

Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**Отчет по лабораторной работе №2**

**Дисциплина:** Низкоуровневое программирование.

**Тема:** Программирование на EDSAC.

Выполнил

студент гр. 3530901/90003 \_\_\_\_\_ Бехтольд Ек.В.  
(подпись)

Принял

преподаватель \_\_\_\_\_ Алексюк А.О.  
(подпись)

«\_\_\_» \_\_\_\_\_ 2021 г.

Санкт-Петербург  
2021

## ОГЛАВЛЕНИЕ

|  |    |
|--|----|
| 1. Техническое задание .....               | 3  |
| 2. Метод решения .....                     | 3  |
| 3. Программа Initial Orders 1.....         | 4  |
| 4. Работа программы Initial Orders 1 ..... | 5  |
| 5. Программа Initial Orders 2 .....        | 6  |
| 6. Работа программы Initial Orders 2 ..... | 9  |
| 7. Выводы .....                            | 10 |

## 1. Техническое задание.

1.1 Разработать программу для EDSAC, реализующую циклический сдвиг массива чисел на заданное количество разрядов влево, 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.

1.2 Выделить циклический сдвиг массива чисел на заданное количество разрядов влево 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.

## 2. Метод решения.

Пример: пусть в смежных ячейках памяти машины, начиная, например, с адреса 100, размещены следующие значения:  $1 \cdot 2^{-16}$ ,  $2 \cdot 2^{-16}$ ,  $3 \cdot 2^{-16}$ ,  $4 \cdot 2^{-16}$ ,  $5 \cdot 2^{-16}$ . В двоичном представлении:

101: 0000000000000001

102: 0000000000000010

103: 0000000000000011

104: 0000000000000100

105: 0000000000000101

В целочисленной интерпретации, эти комбинации разрядов соответствуют числам 1, 2, 3, 4, 5. После выполнения программы в тех же ячейках памяти должны располагаться числа: 4, 5, 1, 2, 3 (осуществлен циклический сдвиг на 3 разряда влево).

Алгоритм решения задачи:

- Шаг 1: значение нулевого элемента массива помещаем в рабочую ячейку.
- Шаг 2: значение первого элемента массива помещаем на место нулевого элемента.
- Шаг 3: значение из рабочей ячейки переносим на место первого элемента.
- Шаг 4: повторяем шаги 1-3 для второго и третьего элемента, затем для следующих двух элементов и так до конца массива.
- Шаг 5: повторяем цикл столько раз, на сколько требуется сдвинуть массив чисел.

### 3. Программа Initial Orders 1

#### Листинг 1: Программа для Initial Orders 1

[ T3: Разработать программу для EDSAC, реализующую циклический сдвиг массива чисел ]  
 [ на заданное количество разрядов влево, и предполагающую загрузчик IO1. ]  
 [ Массив хранится в ячейках 102-106 ]  
 [ Количество сдвигов хранится в ячейке 101 ]  
 [ результат сохраняется на месте исходного массива ячейки 102-106 ]  
 [ ячейки 0-4 рабочие ]

[31:] T 107 S

[32:] Z 0 S [ останов для отладки ]

[33:] T 0 S [ запись аккумулятора в рабочую ячейку 0, обнуление аккумулятора ]

[34:] A 101[count:] S [ загрузка в аккумулятор количества сдвигов ]

[35:] T 2 S [ запись этого значения в ячейку 2 ]

[36:] A 100[length:] S [ загрузка в аккумулятор длины обрабатываемого массива ]

[37:] U 4 S [ запись этого значения в ячейку 4 ]

[38:] L0L [ сдвиг аккумулятора на 1 разряд влево ]

[39:] T 98[cLen:] S [ запись этого значения в ячейку 98 ]

[40:] A 99[<addr>] S [ загрузка в аккумулятор адреса 0-го элемента массива ]

[41:] L0L [ сдвиг аккумулятора на 1 разряд влево ]

[42:] A 64[<r1>] S [ прибавляем код инструкции с нулевым полем адреса ]

[43:] T 64[<r1>] S [ запись сформированной инструкции, обнуление аккумулятора ]

[44:] A 99[<addr>] S [ загрузка в аккумулятор адреса 0-го элемента массива ]

[45:] L0L [ сдвиг аккумулятора на 1 разряд влево ]

[46:] A 67[<w1>] S [ прибавляем код инструкции с нулевым полем адреса ]

[47:] T 67[<w1>] S [ запись сформированной инструкции, обнуление аккумулятора ]

[48:] A 99[<addr>]S [ загрузка в аккумулятор адреса 0-го элемента массива ]  
 [49:] L0L [ сдвиг аккумулятора на 1 разряд влево ]  
 [50:] A 66[<r2>]S [ прибавляем код инструкции с полем адреса 1 ]  
 [51:] T 66[<r2>]S [ запись сформированной инструкции, обнуление аккумулятора ]  
  
 [52:] A 99[<addr>]S [ загрузка в аккумулятор адреса 0-го элемента массива ]  
 [53:] L0L [ сдвиг аккумулятора на 1 разряд влево ]  
 [54:] A 69[<w2>]S [ прибавляем код инструкции с полем адреса 1 ]  
 [55:] T 69[<w2>]S [ запись сформированной инструкции, обнуление аккумулятора ]  
  
 [loop1:]  
 [56:] A 2[count:] S [ загружаем счетчик необработанных сдвигов массива ]  
 [57:] S 97[c1:] S [ уменьшаем на 1 ]  
 [58:] G 95[exit:] S [ если результат меньше 0, завершаем работу ]  
 [59:] T 2[count:] S [ обновляем значение счетчика и обнуляем аккумулятор ]  
  
 [loop2:]  
 [60:] A 4[length:] S [ загружаем счетчик необработанных сдвигов массива ]  
 [61:] S 97[c1:] S [ уменьшаем на 1 ]  
 [62:] G 79[loop1:] S [ если результат меньше 0, завершаем работу ]  
 [63:] T 4[length:] S [ обновляем значение счетчика и обнуляем аккумулятор ]  
  
 [64:] [r1:] A 0[addr:] S [ загрузка в аккумулятор значения из ячейки N ]  
 [65:] T 0 S [ запись этого значения в рабочую ячейку, обнуление аккумулятора ]  
 [66:] [r2:] A 1[addr +1] S [ загрузка в аккумулятор значения из ячейки N+1 ]  
 [67:] [w1:] T 0[addr:] S [ запись этого значения в ячейку с адресом N, обнуление аккумулятора ]  
 [68:] A 0 S [ загрузка в аккумулятор значения из ячейки 0 ]  
 [69:] [w2:] T 1[addr +1] S [ запись этого значения в ячейку с адресом N+1, обнуление аккумулятора ]  
  
 [70:] A 66[r2:] S [ загрузка в аккумулятор инструкции r2 ]  
 [71:] U 64[r1:] S [ запись этой инструкции вместо r1 ]  
 [72:] A 96[c2:] S [ прибавляем константу 2 ]  
 [73:] T 66[r2:] S [ запись полученной инструкции r2 ]  
  
 [74:] A 69[w2:] S [ загрузка в аккумулятор инструкции w2 ]  
 [75:] U 67[w1:] S [ запись этой инструкции вместо w1 ]  
 [76:] A 96[c2:] S [ прибавляем константу 2 ]  
 [77:] T 69[w2:] S [ запись полученной инструкции w2 ]  
 [78:] E 60[loop2:] S [ повторяем все операции; аккумулятор обнулен ]  
  
 [79:] T 5 S [acc = 0]  
 [80:] A 100[length:] S [ загрузка в аккумулятор длины обрабатываемого массива ]  
 [81:] T 4 S [ запись этого значения в ячейку 4 ]  
  
 [82:] A 64[<r1>]S [ загрузка в аккумулятор кода инструкции с нулевым полем адреса ]  
 [83:] S 98[<cLen>]S [ возвращаем "каретку" в начало массива ]  
 [84:] T 64[<r1>]S [ запись сформированной инструкции, обнуление аккумулятора ]  
  
 [85:] A 67[<w1>]S [ загрузка в аккумулятор кода инструкции с нулевым полем адреса ]  
 [86:] S 98[<cLen>]S [ возвращаем "каретку" в начало массива ]  
 [87:] T 67[<w1>]S [ запись сформированной инструкции, обнуление аккумулятора ]  
  
 [88:] A 66[<r2>]S [ загрузка в аккумулятор кода инструкции с первым полем адреса ]  
 [89:] S 98[<cLen>]S [ возвращаем "каретку" в начало массива ]  
 [90:] T 66[<r2>]S [ запись сформированной инструкции, обнуление аккумулятора ]

```

[91:] A 69[<w2>]S [ загрузка в аккумулятор кода инструкции с первым полем адреса ]
[92:] S 98[<cLen>]S [ возвращаем "каretку" в начало массива ]
[93:] T 69[<w2>]S [ запись сформированной инструкции, обнуление аккумулятора ]
[94:] E 56[loop1:] S [ повторяем все операции; аккумулятор обнулен ]

[exit:]
[95:] Z 0 S [ останов ]

[96:] [c2:] P 1 S [2]
[97:] [c1:] P 0 L [1]
[98:] [cLen:] P 0 S [0]

[ADDR:]
[99:] P 51 [<array>] S [ адрес начала массива ]
[LEN:]
[100:] P 2 S [ длина массива - 5 (0...0|0|0...101|1 = 5) ]
[COUNT:]
[101:] P 1 L [ 3 ] [ количество сдвигов в массиве чисел ]
[array:]
[102:] P 0 L [ 1 ]
[103:] P 1 S [ 2 ]
[104:] P 1 L [ 3 ]
[105:] P 2 S [ 4 ]
[106:] P 2 L [ 5 ]

```

#### 4. Работа программы Initial Orders 1

Исходный массив представлен в пункте 3 строки 102-106. Сдвиг осуществляется на 3 разряда влево. Результат представлен на рис. 1-5.

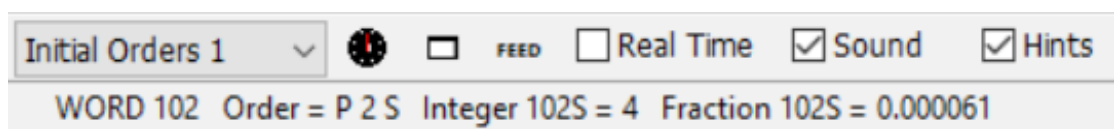


Рис. 1. Результат в ячейке 102.

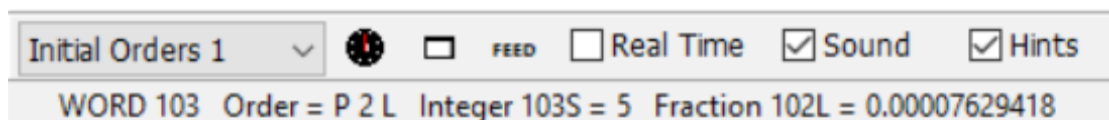


Рис. 2. Результат в ячейке 103.

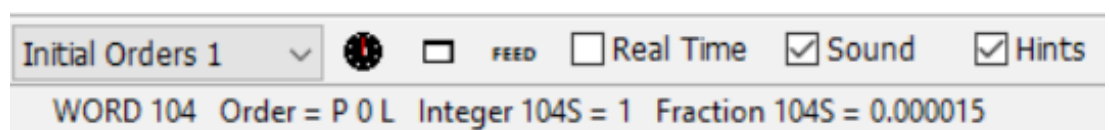


Рис. 3. Результат в ячейке 104.



Рис. 4. Результат в ячейке 105.

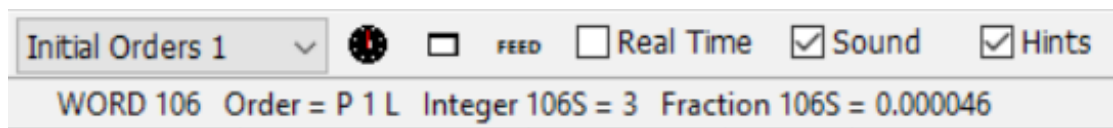


Рис. 5. Результат в ячейке 106.

Как видно из рис.1-5 программа работает верно. Исходный массив: [1, 2, 3, 4, 5] был сдвинут на 3 разряда влево и получен новый массив [4, 5, 1, 2, 3].

## 5. Программа Initial Orders 2

### Листинг 2: Программа для Initial Orders 2

[ T3: Разработать программу для EDSAC, реализующую циклический сдвиг массива чисел ]  
 [ на заданное количество разрядов влево, и предполагающую загрузчик IO2. ]  
 [ Массив хранится в ячейках 132-136 ]  
 [ Количество сдвигов хранится в ячейке 131 ]  
 [ результат сохраняется на месте исходного массива ячейки 132-136 ]  
 [ ячейки 0-4 рабочие ]

|                      |  |
|----------------------|--|
| T 56 K               | [ директива IO2, установка адреса загрузки ]                 |
| G K                  | [ директива IO2, фиксация начального адреса подпрограммы ]   |
| [ 0:] A 3 F          | [ пролог: формирование кода инструкции возврата в Асс ]      |
| [ 1:] T 58 @         | [ пролог: запись инструкции возврата ]                       |
| [ 2:] A 4 F          | [ загрузка в аккумулятор длины обрабатываемого массива ]     |
| [ 3:] U 62 [len:] @  | [ запись этого значения в ячейку 56+58 ]                     |
| [ 4:] L0D            | [ сдвиг аккумулятора на 1 разряд влево ]                     |
| [ 5:] T 61 [cLen:] @ | [ запись этого значения в ячейку 56+57 ]                     |
| [ 6:] A 0 F          | [ загрузка в аккумулятор адреса 0-го элемента массива ]      |
| [ 7:] A 26[r1] @     | [ прибавляем код инструкции с нулевым полем адреса ]         |
| [ 8:] T 26[r1] @     | [ запись сформированной инструкции, обнуление аккумулятора ] |
| [ 9:] A 0 F          | [ загрузка в аккумулятор адреса 0-го элемента массива ]      |
| [10:] A 28[r2] @     | [ прибавляем код инструкции с полем адреса 1 ]               |
| [11:] T 28[r2] @     | [ запись сформированной инструкции, обнуление аккумулятора ] |
| [12:] A 0 F          | [ загрузка в аккумулятор адреса 0-го элемента массива ]      |
| [13:] A 29[w1] @     | [ прибавляем код инструкции с нулевым полем адреса ]         |
| [14:] T 29[w1] @     | [ запись сформированной инструкции, обнуление аккумулятора ] |
| [15:] A 0 F          | [ загрузка в аккумулятор адреса 0-го элемента массива ]      |
| [16:] A 31[w2] @     | [ прибавляем код инструкции с полем адреса 1 ]               |
| [17:] T 31[w2] @     | [ запись сформированной инструкции, обнуление аккумулятора ] |

```

[loop1:]
[18:] A 2[count:] F      [ загружаем счетчик необработанных сдвигов массива ]
[19:] S 60[c1:] @       [ уменьшаем на 1 ]
[20:] G 57[exit:] @     [ если результат меньше 0, завершаем работу ]
[21:] T 2[count:] F     [ обновляем значение счетчика и обнуляем аккумулятор ]

[loop2:]
[22:] A 4[length:] F    [ загружаем счетчик необработанных сдвигов массива ]
[23:] S 60[c1:] @       [ уменьшаем на 1 ]
[24:] G 41[loop1:] @    [ если результат меньше 0, завершаем работу ]
[25:] T 4[length:] F    [ обновляем значение счетчика и обнуляем аккумулятор ]

[26:] [r1:] A 0[addr:] F [ загрузка в аккумулятор значения из ячейки N ]
[27:]      T 0 F        [ запись этого значения в рабочую ячейку, обнуление аккумулятора ]
[28:] [r2:] A 1[addr +1] F [ загрузка в аккумулятор значения из ячейки N+1 ]
[29:] [w1:] T 0[addr:] F  [ запись этого значения в ячейку с адресом N, обнуление аккумулятора ]
[30:]      A 0 F        [ загрузка в аккумулятор значения из ячейки 0 ]
[31:] [w2:] T 1[addr +1] F [ запись этого значения в ячейку с адресом N+1, обнуление
                                                                    аккумулятора ]

[32:] A 28[r2:] @       [ загрузка в аккумулятор инструкции r2 ]
[33:] U 26[r1:] @       [ запись этой инструкции вместо r1 ]
[34:] A 59[c2:] @       [ прибавляем константу 2 ]
[35:] T 28[r2:] @       [ запись полученной инструкции r2 ]

[36:] A 31[w2:] @       [ загрузка в аккумулятор инструкции w2 ]
[37:] U 29[w1:] @       [ запись этой инструкции вместо w1 ]
[38:] A 59[c2:] @       [ прибавляем константу 2 ]
[39:] T 31[w2:] @       [ запись полученной инструкции w2 ]
[40:] E 22[loop2:] @    [ повторяем все операции; аккумулятор обнулен ]

[41:] T 5 F            [acc = 0]
[42:] A 62[len:] @     [ загрузка в аккумулятор длины обрабатываемого массива ]
[43:] T 4 F            [ запись этого значения в ячейку 4 ]

[44:] A 26[<r1>]@      [ загрузка в аккумулятор кода инструкции с нулевым полем адреса ]
[45:] S 61[<cLen>]@    [ возвращаем "карьерку" в начало массива ]
[46:] T 26[<r1>]@      [ запись сформированной инструкции, обнуление аккумулятора ]

[47:] A 29[<w1>]@      [ загрузка в аккумулятор кода инструкции с нулевым полем адреса ]
[48:] S 61[<cLen>]@    [ возвращаем "карьерку" в начало массива ]
[49:] T 29[<w1>]@      [ запись сформированной инструкции, обнуление аккумулятора ]

[50:] A 28[<r2>]@      [ загрузка в аккумулятор кода инструкции с первым полем адреса ]
[51:] S 61[<cLen>]@    [ возвращаем "карьерку" в начало массива ]
[52:] T 28[<r2>]@      [ запись сформированной инструкции, обнуление аккумулятора ]

[53:] A 31[<w2>]@      [ загрузка в аккумулятор кода инструкции с первым полем адреса ]
[54:] S 61[<cLen>]@    [ возвращаем "карьерку" в начало массива ]
[55:] T 31[<w2>]@      [ запись сформированной инструкции, обнуление аккумулятора ]
[56:] E 18[loop1:] @   [ повторяем все операции; аккумулятор обнулен ]

[exit:]
[57:] T 0 F            [ обнуление аккумулятора ]

[ret:]
[58:] E 0 F            [ эпилог: инструкция возврата из подпрограммы ]

[const:]

```



```

[59:] [c2:] P 1 F [2]
[60:] [c1:] P 0 D [1]
[61:] [cLen:] P 0 F [0]
[62:] [len:] P 0 F [0]

G K [ директива IO2, фиксация начального адреса программы ]
[ 0:] Z 0 F
[ 1:] A 10 [ADDR:] @ [ адрес начала массива ]
[ 2:] T 0 F [ запись адреса массива в ячейку 0, обнуление аккумулятора ]
[ 3:] A 11 [<len>] @ [ длина массива ]
[ 4:] T 4 F [ запись длины массива в ячейку 4, обнуление аккумулятора ]
[ 5:] A 12 [COUNT] @ [ счетчик ]
[ 6:] T 2 F [ запись счетчика в ячейку 2, обнуление аккумулятора ]
[ 7:] A 7 @ [ вызов ]
[ 8:] G 56 [<sub>] F [/ подпрограммы ]
[ 9:] Z 0 F [ останов ]

[ADDR:]
[10:] P 13 [<array>] @ [ адрес массива = <Начало программы>+13 (см. ниже) ]
[LEN:]
[11:] P 2 F [ длина массива - 5 (0...0|0|0...101|1 = 5) ]
[COUNT:]
[12:] P 1 D [ 3 ] [ количество сдвигов в массиве чисел ]
[array:]
[13:] P 0 D [ 1 ]
[14:] P 1 F [ 2 ]
[15:] P 1 D [ 3 ]
[16:] P 2 F [ 4 ]
[17:] P 2 D [ 5 ]
EZ PF [ директива IO2, переход в исполнение ]

```

## 6. Работа программы Initial Orders 2

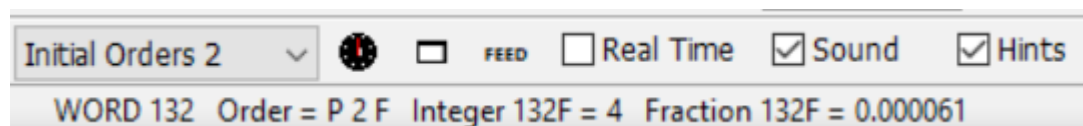


Рис. 6. Результат в ячейке 132.



Рис. 7. Результат в ячейке 133.

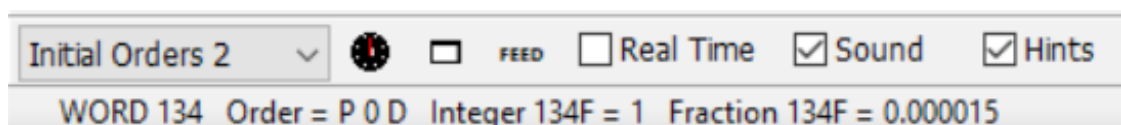


Рис. 8. Результат в ячейке 134.

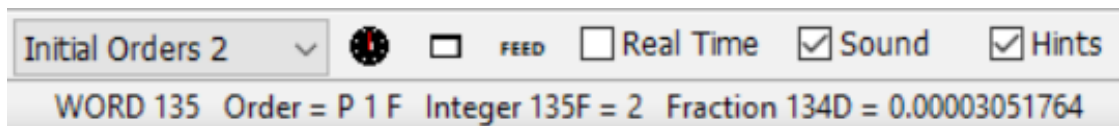


Рис. 9. Результат в ячейке 135.



Рис. 10. Результат в ячейке 136.

Как видно из рис.6-10 программа работает верно. Исходный массив: [1, 2, 3, 4, 5] был сдвинут на 3 разряда влево и получен новый массив [4, 5, 1, 2, 3].

### **Вывод**

В данной работе были получены навыки работы с программой EDSAC Simulator. Для указанного симулятора была успешно реализована программа сдвига массива чисел на заданное количество разрядов влево для разных режимов работы: Initial Orders 1 и Initial Orders 2.