**Министерство образования и науки**

**Российской Федерации**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Факультет программной инженерии и компьютерной техники

Дисциплина: Основы профессиональной деятельности

**Лабораторная работа №5**

Вариант 53301

Выполнил студент группы Р3133 Анисимов Максим Дмитриевич

Проверил Блохина Елена Николаевна

Санкт-Петербург

2023 г

Оглавление

[Задание 3](#_Toc134660435)

[Описание 3](#_Toc134660436)

[Текст исходной программы на языке Ассемблер 3](#_Toc134660437)

[Текст исходной программы 3](#_Toc134660438)

[Описание программы 5](#_Toc134660439)

[Область представления 5](#_Toc134660440)

[Расположение данных в памяти 5](#_Toc134660441)

[Адрес первой и последней выполняемой команды: 5](#_Toc134660442)

[Таблица трассировки 5](#_Toc134660443)

# Задание

Изображение выглядит как текст

Автоматически созданное описание

Стоп-символ можно поменять

Все остальное согласно заданию

# Описание

Передаваемое сообщение: “Потолок(“

В кодировке КОИ-8: F0 CF D4 CF CC CF CB 29

# Текст исходной программы на языке Ассемблер

|  |  |  |
| --- | --- | --- |
| ORG | 0x4EE | ; |
| START: | CLA | ; |
| S1: | IN | 7; |
|  | AND | #0x40; |
|  | BEQ | S1; |
|  | IN | 6; |
|  | SWAB | ; |
|  | ST | BEGIN; |
|  | BEQ | EXIT; |
| S2: | IN | 7; |
|  | AND | #0x40; |
|  | BEQ | S2; |
|  | LD | BEGIN; |
|  | IN | 6; |
|  | ST | (BEGIN); |
|  | BEQ | EXIT; |
|  | LD | (BEGIN)+; |
|  | CLA | ; |
|  | JUMP | S1; |
| EXIT: | LD | (BEGIN)+; |
| BEGIN: | WORD | 0x60B; |
|  | HLT | ; |

# Текст исходной программы

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код команды | Мнемоника | Комментарии |
| 4EE | 0200 | CLA | Очистка аккумулятора |
| 4EF | 1207 | IN 7 | Чтение регистра состояния ВУ-3 |
| 4F0 | 2F40 | AND #0x40 | Проверка на наличие введённого символа |
| 4F1 | F0FD | BEQ (IP-3) | Если ввод не обнаружен, то ожидание символов начинается по новой |
| 4F2 | 1206 | IN 6 | Чтение регистра данных ВУ-3 |
| 4F3 | 0680 | SWAB | Обмен старших и младших байтов |
| 4F4 | E80F | ST (IP + 12) | Сохранение символа памяти по ссылке |
| 4F5 | 7E10 | CMP (IP+17) | Установка флагов по результату AC – MEM |
| 4F6 | F00C | BEQ (IP+13) | Если аккумулятор равен 0, то происходит завершение программы |
| 4F7 | 1207 | IN 7 | Чтение регистра состояния ВУ-3 |
| 4F8 | 2F40 | AND #0x40 | Проверка на наличие введённого символа |
| 4F9 | F0FD | BEQ (IP-3) | Если ввод не обнаружен, то ожидание символов начинается по новой |
| 4FA | A809 | LD (IP+9) | Загрузка символа, хранящегося в памяти, в аккумулятор |
| 4FB | 1206 | IN 6 | Чтение регистра данных ВУ-3 |
| 4FC | E807 | ST (IP +7) | Сохранение символа памяти по ссылке |
| 4FD | 2E09 | AND (IP + 9) |  |
| 4FE | 7E06 | CMP (IP+6) | Установка флагов по результату AC – MEM |
| 4FF | F003 | BEQ (IP+3) | Если аккумулятор равен 0, то происходит завершение программы |
| 500 | AA03 | LD (IP + 3) | Инкрементация ссылки на результат |
| 501 | 0200 | CLA | Очистка аккумулятора |
| 502 | CEEC | JUMP (IP-18) | Возвращение в начало цикла |
| 503 | AA00 | LD (IP+1) | Инкрементация ссылки на результат |
| 506 | HLT | 0100 | Остановка программы |

# Описание программы

Данная программа выполняет асинхронный ввод через ВУ-3 и посимвольную запись в память. Программа получает данные до тех пор пока не будут введены символы 00. После ввода символов 00, программа останавливает своё выполнение

# Область представления

BEGIN – 11-разрядная ячейка, в которую на момент выполнения программы, сохраняются введённые символы.

FINISH – 16-разрядная ячейка, хранящая в себе стоп-символ

060B - ? – 16-рзрядные ячейки, хранящие в себе введённые данные

# Расположение данных в памяти

04EE-503, 506 – команды

503, 504,507 – исходные данные

060B - ? – результат

# Адрес первой и последней выполняемой команды:

Адрес первой команды: 04EE

Адрес последней команды: 0506

# Область допустимых значений:

BEGIN є [1547; 2047]

Введённый символ є [00; FF]

# Таблица трассировки

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Выполняемая команда | | Содержимое регистров процессора после выполнения программы | | | | | | | | Ячейка, содержимое которой поменялось после выполнения программы | |
| Адр | Знчн | IP | CR | AR | DR | SP | BR | AC | NZVC | Адр | Знчн |
| 4EE | 0200 | 4EF | 0200 | 4EE | 0200 | 000 | 04EE | 0000 | 0100 |  |  |
| 4EF | 1207 | 4F0 | 1207 | 4EF | 1207 | 000 | 04EF | 0040 | 0100 |  |  |
| 4F0 | 2F40 | 4F1 | 2F40 | 4F0 | 0040 | 000 | 0040 | 0040 | 0000 |  |  |
| 4F1 | F0FD | 4F2 | F0FD | 4F1 | F0FD | 000 | 04F1 | 0040 | 0000 |  |  |
| 4F2 | 1206 | 4F3 | 1206 | 4F2 | 1206 | 000 | 04F2 | 00F0 | 0000 |  |  |
| 4F3 | 7E10 | 4F4 | 7E10 | 504 | 0000 | 000 | 0010 | 00F0 | 0001 |  |  |
| 4F4 | F00D | 4F5 | F00D | 4F4 | F00D | 000 | 04F4 | 00F0 | 0001 |  |  |
| 4F5 | 0680 | 4F6 | 0680 | 4F5 | 0680 | 000 | 04F5 | F000 | 1001 |  |  |
| 4F6 | E80C | 4F7 | E80C | 60B | F000 | 000 | 000C | F000 | 1001 | 60B | F000 |
| 4F7 | 1207 | 4F8 | 1207 | 4F7 | 1207 | 000 | 04F7 | F040 | 1001 |  |  |
| 4F8 | 2F40 | 4F9 | 2F40 | 4F8 | 0040 | 000 | 0040 | 0040 | 0001 |  |  |
| 4F9 | F0FD | 4FA | F0FD | 4F9 | F0FD | 000 | 04F9 | 0040 | 0001 |  |  |
| 4FA | A808 | 4FB | A808 | 60B | F000 | 000 | 0008 | F000 | 1001 |  |  |
| 4FB | 1206 | 4FC | 1206 | 4FB | 1206 | 000 | 04FB | F0CF | 1001 |  |  |
| 4FC | 7E07 | 4FD | 7E07 | 504 | 0000 | 000 | 0007 | F0CF | 1001 | 60B | F0CF |
| 4FD | F004 | 4FE | F004 | 4FD | F004 | 000 | 04FD | F0CF | 1001 |  |  |
| 4FE | E804 | 4FF | E804 | 60B | F0CF | 000 | 0004 | F0CF | 1001 | 60B | F0CF |
| 4FF | AA03 | 500 | AA03 | 60B | F0CF | 000 | 0003 | F0CF | 1001 | 503 | 060C |
| 500 | 0200 | 501 | 0200 | 500 | 0200 | 000 | 0500 | 0000 | 0101 |  |  |
| 501 | CEED | 4EF | CEED | 501 | 04EF | 000 | FFED | 0000 | 0101 |  |  |
| 4EF | 1207 | 4F0 | 1207 | 4EF | 1207 | 000 | 04EF | 0040 | 0101 |  |  |
| 4F0 | 2F40 | 4F1 | 2F40 | 4F0 | 0040 | 000 | 0040 | 0040 | 0001 |  |  |
| 4F1 | F0FD | 4F2 | F0FD | 4F1 | F0FD | 000 | 04F1 | 0040 | 0001 |  |  |
| 4F2 | 1206 | 4F3 | 1206 | 4F2 | F0FD | 000 | 04F2 | 0000 | 0001 |  |  |
| 4F3 | 7E10 | 4F4 | 7E10 | 504 | F0FD | 000 | 0010 | 0000 | 0101 |  |  |
| 4F4 | F00D | 502 | F00D | 4F4 | F00D | 000 | 000D | 0000 | 0101 |  |  |
| 502 | AA00 | 503 | AA00 | 60C | 0000 | 000 | 0000 | 0000 | 0101 | 503 | 060D |
| 503 | 060D | 504 | 060D | 503 | 060D | 000 | 0503 | 0000 | 0101 |  |  |
| 504 | 0000 | 505 | 0000 | 504 | 0000 | 000 | 0504 | 0000 | 0101 |  |  |
| 505 | 0100 | 506 | 0100 | 505 | 0100 | 000 | 0505 | 0000 | 0101 |  |  |

# Вывод

Во время работы над лабораторной работой я познакомился с БЭВМ реализацией Assembler и использовал его более быстрого написания кода. Также познакомился с ВУ и написал работающий вариант бегущей строки, где ввод идет с внутренней клавиатуры.

# Программа осуществляет ввод данных с ВУ-3 до стоп-символа, после чего вывод нечетные символы в обратном порядке

|  |  |  |
| --- | --- | --- |
| ORG | 0x4EE | ; |
| START: | CLA | ; |
| S1: | IN | 7; |
|  | AND | #0x40; |
|  | BEQ | S1; |
|  | IN | 6; |
|  | SWAB | ; |
|  | ST | (BEGIN); |
|  | AND | STOP1; |
|  | BEQ | EXIT1; |
| S2: | IN | 7; |
|  | AND | #0x40; |
|  | BEQ | S2; |
|  | LD | (BEGIN); |
|  | IN | 6; |
|  |  |  |
|  | ST | (BEGIN); |
|  | AND | STOP2; |
|  | BEQ | EXIT2; |
|  | LD | (BEGIN)+; |
|  | CLA | ; |
|  | JUMP | S1; |
|  |  |  |
| EXIT1: | LD | BEGIN; |
|  | AND | MAIN; |
|  | BEQ | CONEC; |
|  | LD | (BEGIN); |
|  | AND | AND1; |
|  | SWAB | ; |
|  | OUT | 12; |
|  | LOOP | BEGIN; |
|  |  |  |
|  | JUMP | EXIT1; |
|  | HLT | ; |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| EXIT2: | LD | BEGIN; |
|  | AND | MAIN; |
|  | BEQ | CONEC; |
|  |  |  |
|  | LD | (BEGIN); |
|  | AND | AND2; |
|  | OUT | 12; |
|  |  |  |
|  |  |  |
|  |  |  |
|  | LOOP | BEGIN; |
|  |  |  |
|  | JUMP | EXIT2; |
| CONEC: | HLT | ; |
|  |  |  |
|  |  |  |
| MAIN: | WORD | 0x9F5; |
| BEGIN: | WORD | 0x60B; |
| FINISH: | WORD | 0xE0; |
| STOP2: | WORD | 0x001F; |
| STOP1: | WORD | 0x1F00; |
| AND1: | WORD | 0xFF00; |
| AND2: | WORD | 0x00FF; |
| NUMBER: | WORD | 0x01; |