МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

Национальный исследовательский университет ИТМО

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ**

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ

Вариант: 3255

Выполнил:

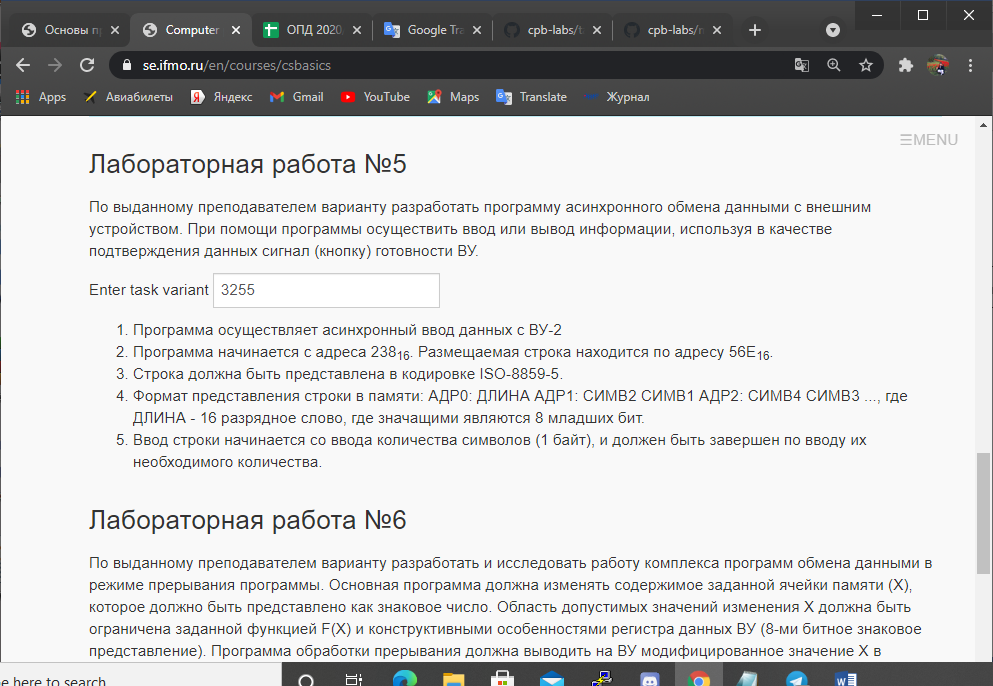
Студент группы P3132

Вали Насибулла

Преподаватель:

Санкт-Петербург, 2021 

**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.

**Текст программы:**

ORG 0x238

DAT: WORD $LEN

ADDR: WORD 0

NOW: WORD 0

START: CLA

LD DAT

ST ADDR

FIRST\_1:

IN 5

AND #0x40

BEQ FIRST\_1

IN 4

BEQ STOP

ST NOW

ST (ADDR)+

SECOND\_2:

CLA

IN 5

AND #0x40

BEQ SECOND\_2

IN 4

ST (ADDR)

LOOP NOW

JUMP THIRD\_3

JUMP STOP

THIRD\_3: CLA

IN 5

AND #0x40

BEQ THIRD\_3

IN 4

SWAB

OR (ADDR)

ST (ADDR)+

LOOP NOW

JUMP SECOND\_2

STOP: HLT

ORG 0x56E

LEN: WORD ?

**Вводимая строка:**

|  |  |  |  |
| --- | --- | --- | --- |
| № | ISO-8859-5 | UTF-8 | UTF-16 |
| S | 53 | 53 | 83 |
| K | 4B | 4b | 75 |
| Y | 59 | 59 | 89 |

**Описание программы:**

**Назначение программы:**

Программа реализует посимвольный асинхронный ввод с ВУ-2 в кодировке ISO8859-5. В 16-битной ячейке памяти БЭВМ размещается два 8-битных символа, начиная с ячейки 0x238. Цикл ввода продолжается до тех пор, пока не будет введен символ LEN **.**

**Область представления и область допустимых значений данных**

**Область представления данных**

Ячейки ADDR,START, NOW, LEN: 16-разрядные беззнаковые целые числа

Ячейки с введенной строкой: 16-разрядные беззнаковые целые числа

**Область допустимых значений данных**

Длина вводимой строки: 0……256

**Расположение в памяти ЭВМ**

Программа: 238…..255

Адрес ячейки первого символа строки: 238(START)

Адрес текущей ячейки записи символов: 239 (ADDR)

Сколько символов осталось записать: 23A (NOW)

**Адреса первой и последней выполняемой команд программы**

Адрес первой команды программы: 238

Адрес последней команды программы: 256

**Таблица трассировки**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Адрес | Код | IP | CR | AR | DR | SP | BR | AC | PS | NZVC | Адрес | Новое значение |
| 23B | 0000 | 23B | 0000 | 0000 | 0000 | 000 | 0000 | 0000 | 000 | 0000 | 0000 |  |
| 23B | 0200 | 23C | 0200 | 23B | 0200 | 000 | 023B | 0000 | 004 | 0100 |  |  |
| **23C** | AEFB | 23D | AEFB | 238 | 056E | 000 | FFFB | 056E | 000 | 0000 |  |  |
| 23D | EEFB | 23E | EEFB | 239 | 056E | 000 | FFFB | 056E | 000 | 0000 | 239 |  |
| 23E | 1205 | 23F | 1205 | 23E | 1205 | 000 | 023E | 0540 | 000 | 0000 |  |  |
| 23F | 2F40 | 240 | 2F40 | 23F | 0040 | 000 | 0040 | 0040 | 000 | 0000 |  |  |
| 240 | F0FD | 241 | F0FD | 240 | F0FD | 000 | 0240 | 0040 | 000 | 0000 |  |  |
| **241** | 1204 | 242 | 1204 | 241 | 1204 | 000 | 0241 | 0003 | 000 | 0000 |  |  |
| 242 | F015 | 243 | F015 | 242 | F015 | 000 | 0242 | 0003 | 000 | 0000 |  |  |
| **243** | EEF6 | 244 | EEF6 | 23A | 0003 | 000 | FFF6 | 0003 | 000 | 0000 | 23A | 0003 |
| **244** | **EAF4** | **245** | **EAF4** | **56E** | **0003** | 000 | FFF4 | 0003 | 000 |  |  |  |
| 245 | 0200 | 246 | 0200 | 245 | 0200 | 000 | 0245 | 0000 | 004 | 0100 |  |  |
| 246 | 1205 | 247 | 1205 | 246 | 1205 | 000 | 0246 | 0040 | 004 | 0100 |  |  |
| 247 | 2F40 | 248 | 2F40 | 247 | 0040 | 000 | 0040 | 0040 | 000 | 0000 |  |  |
| 248 | F0FC | 249 | F0FC | 248 | F0FC | 000 | 0248 | 0040 | 000 | 0000 |  |  |
| **249** | 1204 | 24A | 1204 | 249 | 1204 | 000 | **0249** | 0053 | 000 | 0000 |  |  |
| 24A | E8EF | 24B | E8EF | 56F | 0053 | 000 | FFEF | 0053 | 000 | 0000 | 56E | 0053 |
| 24B | 8EEF | 24C | 8EEF | 23A | 0002 | 000 | 0001 | 0053 | 000 | 0000 |  |  |
| 24C | CE01 | 24E | CE01 | 24C | 024E | 000 | 0001 | 0053 | 000 | 0000 |  |  |
| 24E | 0200 | 24F | 0200 | 24E | 0200 | 000 | 024E | 0000 | 000 | 0100 |  |  |
| 24F | 1205 | 250 | 1205 | 24F | 1205 | 000 | 024F | 0040 | 000 | 0100 |  |  |
| **250** | 2F40 | 251 | 2F40 | 250 | 0040 | 000 | 0040 | 0040 | 000 | 0000 |  |  |
| 251 | F0FC | 252 | F0FC | 251 | F0FC | 000 | 0251 | 0040 | 000 | 0000 |  |  |
| 252 | 1204 | 253 | 1204 | 252 | 1204 | 000 | 0251 | 004B | 000 | 0000 |  |  |
| **252** | 0680 | 254 | 0680 | 253 | 0680 | 000 | 0253 | 4B00 | 000 | 0000 |  |  |
| 254 | 38E5 | 255 | 38E5 | 56F | 0053 | 000 | B4AC | 4B53 | 000 | 0000 |  |  |
| 255 | EAE3 | 256 | EAE3 | 56F | 4B53 | 000 | FFE3 | 4B53 | 000 | 0000 | 56E | 4B53 |
| 256 | 8EE3 | 257 | 8EE3 | 23A | 0001 | 000 | 0000 | 4B53 | 000 | 0000 |  |  |
| 257 | CEED | 245 | CEED | 257 | 0245 | 000 | FFED | 4B53 | 000 | 0000 |  |  |
| **245** | 0200 | 246 | 0200 | 245 | 0200 | 000 | 0245 | 0000 | 000 | 0100 |  |  |
| 246 | 1205 | 247 | 1205 | 246 | 1205 | 000 | 0246 | 0040 | 000 | 0100 |  |  |
| 247 | 2F40 | 248 | 2F40 | 247 | 0040 | 000 | 0040 | 0040 | 000 | 0000 |  |  |
| 248 | F0FC | 249 | FOFC | 248 | F0FC | 000 | 0248 | 0040 | 000 | 0000 |  |  |
| 249 | 1204 | 24A | 1204 | 249 | 1204 | 000 | 0249 | 0059 | 000 | 0000 |  |  |
| 24A | E8EF | 24B | E8EF | 570 | 0059 | 000 | FFEF | 0059 | 000 | 0000 | 56F | 0059 |
| 24B | 8EEF | 24D | 8EEF | 23A | 0000 | 000 | FFFF | 0059 | 000 | 0000 |  |  |
| 24D | CE0A | 258 | CE0A | 24D | 0258 | 000 | 000A | 0059 | 000 | 0000 |  |  |
| 258 | 0100 | 259 | 0100 | 258 | 0100 | 000 | 0258 | 0059 | 000 | 0000 |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

**EXTRA TASK:**

Таймер. Если на ВУ-3 положительное число, таймер "тикает", если отрицательное - останавливается. Вывод цифр на ВУ-7, используем только младшие три разряда, после значения 999 - таймер сбрасывается на 000.

ORG 0x20

NUMS: WORD 0

TENS: WORD 0

HUNDREDS: WORD 0

STATE: WORD 0

START: CLA

JUMP ERISE

SPIN: IN 7

AND #0x40

BEQ CMP\_ST

IN 6

ST STATE

CMP\_ST: LD STATE

AND #0x80

BEQ INC\_N

JUMP SHOW

INC\_N: LD NUMS

INC

CMP #0xA

BEQ INC\_T

ST NUMS

JUMP SHOW

INC\_T: CLA

ST NUMS

LD TENS

INC

CMP #0xA

BEQ INC\_H

ST TENS

JUMP SHOW

INC\_H: CLA

ST TENS

LD HUNDREDS

INC

CMP #0xA

BEQ ERISE

ST HUNDREDS

JUMP SHOW

ERISE: CLA

ST NUMS

ST TENS

ST HUNDREDS

JUMP SHOW

SHOW: CLA

LD NUMS

OUT 0x14

CLA

LD TENS

OR #0x10

OUT 0x14

CLA

LD HUNDREDS

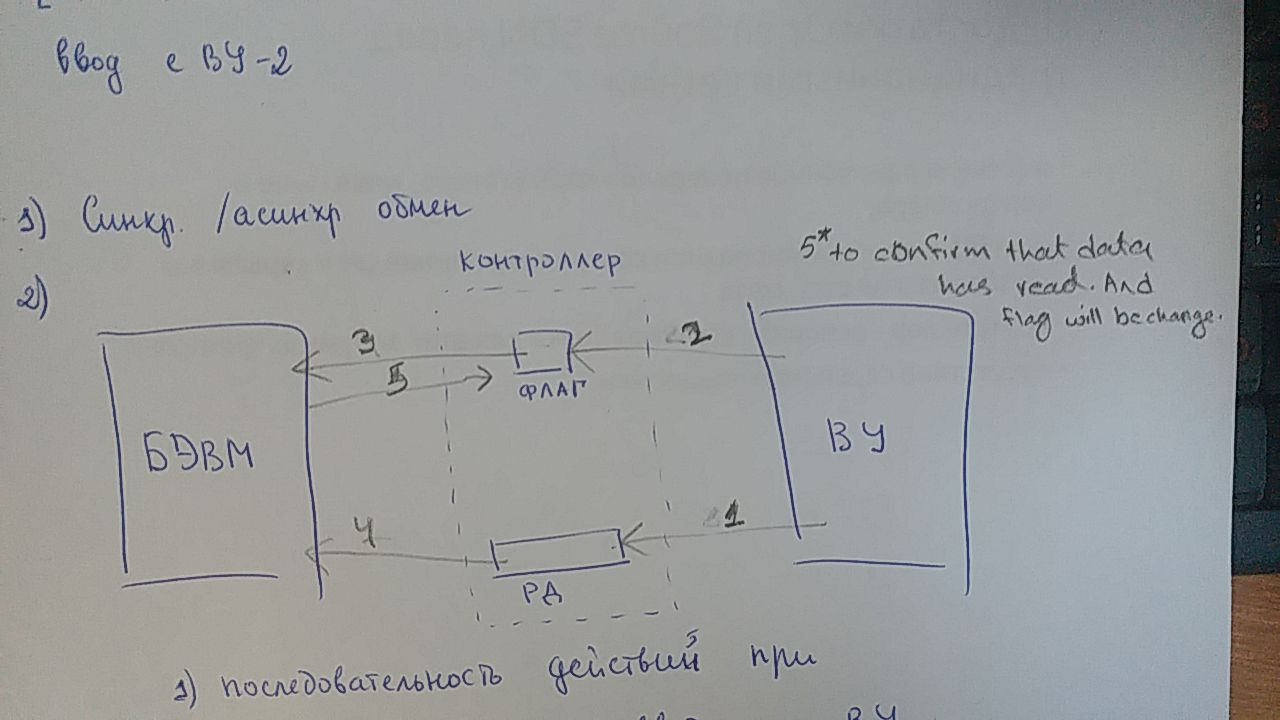
OR #0x20

OUT 0x14

JUMP SPIN

**Defending questions:**

**1-Synchronous and asynchronous transition?**

**2-**

**3- program is chat:**

ADDR: word 0x00

LD #0xFF

ADD #0xFF

AND #0x00

IN 0x6

BEQ EXIT

ADD (ADDR)+

EXIT: HLT

* It loads FFFF to AC, then adds FFFF to it
* Then makes it 0000, using AND
* And setting flags by this command too
* Then it gets the 6th register of ВУ and stops because BEQ works if Z is set (it is set by AND, and IN does not touch flags)
* So basically you only get IN 6
* And it actually do **Nothing**.

**Вывод**

После выполнения этой операции мы знаем, как внешние устройства работают с INPUT. и OUTPUT. Кроме того, второй использовался для написания предварительно собранного кода.