

Санкт-Петербургский политехнический университет Петра Великого  
Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе**  
**Дисциплина:** Низкоуровневое программирование  
**Тема:** Программирование EDSAC (Вариант 1)

Выполнил студент гр. 3530901/00002  
Эссименган А.Л.М

\_\_\_\_\_ Аингон

(подпись)

Преподаватель

\_\_\_\_\_ Д.С  
Степанов

(подпись)

Санкт-Петербург  
2021

Задача:

- 1) Разработать программу для EDSAC, реализующую определенную вариантом задания функциональность, и предполагающую загрузчик Initial Orders 1. Массив (массивы) данных и другие параметры (преобразуемое число, длина массива, параметр статистики и пр.) располагаются в памяти по фиксированным адресам.
- 2) Выделить определенную вариантом задания функциональность в замкнутую (closed) подпрограмму, разработать вызывающую ее тестовую программу. Использовать возможности загрузчика Initial Orders 2. Адрес обрабатываемого массива данных и другие параметры передавать через ячейки памяти с фиксированными адресами.

Задание по варианту:

Формирование в памяти десятичного представления целого числа со знаком.

Код для Initial Order 1

T 91 S [31 ]

A 84 S [32 - приравниваем асс к исходному числу]

E 38 S [33 - знаковый разряд]

[IF]

T 93 S [34 ]

A 85 S [35 - добавляем в асс число 22]

T 106 S[36 - сохраняю в ячейку 106 для знакового разряда]

S 84 S [37 - делаем из отрицательного числа положительное]

[ENDIF]

[4 разряд]

U 92 S [38 - сохраняю текущее значение в ячейке 92]

S 86 S [39 - вычитаю из асс значение 10000]

G 47 S [40 - проверяю на отрицательность]

[IF]

T 92 S [41 - сохраняю текущее значение в ячейке 92]

A 107 S [42 - содержимое в ячейке для старшего разряда в асс]

A 90 S [43 - прибавляю 1]

T 107 S [44 - сохраняю в ячейке для 4 разряда]

A 92 S [45 - беру исходное число после вычитания]  
E 38 S [46 - переход к новой итерации по циклу или к следующему разряду]  
[ENDIF]  
[3 разряд]  
T 93 S [47]  
A 92 S [48]  
U 92 S [49 - сохраняю текущее значение в ячейке 92]  
S 87 S [50 - вычитаю из асс значение 1000]  
G 58 S [51 - проверяю на отрицательность]  
[IF]  
T 92 S [52 - сохраняю текущее значение в ячейке 92]  
A 108 S [53 - содержимое в ячейке для 3 разряда в асс]  
A 90 S [54 - прибавляю 1]  
T 108 S [55 - сохраняю в ячейке для 3 разряда]  
A 92 S [56 - беру исходное число после вычитания]  
E 49 S [57 - переход к новой итерации по циклу или к следующему разряду]  
[ENDIF]  
[2 разряд]  
T 250 S [58]  
A 92 S [59]  
U 92 S [60 - сохраняю текущее значение в ячейке 92]  
S 88 S [61 - вычитаю из асс значение 100]  
G 69 S [62 - проверяю на отрицательность]  
[IF]  
T 92 S [63 - сохраняю текущее значение в ячейке 92]  
A 109 S [64 - содержимое в ячейке для 2 разряда в асс]  
A 90 S [65 - прибавляю 1]

Т 109 S [66 - сохраняю в ячейке для 2 разряда]  
 А 92 S [67 - беру исходное число после вычитания]  
 Е 60 S [68 - переход к новой итерации по циклу или к следующему разряду]  
 [ENDIF]  
 [1 разряд]  
 Т 93 S [69]  
 А 92 S [70]  
 U 92 S [71 - сохраняю текущее значение в ячейке 128]  
 S 89 S [72 - вычитаю из асс значение 10]  
 G 80 S [73 - проверяю на отрицательность]  
 [IF]  
 Т 92 S [74 - сохраняю текущее значение в ячейке 128]  
 А 110 S [75 - содержимое в ячейке для 1 разряда в асс]  
 А 90 S [76 - прибавляю 1]  
 Т 110 S [77 - сохраняю в ячейке для 1 разряда]  
 А 92 S [78 - беру исходное число после вычитания]  
 Е 71 S [79 - переход к новой итерации по циклу или к следующему разряду]  
 [ENDIF]  
 [0 разряд]  
 Т 93 S [80]  
 А 92 S [81 - оставшиеся единичные значения сохраняем в ячейке для младшего разряда]  
 Т 111 S [82]  
 Z S [83]  
 [константы и исходные данные]  
 Р 57005 L [84 - исходное число -17061]  
 Р 11 S [85 - минус (22)]  
 Р 5000 S [86 - 10000]

P 500 S [87 - 1000]

P 50 S [88 - 100]

P 5 S [89 - 10]

P 0 L [90 - 1]

В ходе выполнения данной программы заданное число преобразуется в символьное представление в памяти. Число задается в адресе 84. После выполнения программы число содержится в диапазоне адресов 106-111, где:

106 - знаковый разряд;

107 - 4 числовой разряд;

108 - 3 числовой разряд;

109 - 2 числовой разряд;

110 - 1 числовой разряд;

111 - 0 числовой разряд

Пример:

На вход программы подается исходное число -17061.

На выходе получаем:

```
WORD 106 Order = P 11 S Integer 106S = 22 Fraction 106S = 0.000336
WORD 107 Order = P 0 L Integer 107S = 1 Fraction 106L = 0.00001526007
WORD 108 Order = P 3 L Integer 108S = 7 Fraction 108S = 0.000107
WORD 109 Order = P 0 S Integer 109S = 0 Fraction 108L = 0.00000000041
WORD 110 Order = P 3 S Integer 110S = 6 Fraction 110S = 0.000092
WORD 111 Order = P 0 L Integer 111S = 1 Fraction 110L = 0.00001525914
```

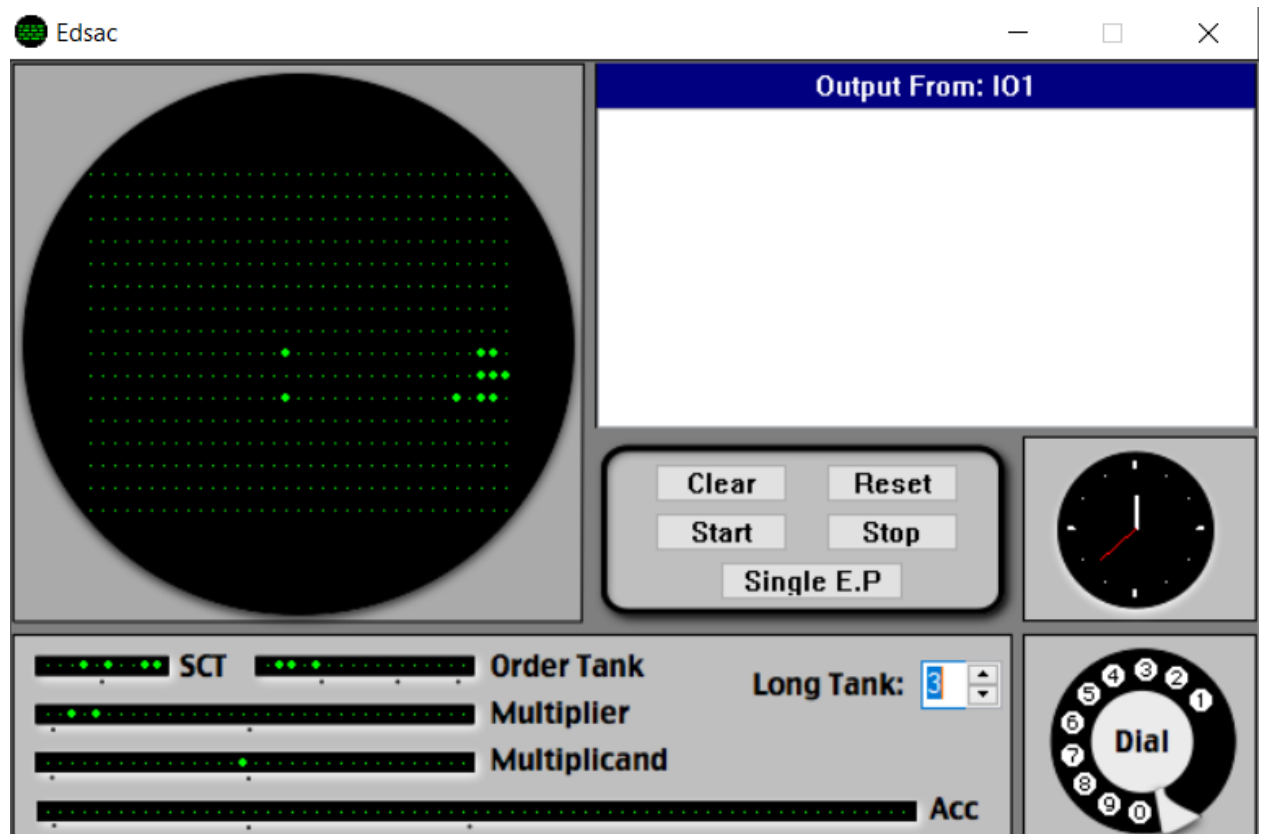


Рис. 1 Результат работы программы. Initial Order 1.

Код для Initial Order 2

Закрытая подпрограмма.

T 56 K [- директива IO2, установка адреса загрузки]

G K [- директива IO2, фиксация начального адреса подпрограммы]

[0]A3F[- пролог: формирование кода инструкции возврата в аккумуляторе]

[1]T53@[- пролог запись инструкции возврата]

[2]A1F[- добавление ячейки с исходным числом в аккумулятор]

[3]E8@[- Если число положительное, то переходим к проверке старшего разряда]

[IF]

[4]T300F

[5]A54@[- добавляем в асс число 22]

[6]T168F[- сохраняю в ячейку 168 для знакового разряда]

[7]S1F[- делаем из отрицательного числа положительное]

[ENDIF]

[4 разряд]

[8]U128F[- сохраняю текущее значение в ячейке 128]

[9]S55@[- вычитаю из асс значение 10000]

[10]G17@[- проверяю на отрицательность]

[IF]

[11]T128F[- сохраняю текущее значение в ячейке 128]

[12]A169F[- содержимое в ячейке для старшего разряда в асс]

[13]A59@[- прибавляю 1]

[14]T169F[- сохраняю в ячейке для 4 разряда]

[15]A128F[- беру исходное число после вычитания]

[16]E8@[- переход к новой итерации по циклу или к следующему разряду]  
[ENDIF]

[3 разряд]

[17]T300F

[18]A128F

[19]U128F[- сохраняю текущее значение в ячейке 128]

[20]S56@[- вычитаю из асс значение 1000]

[21]G28@[- проверяю на отрицательность]

[IF]

[22]T128F[- сохраняю текущее значение в ячейке 128]

[23]A170F[- содержимое в ячейке для 3 разряда в асс]

[24]A59@[- прибавляю 1]

[25]T170F[- сохраняю в ячейке для 3 разряда]

[26]A128F[- беру исходное число после вычитания]

[27]E19@[- переход к новой итерации по циклу или к следующему разряду]

[ENDIF]

[2 разряд]

[28]T300F

[29]A128F

[30]U128F[- сохраняю текущее значение в ячейке 128]

[31]S57@[- вычитаю из асс значение 100]

[32]G39@[- проверяю на отрицательность]

[IF]



[33]T128F[- сохраняю текущее значение в ячейке 128]  
[34]A171F[- содержимое в ячейке для 2 разряда в асс]  
[35]A59@[- прибавляю 1]  
[36]T171F[- сохраняю в ячейке для 2 разряда]  
[37]A128F[- беру исходное число после вычитания]  
[38]E30@[- переход к новой итерации по циклу или к следующему разряду]  
[ENDIF]

[1 разряд]

[39:]T300F  
[40:]A128F  
[41:]U128F[- сохраняю текущее значение в ячейке 128]  
[42:]S58@[- вычитю из асс значение 10]  
[43:]G50@[- проверяю на отрицательность]  
[IF]  
[44]T128F[- сохраняю текущее значение в ячейке 128]  
[45]A172F[- содержимое в ячейке для 1 разряда в асс]  
[46]A59@[- прибавляю 1]  
[47]T172F[- сохраняю в ячейке для 1 разряда]  
[48]A128F[- беру исходное число после вычитания]  
[49]E41 @[- переход к новой итерации по циклу или к следующему разряду]  
[ENDIF]

[0 разряд]

[50:]T300F

[51:]A128F [- оставшиеся единичные значения сохраняем в ячейке для младшего разряда]

[52:]T173 F

[53]E0F[- эпилог: инструкция возврата из подпрограммы]

[54]P11F[- минус (22)]

[55]P5000F [- 10000]

[56]P500F[- 1000]

[57]P50F[- 100]

[58]P5F[- 10]

[59]P0D[- 1]

\*Код программы, использующий подпрограмму.

G K [- директива IO2, фиксация начального адреса программы]

[0]A5@[- загрузка в аккумулятор исходного числа]

[1]T1F[- запись исходного числа в ячейку 1]

[2]A2@[- вызов]

[3]G56F [- подпрограммы]

[4]ZF[- остановка]

[5]P62D[- исходное число 125]

EZ PFB

данной программе исходное число задается в ячейку 1. Далее идет вызов подпрограммы, которая записывает исходное число в соответствующие ячейки памяти. После выполнения программы число содержится в диапазоне адресов 168-173, где:

168 - знаковый разряд;

169 - 4 числовой разряд;

170 - 3 числовой разряд;

171 - 2 числовой разряд;

172 - 1 числовой разряд;

173 - 0 числовой разряд.

Пример:

На вход программы подается исходное число 125.

На выходе получаем:

```
WORD 168 Order = P 0 F Integer 168F = 0 Fraction 168F = 0.000000
WORD 169 Order = P 0 F Integer 169F = 0 Fraction 168D = 0.000000000000
WORD 170 Order = P 0 F Integer 170F = 0 Fraction 170F = 0.000000
WORD 171 Order = P 0 D Integer 171F = 1 Fraction 170D = 0.00001525879
WORD 172 Order = P 1 F Integer 172F = 2 Fraction 172F = 0.000031
WORD 173 Order = P 2 D Integer 173F = 5 Fraction 172D = 0.00007629406
```

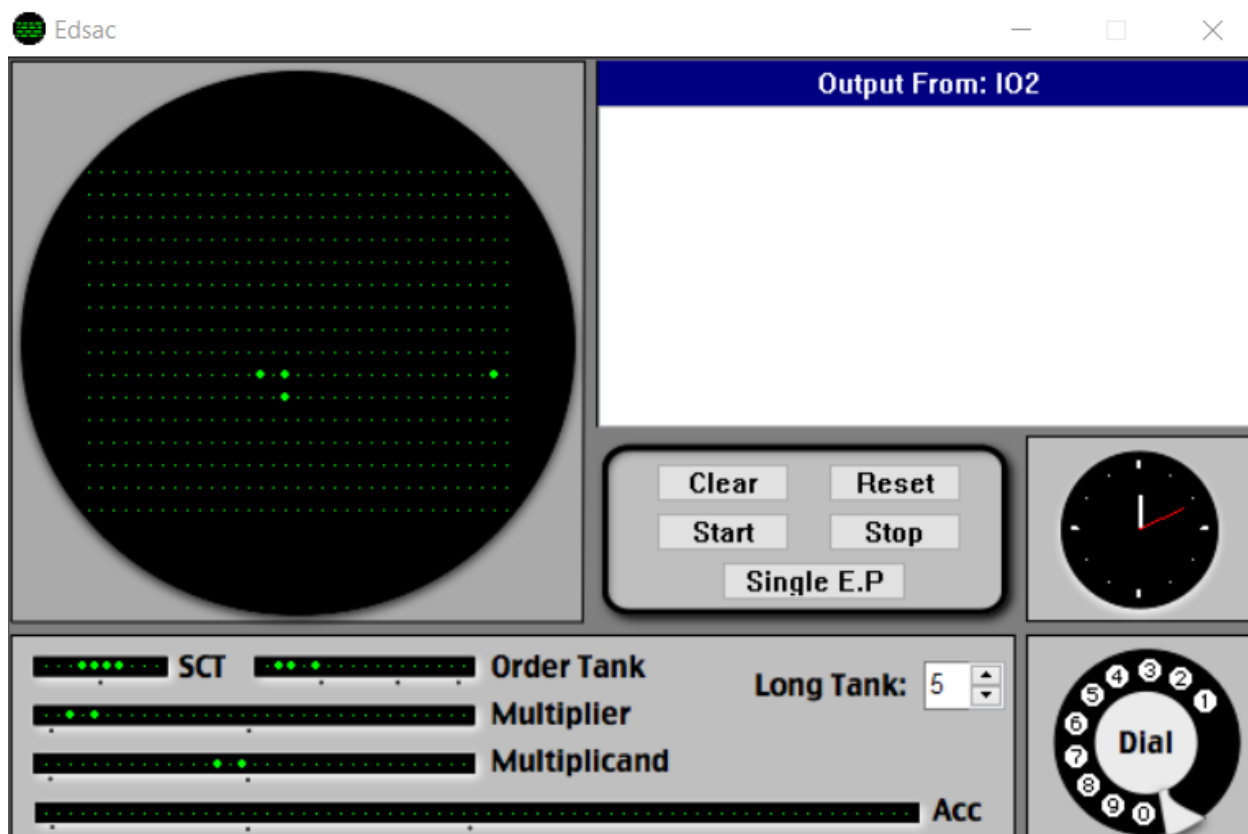


Рис. 2 Результат работы программы. Initial Order 2.

#### Вывод:

В данной работе мы ознакомились с работой EDSAC и его двумя загрузчиками. Была разработана программа, использующая загрузчик Initial Orders 1, которая реализовала определенную функциональность. Кроме того, используя загрузчик Initial Orders 2, была выделена заданная функциональность в замкнутую подпрограмму и разработана вызывающая ее программа. Ожидаемые результаты совпали с результатами работы симулятора, значит программа работает корректно.

#### ➤ Список использованных источников

[http://kspt.icc.spbstu.ru/media/files/2019/lowlevelprog/edsac\\_io.pdf](http://kspt.icc.spbstu.ru/media/files/2019/lowlevelprog/edsac_io.pdf)

[http://kspt.icc.spbstu.ru/media/files/2019/lowlevelprog/edsac\\_prg.pdf](http://kspt.icc.spbstu.ru/media/files/2019/lowlevelprog/edsac_prg.pdf)

[http://kspt.icc.spbstu.ru/media/files/2020/lowlevelprog/edsac\\_sub.pdf](http://kspt.icc.spbstu.ru/media/files/2020/lowlevelprog/edsac_sub.pdf)

<http://www.cl.cam.ac.uk/~mr/edsacposter.pdf>