Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

МФКТиУ, кафедра Вычислительной техники

Лабораторная работа №5

по дисциплине

«Основы вычислительной техники»

Выполнил: Студент группы P3131

Зубахин Дмитрий

Преподаватель:

Тимофей Сергеевич Перцев

Санкт-Петербург

2021 г.

# Задание: (3186 вариант)

Изображение выглядит как текст

Автоматически созданное описаниеПо 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.

# Программа Assembler:

ORG 0x072 ; Адрес начала программы

RES: WORD 0 ; Хранение старшего байта слова

ADDR: WORD $X ; Ссылка на следующее слово

START: LD (ADDR)+ ; Очередное слово

ST RES ; Сохранение старшего байта

PUSH ; Поместить 2 байта в стек

CALL F ; Вызов подпрограммы

POP ; Очистить стек

LD RES ; Загрузить старший байт

SWAB ; Поменять байты местами

PUSH ; Поместить 2 байта в стек

CALL F ; Вызов подпрограммы

POP ; Очистить стек

BR START ; Вернуться к началу программы

F: IN 7 ; Значения регистра SR ВУ-3

AND #0x40 ; Проверка значения

BEQ F ; Spin-Loop

LD &1 ; Взятие данных из стека

CMP #0x0A ; Проверка на стоп символ

OUT 6 ; Вывод байта в регистр DR ВУ-3

BEQ STOP ; Завершение программ если байт – стоп слово

RET ; Возврат из подпрограммы

STOP: HLT ; Остановка программы

ORG 0x614 ; Массив слов

X:

# Программа БЭВМ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес | Команда | Мнемоника | Описание | Комментарий |
| 072 | 0000 | WORD | RES | RES |
| 073 | 0614 | WORD | ADDR | ADDR |
| 074 | AAFE | LD (IP-2)+ | MEM(ADDR) → AC  ADDR = ADDR + 1 | START |
| 075 | EEFC | ST (IP-4)+ | AC → RES |  |
| 076 | 0C00 | PUSH | AC → -(SP) |  |
| 077 | DE07 | CALL 07F | CALL F |  |
| 078 | 0800 | POP | (SP)+ → AC |  |
| 079 | AEF8 | LD (IP-8) | RES → AC |  |
| 07A | 0680 | SWAB | AC15… AC­8 ↔ AC7… AC­0 |  |
| 07B | 0C00 | PUSH | AC → -(SP) |  |
| 07C | DE02 | CALL 07F | CALL F |  |
| 07D | 0800 | POP | (SP)+ → AC |  |
| 07E | CEF5 | JUMP (IP-11) | 074 → IP |  |
| 07F | 1207 | IN 7 | 0x07 → AC | F |
| 080 | 2F40 | AND #64 | AND 0x40 |  |
| 081 | F0FD | BEQ (IP-3) | 07F → IP |  |
| 082 | AC01 | LD (SP+1) | MEM(7FE) → AC |  |
| 083 | 7F0A | CMP #10 | FLAGS (AC - 10) |  |
| 084 | 1306 | OUT 6 | AC → 0x06 |  |
| 085 | F001 | BEQ (IP+1) | 592 → IP |  |
| 086 | 0A00 | RET | MEM(7FF) → IP |  |
| 087 | 0100 | HLT | STOP | STOP |

# Описание программы:

|  |  |
| --- | --- |
| Ячейка | Назначение |
| 0x072 – 0x07E | Программа |
| 0x07F – 0x086 | Подпрограмма |
| 0x614 - ? | Строка |

# ОПИ:

|  |  |
| --- | --- |
| Ячейка | ОПИ |
| 0x072 – 0x087 | 16-ти разрядные команды |
| 0x614 - ? | 16-ти разрядные символы в кодировке ISO-8859-5 |

# Вывод:

В процессе выполнения лабораторной работы я познакомился с вводом-выводом в БЭВМ через ВУ-3, командами IN и OUT, а так же с языком Assembler БЭВМ.

# Строка:

|  |  |
| --- | --- |
| Текст | Пощады не будет! |
| КОИ-8 | BFDE E9D0 D4EB 20DD D520 D1E3 D4D5 E221 |
| БЭВМ | *DEBF D0E9 EBD4 DD20 20D5 E3D1 D5D4 21E2 000A* |

# Трассировка:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Выполняемая команда | | Содержание 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 |
| 074 | AAFE | 075 | AAFE | 614 | DEBF | 000 | FFFE | DEBF | 1000 | 073 | 0615 |
| 075 | EEFC | 076 | EEFC | 072 | DEBF | 000 | FFFC | DEBF | 1000 | 072 | DEBF |
| 076 | 0C00 | 077 | 0C00 | 7FF | DEBF | 7FF | 0076 | DEBF | 1000 | 7FF | DEBF |
| 077 | DE07 | 07F | DE07 | 7FE | 0078 | 7FE | 007F | DEBF | 1000 | 7FE | 0078 |
| 07F | 120D | 080 | 120D | 07F | 120D | 7FE | 007F | DE40 | 1000 |  |  |
| 080 | 2F40 | 081 | 2F40 | 080 | 0040 | 7FE | 0040 | 0040 | 0001 |  |  |
| 081 | F0FD | 082 | F0FD | 081 | F0FD | 7FE | 0081 | 0040 | 0001 |  |  |
| 082 | AC01 | 083 | AC01 | 7FF | DEBF | 7FE | 0001 | DEBF | 1000 |  |  |
| 083 | 7F0A | 084 | 7F0A | 083 | 000A | 7FE | 000A | DEBF | 1001 |  |  |
| 084 | 130C | 085 | 130C | 084 | 130C | 7FE | 0084 | DEBF | 1001 |  |  |
| 085 | F001 | 086 | F001 | 085 | F001 | 7FE | 0085 | DEBF | 1001 |  |  |
| 086 | 0A00 | 078 | 0A00 | 7FE | 0078 | 7FF | 0086 | DEBF | 1001 |  |  |
| 078 | 0800 | 079 | 0800 | 7FF | DEBF | 000 | 0078 | DEBF | 1001 |  |  |
| 079 | AEF8 | 07A | AEF8 | 072 | DEBF | 000 | FFF8 | DEBF | 1001 |  |  |
| 07A | 0680 | 07B | 0680 | 07A | 0680 | 000 | FFF8 | BFDE | 1001 |  |  |
| 07B | 0C00 | 07C | 0C00 | 07B | BFDE | 7FF | 007B | BFDE | 1001 | 7FF | BFDE |
| 07C | DE02 | 07F | DE02 | 7FE | 007D | 7FE | 007F | BFDE | 1001 | 7FE | 007D |
| 07F | 120D | 080 | 120D | 07F | 120D | 7FE | 007F | BF40 | 1001 |  |  |
| 080 | 2F40 | 081 | 2F40 | 080 | 0040 | 7FE | 0040 | 0040 | 0001 |  |  |
| 081 | F0FD | 082 | F0FD | 081 | F0FD | 7FE | 0081 | 0040 | 0001 |  |  |
| 082 | AC01 | 083 | AC01 | 7FF | BFDE | 7FE | 0001 | BFDE | 1001 |  |  |
| 083 | 7F0A | 084 | 7F0A | 083 | 000A | 7FE | 000A | BFDE | 1001 |  |  |
| 084 | 130C | 085 | 130C | 084 | 130C | 7FE | 0084 | BFDE | 1001 |  |  |
| 085 | F001 | 086 | F001 | 085 | F001 | 7FE | 0085 | BFDE | 1001 |  |  |
| 086 | 0A00 | 07D | 0A00 | 7FF | 007D | 7FF | 0086 | BFDE | 1001 |  |  |
| 07D | 0800 | 07E | 0800 | 7FF | BFDE | 000 | 007D | BFDE | 1001 |  |  |
| 07E | CEF5 | 074 | CEF5 | 07E | 0074 | 000 | FFF5 | BFDE | 1001 |  |  |