Национальная научно-образовательная корпорация ИТМО

Факультет программной инженерии и компьютерной техники

ЛАБОРАТОРНАЯ РАБОТА №5

по дисциплине

«ОСНОВЫ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ»

Вариант № 1151

Выполнил:

Студент группы P3111

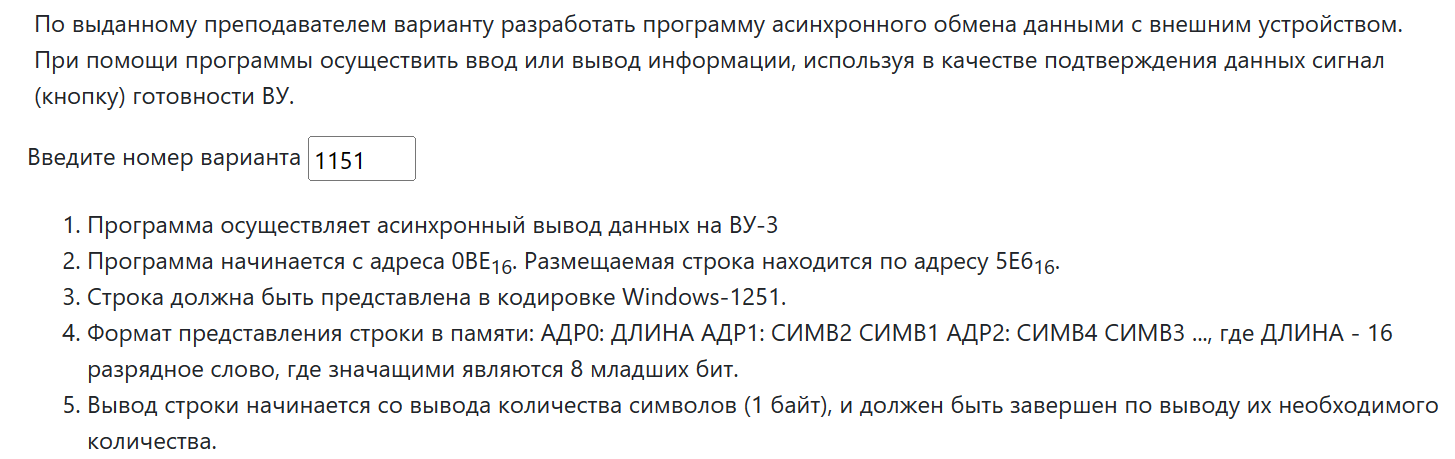
Наземцев Сергей Дмитриевич

Преподаватель:

Ткешелашвили Нино Мерабиевна

Санкт-Петербург, 2023

# Задание



# Исходная программа

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код команды | Мнемоника | Комментарий |
| 0BE | 05E6 | res | Ссылка на ячейку данных |
| 0BF | 00XX | count | Оставшееся количество символов |
| 0C0 | +0200 | CLA | Очистка аккумулятора |
| 0C1 | AEFD | LD (IP – 3) | Загружаем ячейку count |
| 0C2 | 1207 | IN 7 | Ожидание действия |
| 0C3 | 2F40 | AND #040 |  |
| 0C4 | F0FD | BEQ IP-3 |  |
| 0C5 | 1306 | OUT 6 | Вывод длины слов на ВУ-3 |
| 0C6 | 1207 | IN 7 | Ожидание действия |
| 0C7 | 2F40 | AND #040 | Проверка сигнала готовности |
| 0C8 | F0FD | BEQ IP-3 | Если нет сигнала, то спин луп |
| 0C9 | AEF7 | LD (IP – 9) | Загружаем младшие 8 бит первого слова |
| 0CA | 1306 | OUT 6 | Выводим их на ВУ-3 |
| 0CB | AEF6 | LD (IP – 10) | Загружаем переменную count в аккумулятор |
| 0CC | 0740 | DEC | Уменьшаем count на 1 |
| 0CD | EEF4 | ST (IP – 12) | Сохраняем переменную count |
| 0CE | 7F00 | CMP #0 | Сравниваем count с нулем |
| 0CF | F00C | BEQ (IP+12) | Если сount =0, переходим к остановке |
| 0D0 | 1207 | IN 7 | Ожидание действия |
| 0D1 | 2F40 | AND #040 | Проверка сигнала готовности |
| 0D2 | F0FD | BEQ IP-3 | Если нет сигнала, то спин луп |
| 0D3 | AAED | LD (IP – 19) + | Загружаем вторые 8 бит слова и делаем инкремент в следующую ячейку |
| 0D4 | 0680 | SWAB | Обмен байтов слова |
| 0D5 | 1306 | OUT 6 | Выводим на ВУ-3 |
| 0D6 | AEEA | LD (IP – 22) | Выгружаем значение count |
| 0D7 | 0740 | DEC | Уменьшаем count на 1 |
| 0D8 | EEE8 | ST (IP – 24) | Сохраняем значение count |
| 0D9 | 7F00 | CMP #0 | Сравниваем count с нулем |
| 0DA | F00C | BEQ (IP+1) | Если сount =0, переходим к остановке |
| 0DB | CEEA | JUMP (IP – 22) | Возвращаемся к чтению младшего байта слова |
| 0DC | AEE1 | LD (IP – 31) | Выгружаем адрес |
| 0DD | 6F02 | SUB #2 | Сдвигаем адрес в исходное положение |
| 0DE | EEDF | ST (IP – 33) | Сохраняем изначальное значение адреса |
| 0DF | AEDF | LD (IP – 33) | Выгружаем значение count |
| 0E0 | 4F04 | ADD #4 | Возвращаем count в исходное состояние |
| 0E1 | EEDD | ST (IP – 35) | Сохраняем count |
| 0E2 | 0100 | HLT | Остановка |
| … | | | |
| 5E6 | D0CA |  | Ячейка данных |
| 5E7 | C9C0 |  | Ячейка данных |

# Код программы на Ассемблере

ORG 0x0BE

ADDR: WORD $STRING

count: WORD 4

START:

CLA

IN 7

AND #0x40

BEQ START

LD count

OUT 6

S1: IN 7

AND #0x40

BEQ S1

LD (ADDR)

OUT 6

LD count

DEC

ST count

CMP #0

BEQ STOP\_POINT

S2: IN 7

AND #0x40

BEQ S2

LD (ADDR)+

SWAB

OUT 6

LD count

DEC

ST count

CMP #0

BEQ STOP\_POINT

JUMP S1

STOP\_POINT:

LD ADDR

SUB #2

ST ADDR

LD count

ADD #4

ST count

HLT

ORG 0x5E6

STRING:

WORD 0xD0CA

WORD 0xC9C0

Слово КРАЙ в Windows-1251: 0xCA, 0xD0, 0xC0, 0xC9

Слово КРАЙ в UTF-8: D09A, D0A0, D090, D099

Слово в UTF-16: 001A, 0020, 0010, 0011

# Описание программы

Программа осуществляет посимвольный асинхронный вывод данных из памяти на ВУ-3. Программа будет считывать символы из памяти до тех пор, пока не будет достигнута общая длина строк, которую мы указываем заранее. Тогда программа прекратит свое выполнение.

# Область представления

res – 11-разрядная ячейка со ссылкой на результат.

count – 16-разрядная ячейка, у которой значащие младшие 8 бит.

5E6 - ? – 16-разрядные ячейки, хранящие в себе по два символа в кодировке Windows-1251.

# Расположение в памяти данных

0C0 – 0D9 – команды

5E6 - ? – исходные данные

# Область допустимых значений

res ∈ [0x5E6; 0x7FF]

Хранимый в памяти символ: [00; FF]

Адрес первого элемента массива равен 0x5E6 по условию. Т.к. 2047 – 1510 = 537 – количество ячеек, которые могут использоваться для хранения результата => 537\*2 = 1047 – максимально возможное количество хранимых символов (т.к. в данной кодировке символ занимает 1 байт) => Кол-во хранимых символов ∈ [1;1047].

Код для допа:

ORG 0x000

count: WORD 0

seconds: WORD 0

seconds\_ones: WORD 0

minutes: WORD 0

;zero\_const: WORD 0

letter\_ch: WORD 0xDE

letter\_m: WORD 0xCD

letter\_s: WORD 0xD3

space: WORD 0x9A

brake\_letter: WORD 0xF9

START: CLA

IN 7

AND #0x40

BEQ START

IN 6

ST count

; Count minutes

COUNT\_MINUTES:

LD count

SUB #0x3C

BMI COUNT\_SECONDS

ST count

LD minutes

INC

ST minutes

JUMP COUNT\_MINUTES

COUNT\_SECONDS:

LD count

SUB #0x0A

BMI COUNT\_ONES

ST count

LD seconds

INC

ST seconds

JUMP COUNT\_SECONDS

COUNT\_ONES:

LD count

DEC

BMI PRINT\_MINUTES

ST count

LD seconds\_ones

INC

ST seconds\_ones

JUMP COUNT\_ONES

;PRINT\_HOURS:

; Print hours (always 0)

;LD zero\_const

;ADD #0x30

;OUT 0x0C

;LD space

;OUT 0xC0

;LD letter\_ch

;OUT 0x0C

; Print minutes

PRINT\_MINUTES:

LD 0

OUT 0x0C

LD minutes

BEQ PRINT\_TENS

ADD #0x30

OUT 0x0C

LD letter\_m

OUT 0x0C

; Print tens seconds

PRINT\_TENS:

LD space

OUT 0x0C

LD seconds

BEQ PRINT\_ONES

ADD #0x30

OUT 0x0C

;print ones

PRINT\_ONES:

LD seconds\_ones

BEQ BRAKE\_POINT

ADD #0x30

OUT 0x0C

STOP:

LD letter\_s

OUT 0x0C

HLT

BRAKE\_POINT:

LD brake\_letter

OUT 0x0C

HLT

# Трассировка

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Выполняемая команда | | Содержимое регистров процессора после выполнения команды | | | | | | | | Ячейка, содержимое которой изменилось после выполнения команды | |
| Адр | Код | IP | CR | AR | DR | SP | BR | AC | NZVC | Адр | Новый код |
| 0C0 | 0200 | 0C0 | 0000 | 000 | 0000 | 000 | 0000 | 0000 | 0100 |  |  |
| 0C1 | 1207 | 0C2 | 1207 | 0C1 | 1207 | 000 | 00C1 | 0000 | 0100 |  |  |
| 0C2 | 2F40 | 0C3 | 2F40 | 0C2 | 0040 | 000 | 0040 | 0000 | 0100 |  |  |
| 0C3 | F0FC | 0C0 | F0FC | 0C3 | F0FC | 000 | FFFC | 0000 | 0100 |  |  |
| 0C4 | AEFA | 0C5 | AEFA | 0BF | 0004 | 000 | FFFA | 0004 | 0000 |  |  |
| 0C5 | 1306 | 0C6 | 1306 | 0C5 | 1306 | 000 | 00C5 | 0004 | 0000 |  |  |
| 0C6 | 1207 | 0C7 | 1207 | 0C6 | 1207 | 000 | 00C6 | 0000 | 0000 |  |  |
| 0C7 | 2F40 | 0C8 | 2F40 | 0C7 | 0040 | 000 | 0040 | 0000 | 0100 |  |  |
| 0C8 | F0FD | 0C6 | F0FD | 0C8 | F0FD | 000 | FFFD | 0000 | 0100 |  |  |
| 0C9 | A8F4 | 0CA | A8F4 | 5E6 | D0CA | 000 | FFF4 | D0CA | 1000 |  |  |
| 0CA | 1306 | 0CB | 1306 | 0CA | 1306 | 000 | 00CA | D0CA | 1000 |  |  |
| 0CB | AEF3 | 0CC | AEF3 | 0BF | 0004 | 000 | FFF3 | 0004 | 0000 |  |  |
| 0CC | 0740 | 0CD | 0740 | 0CC | 0740 | 000 | 00CC | 0003 | 0001 |  |  |
| 0CD | EEF1 | 0CE | EEF1 | 0BF | 0003 | 000 | FFF1 | 0003 | 0001 | 0BF | 0003 |
| 0CE | 7F00 | 0CF | 7F00 | 0CE | 0000 | 000 | 0000 | 0003 | 0001 |  |  |
| 0CF | F00C | 0D0 | F00C | 0CF | F00C | 000 | 00CF | 0003 | 0001 |  |  |
| 0D0 | 1207 | 0D1 | 1207 | 0D0 | 1207 | 000 | 00D0 | 0040 | 0001 |  |  |
| 0D1 | 2F40 | 0D2 | 2F40 | 0D1 | 0040 | 000 | 0040 | 0040 | 0001 |  |  |
| 0D2 | F0FD | 0D3 | F0FD | 0D2 | F0FD | 000 | 00D2 | 0040 | 0001 |  |  |
| 0D3 | AAEA | 0D4 | AAEA | 5E6 | D0CA | 000 | FFEA | D0CA | 1001 | 0BE | 05E7 |
| 0D4 | 0680 | 0D5 | 0680 | 0D4 | 0680 | 000 | 00D4 | CAD0 | 1001 |  |  |
| 0D5 | 1306 | 0D6 | 1306 | 0D5 | 1306 | 000 | 00D5 | CAD0 | 1001 |  |  |
| 0D6 | AEE8 | 0D7 | AEE8 | 0BF | 0003 | 000 | FFE8 | 0003 | 0001 |  |  |
| 0D7 | 0740 | 0D8 | 0740 | 0D7 | 0740 | 000 | 00D7 | 0002 | 0001 |  |  |
| 0D8 | EEE6 | 0D9 | EEE6 | 0BF | 0002 | 000 | FFE6 | 0002 | 0001 | 0BF | 0002 |
| 0D9 | 7F00 | 0DA | 7F00 | 0D9 | 0000 | 000 | 0000 | 0002 | 0001 |  |  |
| 0DA | F001 | 0DB | F001 | 0DA | F001 | 000 | 00DA | 0002 | 0001 |  |  |
| 0DB | CEEA | 0C6 | CEEA | 0DB | 00C6 | 000 | FFEA | 0002 | 0001 |  |  |
| 0C6 | 1207 | 0C7 | 1207 | 0C6 | 1207 | 000 | 00C6 | 0040 | 0001 |  |  |
| 0C7 | 2F40 | 0C8 | 2F40 | 0C7 | 0040 | 000 | 0040 | 0040 | 0001 |  |  |
| 0C8 | F0FD | 0C9 | F0FD | 0C8 | F0FD | 000 | 00C8 | 0040 | 0001 |  |  |
| 0C9 | A8F4 | 0CA | A8F4 | 5E7 | C9C0 | 000 | FFF4 | C9C0 | 1001 |  |  |
| 0CA | 1306 | 0CB | 1306 | 0CA | 1306 | 000 | 00CA | C9C0 | 1001 |  |  |
| 0CB | AEF3 | 0CC | AEF3 | 0BF | 0002 | 000 | FFF3 | 0002 | 0001 |  |  |
| 0CC | 0740 | 0CD | 0740 | 0CC | 0740 | 000 | 00CC | 0001 | 0001 |  |  |
| 0CD | EEF1 | 0CE | EEF1 | 0BF | 0001 | 000 | FFF1 | 0001 | 0001 | 0BF | 0001 |
| 0CE | 7F00 | 0CF | 7F00 | 0CE | 0000 | 000 | 0000 | 0001 | 0001 |  |  |
| 0CF | F00C | 0D0 | F00C | 0CF | F00C | 000 | 00CF | 0001 | 0001 |  |  |
| 0D0 | 1207 | 0D1 | 1207 | 0D0 | 1207 | 000 | 00D0 | 0040 | 0001 |  |  |
| 0D1 | 2F40 | 0D2 | 2F40 | 0D1 | 0040 | 000 | 0040 | 0040 | 0001 |  |  |
| 0D2 | F0FD | 0D3 | F0FD | 0D2 | F0FD | 000 | 00D2 | 0040 | 0001 |  |  |
| 0D3 | AAEA | 0D4 | AAEA | 5E7 | C9C0 | 000 | FFEA | C9C0 | 1001 | 0BE | 05E8 |
| 0D4 | 0680 | 0D5 | 0680 | 0D4 | 0680 | 000 | 00D4 | C0C9 | 1001 |  |  |
| 0D5 | 1306 | 0D6 | 1306 | 0D5 | 1306 | 000 | 00D5 | C0C9 | 1001 |  |  |
| 0D6 | AEE8 | 0D7 | AEE8 | 0BF | 0001 | 000 | FFE8 | 0001 | 0001 |  |  |
| 0D7 | 0740 | 0D8 | 0740 | 0D7 | 0740 | 000 | 00D7 | 0000 | 0101 |  |  |
| 0D8 | EEE6 | 0D9 | EEE6 | 0BF | 0000 | 000 | FFE6 | 0000 | 0101 | 0BF | 0000 |
| 0D9 | 7F00 | 0DA | 7F00 | 0D9 | 0000 | 000 | 0000 | 0000 | 0101 |  |  |
| 0DA | F001 | 0DC | F001 | 0DA | F001 | 000 | 0001 | 0000 | 0101 |  |  |
| 0DC | 0100 | 0DD | 0100 | 0DC | 0100 | 000 | 00DC | 0000 | 0101 |  |  |

# Вывод

При выполнении данной лабораторной работы я познакомился с асинхронным вводом-выводом данных в БЭВМ, узнал больше о внешних устройствах, а также познакомился с синтаксисом ассемблера и попрактиковался в написании кода на нем.