Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

**Лабораторная работа № 3**

“Выполнение циклических программ”

Вариант № 1308

Выполнил:

Сандов Кирилл Алекссевич

Группа:

P3113

Проверила:

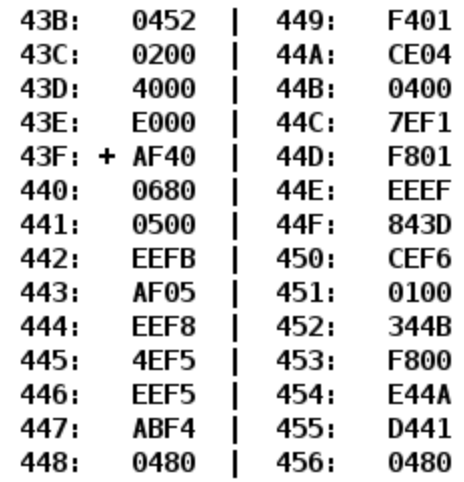
преподаватель Блохина Елена Николаевна

Санкт-Петербург

2023

# Задание

По выданному преподавателем варианту восстановить текст заданного варианта программы (Рисунок 1), определить предназначение и составить описание программы, определить область представления и область допустимых значений исходных данных и результата, выполнить трассировку программы.



Рисунок

# Пункт 1

Запишем исходную программу в виде таблицы (Таблица 1).

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарии** |
| 43F | AF40 | LD #0x40 | Записать значение 40 в аккумулятор |
| 440 | 0680 | SWAB | Обмен старшего и младшего байта аккумулятора |
| 441 | 0500 | ASL | Арифметический сдвиг влево аккумулятора |
| 442 | EEFB | ST IP-5 | Сохранить значение аккумулятора в ячейку 442+1-5=43E |
| 443 | AF05 | LD #0x05 | Загрузить значение 5 в аккумулятор |
| 444 | EEF8 | ST IP-8 | Сохранить значение аккумулятора в ячейку 444+1-8=43D |
| 445 | 4EF5 | ADD IP-B | Прибавить значение из ячейки 445+1-B=43B к аккумулятору |
| 446 | EEF5 | ST IP-B | Сохранить значение аккумулятора в ячейку 446+1-B=43C |
| 447 | ABF4 | LD –(-C) | Получить адрес, который является значением ячейки 447+1-C=43C, декрементировать его и записать обратно в 43С, загрузить значение из ячейки по этому адресу в аккумулятор |
| 448 | 0480 | ROR | Циклический сдвиг вправо аккумулятора |
| 449 | F401 | BHIS IP+1 | Переход в ячейку 449+1+1=44B, если стоит флаг переноса, иначе в следующую ячейку (44A) |
| 44A | CE04 | JUMP IP+4 | Установить значение регистра адреса равным 44A+1+4=44F |
| 44B | 0400 | ROL | Циклический сдвиг влево аккумулятора |
| 44С | 7EF1 | CMP IP-F | Получить число из ячейки 44C+1-F=43E и установить флаги NZVC по результату операции разности значений аккумулятора и полученного числа |
| 44D | F801 | BLT IP+1 | Переход в ячейку 44D+1+1=44F, если (число из аккумулятора строго меньше числа из ячейки 43Е, полученного на предыдущем шаге), иначе переход в следующую ячейку (44E) |
| 44E | EEEF | ST IP-11 | Сохранить значение аккумулятора в ячейку 44E+1-11=43E |
| 44F | 843D | LOOP 43D | Декрементировать значение в  ячейке 43D, и если оно стало равным  0, то перейти в ячейку 44F+1+1=451, иначе в следующую ячейку (450) |
| 450 | CEF6 | JUMP IP-A | Установить значение регистра адреса равным 450+1-A=447 |
| 451 | 0100 | HLT | Останов |

Таблица

# Пункт 2

Описание программы:

* **Назначение программы:** программа находит наибольшее нечётное число в одномерном массиве длиной 5
* **Новые исходные данные**
* -90
* 300
* 0
* 333
* 9999
* **Описание и назначение исходных данных, область представления и область допустимых значений исходных данных и результата:**

- R – переменная, хранящая результат работы программы – битовый набор, составленный из первых битов элементов массива;

- I – переменная-счётчик, показывающая, сколько итераций цикла осталось. Изначально равняется размеру массива;

- S – переменная, содержащая адрес текущей рассматриваемой ячейки массива

Область представления:

* + R – набор из 16-ти логических однобитовых значений,

;

* + I – число, не превышающее общее количество ячеек памяти,

2047;

* + S – адрес ячейки памяти,

;

Область допустимых значений:

* **Расположение в памяти ЭВМ программы, исходных данных и результатов:** программа располагается в ячейках 54A-559, исходные данные – в ячейке 546, массив – в ячейках 55A-55C, результаты – в ячейках 547, 548, 549.
* **Адреса первой и последней выполняемой команд программы:** первая – 54A, последняя – 559.

# Пункт 3

Новые исходные данные для таблицы трассировки в 10-формате:

Переведём их в 16-формат, преобразуя в дополнительный код отрицательные числа, и обозначим ячейки памяти в таблице, в которые они будут записаны (Таблица 2):

|  |  |
| --- | --- |
| **Адрес** | **Значение** |
| 131 | 006D |
| 130 | FDC9 |
| 134 | FC86 |

Таблица

Запишем таблицу трассировки программы (Таблица 3):

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Выполняемая команда** | | **Содержимое 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 |
| 128 | 0200 | 129 | 0200 | 128 | 0200 | 000 | 0128 | 0000 | 0100 | - | - |
| 129 | 4131 | 12A | 4131 | 131 | 006D | 000 | 0129 | 006D | 0000 | - | - |
| 12A | 4130 | 12B | 4130 | 130 | FDC9 | 000 | 012A | FE36 | 1000 | - | - |
| 12B | E133 | 12C | E133 | 133 | FE36 | 000 | 012B | FE36 | 1000 | 133 | FE36 |
| 12C | A134 | 12D | A134 | 134 | FC86 | 000 | 012C | FC86 | 1000 | - | - |
| 12D | 3133 | 12E | 3133 | 133 | FE36 | 000 | 012D | FEB6 | 1000 | - | - |
| 12E | E132 | 12F | E132 | 132 | FEB6 | 000 | 012E | FEB6 | 1000 | 132 | FEB6 |
| 12F | 0100 | 130 | 0100 | 12F | 0100 | 000 | 012F | FEB6 | 1000 | - | - |

Таблица

# Пункт 4

Запишем вариант программы с меньшим числом команд в виде таблицы (Таблица 4).

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Комментарии** |
| 129 | 4131 | ADD 131 | Добавить содержимое ячейки памяти 131 к аккумулятору |
| 12A | 4130 | ADD 130 | Добавить содержимое ячейки памяти 130 к аккумулятору |
| 12D | 3133 | OR 133 | Выполнить побитовое ИЛИ для ячейки памяти 133 и аккумулятора и результат записать в аккумулятор |
| 12E | E132 | ST 132 | Записать значение из аккумулятора в ячейку памяти 132 |
| 12F | 0100 | HLT | Останов |

Таблица

# Заключение

В результате выполнения данной лабораторной работы были расширены знания о командах БЭВМ. Во-первых, был рассмотрен пример программы, использующей циклы. Во-вторых, получен навык организации данных в одномерном массиве и перебора его элементов в цикле. Также были изучены дополнительные режимы адресации: относительная прямая, косвенная, автоинкрементная, автодекрементная и с прямой загрузкой операнда.