EE | E4DE | ST 4DE | AC -> 4DE |
| 4EF | 48ED | ADD (IP-19) | AC + (4DD) -> AC |
| 4F0 | EAEC | ST (IP-20)+ | Сохранение в ячейку (4DD)+ |
| 4F1 | B4DE | SWAM 4DE | AC <-> 4DE |
| 4F2 | 74DC | CMP 4DC | Установление флагов от AC-4DC |
| 4F3 | F001 | BEQ (IP+1) | Переход к 4F5, если Z==1 |
| 4F4 | C4DF | JUMP 4DF | Возврат в начало цикла |
| 4F5 | 0100 | HLT | Остановка |

## 2 - Описание работы программы:

Программа производит асинхронный ввод данных с ВУ-3

**Расположение в памяти:**

Основная программа – 4DF-4F5

Результат – 603-…

Стоп символ – 4DC

**Код программы на ассемблере БЭВМ:**

**ORG 4DC**

**STOP: WORD 000A**

**ARR: WORD 0x603**

**HELP: WORD 0000**

**BEGIN: CLA**

**S1: IN 7**

**AND #0x40**

**BEQ S1**

**IN 6**

**SWAB**

**ST ARR**

**SWAB**

**CMP STOP**

**BEQ FINISH**

**CLA**

**S2: IN 7**

**AND #0x40**

**BEQ S2**

**IN 6**

**ST HELP**

**ADD (ARR)**

**ST (ARR)+**

**SWAM HELP**

**CMP STOP**

**BEQ FINISH**

**JUMP BEGIN**

**FINISH: HLT**

## 

## Трассировка:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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** | **Адрес** | **Новый код** |
| **xxx** | **xxxx** | **xxx** | **xxxx** | **xxxx** | **xxxx** | **xxx** | **xxxx** | **xxxx** | **xxx** | **xxxx** | **xxx** | **xxxx** |
| **4DF** | **0200** | 4E0 | 200 | 4DF | 0200 | 000 | 04DF | 0000 | 004 | 0100 |  |  |
| **4E0** | **1207** | 4E1 | 1207 | 4E0 | 1207 | 000 | 04E0 | 0040 | 004 | 0100 |  |  |
| **4E1** | **2F40** | 4E2 | 2F40 | 4E1 | 0040 | 000 | 0040 | 0040 | 000 | 0000 |  |  |
| **4E2** | **F0FD** | 4E3 | F0FD | 4E2 | F0FD | 000 | 04E2 | 0040 | 000 | 0000 |  |  |
| **4E3** | **1206** | 4E4 | 1206 | 4E3 | 1206 | 000 | 04E3 | 00C1 | 000 | 0000 |  |  |
| **4E4** | **E8F8** | 4E5 | E8F8 | 604 | 00C1 | 000 | FFF8 | 00C1 | 000 | 0000 | 604 | 00C1 |
| **4E5** | **7EF6** | 4E6 | 7EF6 | 4DC | 000A | 000 | FFF6 | 00C1 | 001 | 0001 |  |  |
| **4E6** | **F00D** | 4E7 | F00D | 4E6 | F00D | 000 | 04E6 | 00C1 | 001 | 0001 |  |  |
| **4E7** | **0200** | 4E8 | 0200 | 4E7 | 0200 | 000 | 04E7 | 0000 | 005 | 0101 |  |  |
| **4E8** | **1207** | 4E9 | 1207 | 4E8 | 1207 | 000 | 04E8 | 0000 | 005 | 0101 |  |  |
| **4E9** | **2F40** | 4EA | 2F40 | 4E9 | 040 | 000 | 0040 | 0000 | 005 | 0101 |  |  |
| **4EA** | **F0FD** | 4E8 | F0FD | 4EA | F0FD | 000 | FFFD | 0000 | 005 | 0101 |  |  |
| **4EB** | **1206** | 4EC | 1206 | 4EB | 1206 | 000 | 04EB | 00C2 | 001 | 0001 |  |  |
| **4EC** | **EEF1** | 4ED | EEF1 | 4DE | 00C2 | 000 | FFF1 | 00C2 | 001 | 0001 | 4DE | 00C2 |
| **4ED** | **0680** | 4EE | 0680 | 4ED | 0680 | 000 | 04ED | C200 | 009 | 1001 |  |  |
| **4EE** | **48EE** | 4EF | 48EE | 604 | 00C1 | 000 | FFEE | C2C1 | 008 | 1000 |  |  |
| **4EF** | **EAED** | 4F0 | EAED | 604 | C2C1 | 000 | FFED | C2C1 | 008 | 1000 | 4DD | 605 |
|  |  |  |  |  |  |  |  |  |  |  | 604 | C2C1 |
| **4F0** | **BEED** | 4F1 | BEED | 4DE | C2C1 | 000 | 00C2 | 00C2 | 000 | 0000 | 4DE | C2C1 |
| **4F1** | **7EEA** | 4F2 | 7EEA | 4DC | 000A | 000 | FFEA | 00C2 | 001 | 0001 |  |  |
| **4F2** | **F001** | 4F3 | F001 | 4F2 | F001 | 000 | 04F2 | 00C2 | 001 | 0001 |  |  |
| **4F3** | **CEEB** | 4DF | CEEB | 4F3 | 04DF | 000 | FFEB | 00C2 | 001 | 0001 |  |  |
| **4DF** | **0200** | 4E0 | 0200 | 4DF | 0200 | 000 | 04DF | 0000 | 005 | 0101 |  |  |

## Дополнительное задание:

Задание:

С ВУ-3 вводится 16-разрядное число (в два захода, сначала старшая часть, затем младшая). Интерпретируя это число, как количество секунд, вывести на ВУ-6 (бегущая строка) строку в формате "1:30:12", где три числа это часы, минуты и секунды соответственно.

Код на ассемблере БЭВМ:

ORG 0x500 ;коды цифр для бегущей строки

N: WORD 0x24, 0x00

N0: WORD 0x3C, 0x42, 0x42, 0x3C, 0x00

N1: WORD 0x10, 0x22, 0x7E, 0x02, 0x00

N2: WORD 0x32, 0x46, 0x4A, 0x32, 0x00

N3: WORD 0x24, 0x42, 0x5A, 0x24, 0x00

N4: WORD 0x78, 0x08, 0x08, 0x7E, 0x00

N5: WORD 0x74, 0x52, 0x52, 0x4C, 0x00

N6: WORD 0x3C, 0x52, 0x52, 0x0C, 0x00

N7: WORD 0x40, 0x48, 0x7E, 0x08, 0x00

N8: WORD 0x2C, 0x52, 0x52, 0x2C, 0x00

N9: WORD 0x32, 0x4A, 0x4A, 0x3C, 0x00

SN: WORD 0x500 ;позиция начала двоеточия

SN0: WORD 0x502 ;позиция начала цифры 0

SN1: WORD 0x507 ;позиция начала цифры 1

SN2: WORD 0x50C ;позиция начала цифры 2

SN3: WORD 0x511 ;позиция начала цифры 3

SN4: WORD 0x516 ;позиция начала цифры 4

SN5: WORD 0x51B ;позиция начала цифры 5

SN6: WORD 0x520 ;позиция начала цифры 6

SN7: WORD 0x525 ;позиция начала цифры 7

SN8: WORD 0x52A ;позиция начала цифры 8

SN9: WORD 0x52F ;позиция начала цифры 9

ORG 0x600

TOT: WORD 0000 ;введенное число

TIME: WORD 0000 ;оставшееся кол-во секунд

TT: WORD 0000

HOURS: WORD 0000 ;кол-во часов

MINS: WORD 0000 ;кол-во минут

SEC: WORD 0000 ;кол-во секунд

MIN: WORD 003C ;в минуте 60 секунд

HOUR: WORD 0E10 ;в минуте 3600 секунд

POS1: WORD ? ;позиция начала символа

;обнуление счетчиков

BEGIN: CLA

ST HOURS

ST MINS

ST SEC

;ввод старшей части числа

S1: IN 0x07

AND #0x40

BEQ S1

IN 0x06

SWAB

ST TIME

;ввод младшей части числа

S2: IN 0x07

AND #0x40

BEQ S2

IN 0x06

ADD TIME

ST TIME

ST TOT

;проверка на больше число - чтобы бэвм не парсила его как отрицательное

COR: LD TIME

BPL CALC

ROL

CLC

ROR

ST TIME

LD #0x9

ST HOURS

LD #0x6

ST MINS

LD #0x8

ST SEC

JUMP CALC

;начало расчета

CALC: LD TIME

BEQ COR2

CMP HOUR

BGE HC

CMP MIN

BGE MC

JUMP SC

;считает кол-во часов

HC: ADD (HOURS)+

LD TIME

SUB HOUR

ST TIME

JUMP CALC

;считает кол-во минут

MC: ADD (MINS)+

LD TIME

SUB MIN

ST TIME

JUMP CALC

;считает кол-во секунд

SC: LD TIME

ADD SEC

ST SEC

;проверяет корректность значения секунд

COR2: LD SEC

BEQ COR3

CMP MIN

BLT COR3

SUB MIN

ST SEC

ADD (MINS)+

;проверяет корректность значения минут

COR3: LD MINS

BEQ OUTPUT

CMP MIN

BLT OUTPUT

SUB MIN

ST MINS

ADD (HOURS)+

;начало вывода символов

OUTPUT:

LD #0x05

ST $COUNTER

VEH: IN 0x11

AND #0x40

BEQ VEH

LD #0x00

OUT 0x10

LOOP $COUNTER

JUMP VEH

;вывод часов

OHOUR: LD HOURS

PUSH

CALL $TRANS

LD $B1

PUSH

CALL $CHOICE

LD $B2

PUSH

CALL $CHOICE

; вывод двоеточия

O: LD $SN

ST $POS1

LD #0x02

ST $COUNTER

OU: IN 0x11

AND #0x40

BEQ OU

LD (POS1)+

OUT 0x10

LOOP $COUNTER

JUMP OU

;вывод минут

OMINS: LD MINS

PUSH

CALL $TRANS

LD $B1

PUSH

CALL $CHOICE

LD $B2

PUSH

CALL $CHOICE

; вывод двоеточия

O1: LD $SN

ST $POS1

LD #0x02

ST $COUNTER

OU1: IN 0x11

AND #0x40

BEQ OU1

LD (POS1)+

OUT 0x10

LOOP $COUNTER

JUMP OU1

;вывод секунд

OSEC: LD SEC

PUSH

CALL $TRANS

LD $B1

PUSH

CALL $CHOICE

LD $B2

PUSH

CALL $CHOICE

FINISH: HLT

;вывод символа

ORG 0x200

COUNTER: WORD ? ;счетчик

POS: WORD ? ;позиция начала символа

;выбор нужной картинки

CHOICE: LD &0x01

BEQ CH0

CMP #0x01

BEQ CH1

CMP #0x02

BEQ CH2

CMP #0x03

BEQ CH3

CMP #0x04

BEQ CH4

CMP #0x05

BEQ CH5

CMP #0x06

BEQ CH6

CMP #0x07

BEQ CH7

CMP #0x08

BEQ CH8

;загрузка позиции начала нужной картинки

LD $SN9

ST $POS

JUMP OSIM

CH0: LD $SN0

ST $POS

JUMP OSIM

CH1: LD $SN1

ST $POS

JUMP OSIM

CH2: LD $SN2

ST $POS

JUMP OSIM

CH3: LD $SN3

ST $POS

JUMP OSIM

CH4: LD $SN4

ST $POS

JUMP OSIM

CH5: LD $SN5

ST $POS

JUMP OSIM

CH6: LD $SN6

ST $POS

JUMP OSIM

CH7: LD $SN7

ST $POS

JUMP OSIM

CH8: LD $SN8

ST $POS

;загрузка размера картинки в счетчик

OSIM: LD #0x05

ST $COUNTER

;вывод картинки на бегущую строку

OSIM1: IN 0x11

AND #0x40

BEQ OSIM1

LD (POS)+

OUT 0x10

LOOP $COUNTER

JUMP OSIM1

RET

;перевод в 10-ную систему счисления

ORG 0x300

TRANS: LD &0x01

ST NUM

CLA

ST B1

ST B2

;выделяем 1 символ

LD NUM

CLC

ROR

CLC

ROR

CLC

ROR

CLC

ROR

ST A1

;выделяем 2 символ

LD NUM

SWAB

CLC

ROL

CLC

ROL

CLC

ROL

CLC

ROL

CLC

ROL

ROL

ROL

ROL

ROL

ST A2

STEP1: CMP E10

BLT STEP2

SUB E10

ST B2

LD #0x01

ST B1

JUMP STEP3

STEP2: ST B2

STEP3: LD A1

BEQ CHECK

CMP #0x01

BEQ STEP4

CMP #0x02

BEQ STEP5

CMP #0x03

BEQ STEP6

STEP4: ADD (B1)+

LD B2

ADD #0x06

ST B2

JUMP CHECK

STEP5: LD B1

ADD #0x03

ST B1

LD B2

ADD #0x02

ST B2

JUMP CHECK

STEP6: LD B1

ADD #0x04

ST B1

LD B2

ADD #0x08

ST B2

CHECK: LD B2

CMP E10

BLT SAVE

SUB E10

ST B2

ADD (B1)+

SAVE: RET

E10: WORD 0x0A ;10 для перевода

NUM: WORD ? ;переводимое число

A1: WORD ? ;1 символ в 16-ной

A2: WORD ? ;2 символ в 16-ной

B1: WORD ? ;1 символ в 10-ной

B2: WORD ? ;2 символ в 10-ной