

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: 0 , $1 \cdot 2^{-16}$, $2 \cdot 2^{-16}$, $3 \cdot 2^{-16}$, $4 \cdot 2^{-16}$, $5 \cdot 2^{-16}$. В двоичном представлении:

100: 0000000000000000

101: 0000000000000001

102: 0000000000000010

103: 0000000000000011

104: 0000000000000100

105: 0000000000000101

В целочисленной интерпретации, эти комбинации разрядов соответствуют числам 0, 1, 2, 3, 4, 5. После выполнения программы в тех же ячейках памяти должны располагаться числа: 2, 3, 5, 4, 0, 1 (осуществлен циклический сдвиг на 2 разряда влево).

Для реализации данной задачи мы будем брать элемент массива из ячейки и записывать его в предыдущую ячейку, предварительно первый элемент поместив в рабочую ячейку. На место последнего элемента положим элемент хранящийся в рабочей ячейки. Эта процедура будет продолжаться столько раз на сколько надо осуществить сдвиг.

3. Программа Initial Orders 1

Массив хранится в ячейках 66-75. Количество сдвигов хранится в ячейке 65, результат сохраняется на месте исходного массива ячейки 66-75. Рабочие ячейки: 0, 1.

Листинг 1: Программа для Initial Orders 1

```
[31] T 76 S
[32] Z 0 S      [ останов для отладки ]
[33] T 0 S      [ запись аккумулятора в рабочую ячейку 0, обнуление аккумулятора ]
[34] A 65 [<count>] S  [ загрузка в аккумулятор количества единиц сдвига ]
[35] T 1 S      [ запись этого значения в ячейку 1 ]
[36] [loop:] A 1 S  [ загружаем счетчик необработанных элементов массива ]
[37] S 64 [<c1>] S   [ уменьшаем на 1 ]
[38] G 63 [<exit>] S  [ если результат меньше 0, завершаем работу ]
[39] T 1 S      [ обновляем значение счетчика и обнуляем аккумулятор ]
[40] [r1:] A 66 S  [ загрузка в аккумулятор значения первого элемента массива ]
[41] T 0 S      [ запись этого значения в рабочую ячейку, обнуление аккумулятора ]
[42] A 67 S      [ считываем второй элемент массива ]
[43] T 66 S      [ кладем на место первого элемента ]
[44] A 68 S      [ считываем третий элемент массива ]
[45] T 67 S      [ кладем на место второго элемента ]
[46] A 69 S      [ считываем четвертый элемент массива ]
[47] T 68 S      [ кладем на место третьего элемента ]
[48] A 70 S      [ считываем пятый элемент массива ]
[49] T 69 S      [ кладем на место четвертого элемента ]
[50] A 71 S      [ считываем шестой элемент массива ]
[51] T 70 S      [ кладем на место пятого элемента ]
[52] A 72 S      [ считываем седьмой элемент массива ]
[53] T 71 S      [ кладем на место шестого элемента ]
[54] A 73 S      [ считываем восьмой элемент массива ]
[55] T 72 S      [ кладем на место седьмого элемента ]
[56] A 74 S      [ считываем девятый элемент массива ]
[57] T 73 S      [ кладем на место восьмого элемента ]
[58] A 75 S      [ считываем десятый элемент массива ]
[59] T 74 S      [ кладем на место девятого элемента ]
[60] A 0 S       [ считываем первый элемент массива ]
[61] T 75 S      [ кладем на место десятого элемента ]
[62] E 36 [<loop>] S  [ повторяем все операции; аккумулятор обнулен ]
```

[63]	[exit:] Z 0 S	[останов]
[64]	[c1:] P 0 L	[1]
[65]	[count:] P 1 L	[3]
[66]	P 0 L	[1]
[67]	P 1 S	[2]
[68]	P 1 L	[3]
[69]	P 2 S	[4]
[70]	P 2 L	[5]
[71]	P 3 S	[6]
[72]	P 3 L	[7]
[73]	P 4 S	[8]
[74]	P 4 L	[9]
[75]	P 5 S	[10]

4. Работа программы Initial Orders 1

Исходный массив представлен в пункте 3 строки 66-75. Сдвиг осуществляется на 3 разряда влево. Результат представлен на рис. 1-10.

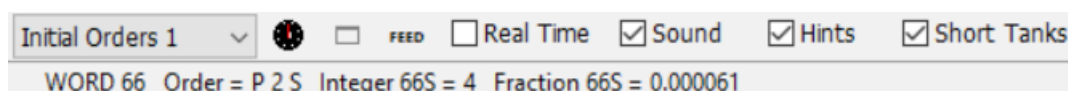


Рис. 1. Результат в ячейке 66.

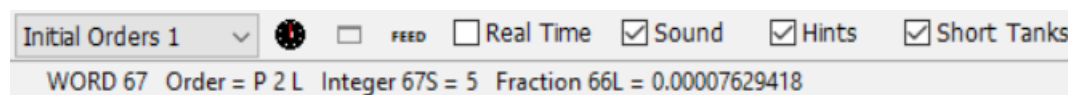


Рис. 2. Результат в ячейке 67.

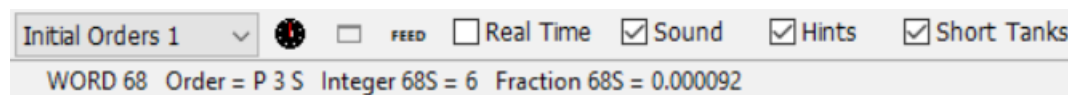


Рис. 3. Результат в ячейке 68.

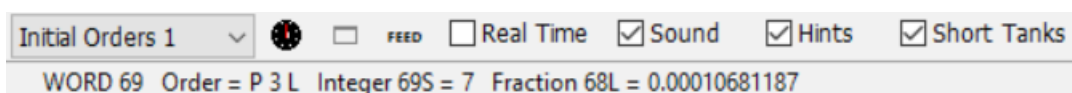


Рис. 4. Результат в ячейке 69.

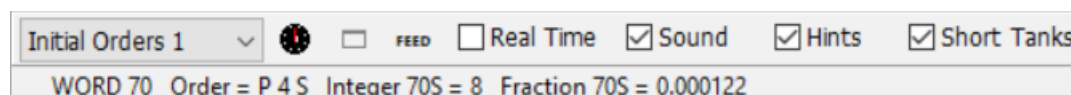


Рис. 5. Результат в ячейке 70.

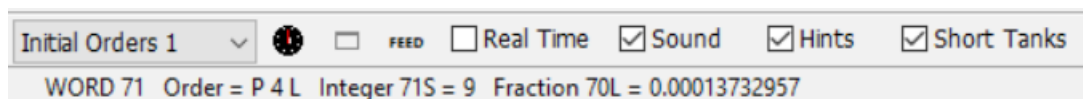


Рис. 6. Результат в ячейке 71.

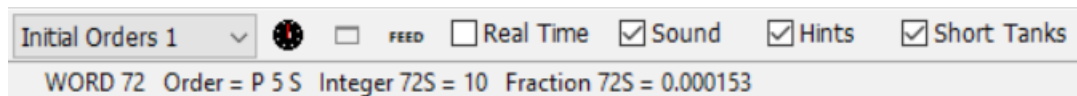


Рис. 7. Результат в ячейке 72.

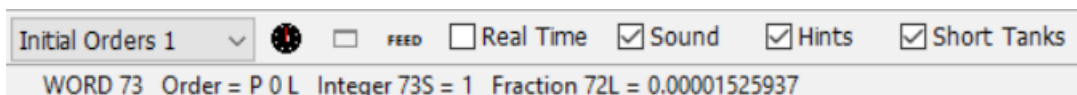


Рис. 8. Результат в ячейке 73.

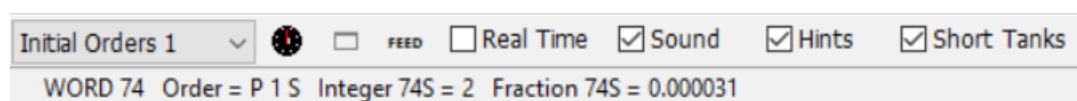


Рис. 9. Результат в ячейке 74.

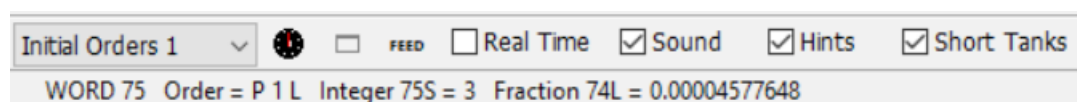


Рис. 10. Результат в ячейке 75.

Как видно из рис.1-10 программа работает верно. Исходный массив: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] был сдвинут на 3 разряда влево и получен новый массив [4, 5, 6, 7, 8, 9, 10, 1, 2, 3].

5. Программа Initial Orders 2

Массив хранится в ячейках 136-142. Количество сдвигов хранится в ячейке 135, результат сохраняется на месте исходного массива ячейки 136-142. Рабочие ячейки: 0-2.

Листинг 2: Программа для Initial Orders 2

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 61 @	[пролог: запись инструкции возврата]
[2:] A 0 F	[загрузка в аккумулятор адреса 0-го элемента массива]
[3:] A 26[r1] @	[прибавляем код инструкции с нулевым полем адреса]
[4:] T 26[r1] @	[запись сформированной инструкции, обнуление аккумулятора]
[5:] A 0 F	[загрузка в аккумулятор адреса 0-го элемента массива]

[6:]	A 28[r2] @	[прибавляем код инструкции с полем адреса 1]
[7:]	S 63[c2] @	[уменьшаем код инструкции на 2]
[8:]	T 28[r2] @	[запись сформированной инструкции, обнуление аккумулятора]
[9:]	A 0 F	[загрузка в аккумулятор адреса 0-го элемента массива]
[10:]	A 29[w1] @	[прибавляем код инструкции с нулевым полем адреса]
[11:]	T 29[w1] @	[запись сформированной инструкции, обнуление аккумулятора]
[12:]	A 0 F	[загрузка в аккумулятор адреса 0-го элемента массива]
[13:]	A 31[w2] @	[прибавляем код инструкции с полем адреса 1]
[14:]	S 63[c2] @	[уменьшаем код инструкции на 2]
[15:]	T 31[w2] @	[запись сформированной инструкции, обнуление аккумулятора]
[16:]	A 1 F	[загрузка в аккумулятор длины массива]
[17:]	T 66 @	[запись этого значения в ячейку 122]
[loop1:]		
[18:]	A 2 F	[загружаем счетчик необработанных элементов массива]
[19:]	S 62[c1] @	[уменьшаем на 1]
[20:]	G 60[<exit>] @	[если результат меньше 0, завершаем работу]
[21:]	T 2 F	[обновляем значение счетчика и обнуляем аккумулятор]
[loop2:]		
[22:]	A 66 @	[загружаем счетчик необработанных элементов массива]
[23:]	S 62[<c1>] @	[уменьшаем на 1]
[24:]	G 41 [<exit>] @	[если результат меньше 0, завершаем работу]
[25:]	T 66 @	[обновляем значение счетчика и обнуляем аккумулятор]
[26:] [r1:]	A 0 F	[загрузка в аккумулятор значения из ячейки N]
[27:]	T 0 F	[запись этого значения в рабочую ячейку, обнуление аккумулятора]
[28:] [r2:]	A 1 F	[загрузка в аккумулятор значения из ячейки N+1]
[29:] [w1:]	T 0 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 F	[запись этого значения в ячейку с адресом N+1, обнуление аккумулятора]
[32:]	A 28[<r2>] @	[загрузка в аккумулятор кода инструкции r2]
[33:]	U 26[<r1>] @	[запись инструкции поле r1]
[34:]	S 63[<c2>] @	[уменьшение кода инструкции]
[35:]	T 28[<r2>] @	[запись сформированной инструкции, обнуление аккумулятора]
[36:]	A 31[<w2>] @	[загрузка в аккумулятор кода инструкции w2]
[37:]	U 29[<w1>] @	[запись инструкции поле w1]
[38:]	S 63[<c2>] @	[уменьшение кода инструкции]
[39:]	T 31[<w2>] @	[запись сформированной инструкции, обнуление аккумулятора]
[40:]	E 22[<loop2:>] @	[повторяем все операции; аккумулятор обнулен]
[41:]	A 65 @	[загрузка в аккумулятор длины массива]
[42:]	T 66 @	[запись этого значения в ячейку 122]
[43:]	A 64[c6] @	[загрузка в аккумулятор константы 6]
[44:]	L 0 D	
[45:]	A 26[r1] @	[прибавляем код инструкции]

[46:] T 26[r1] @ [запись сформированной инструкции, обнуление аккумулятора]
 [47:] A 64[c6] @ [загрузка в аккумулятор константы 6]
 [48:] L O D
 [49:] A 28[r2] @ [прибавляем код инструкции]
 [50:] T 28[r2] @ [запись сформированной инструкции, обнуление аккумулятора]
 [51:] A 64[c6] @ [загрузка в аккумулятор константы 6]
 [52:] L O D
 [53:] A 29[w1] @ [прибавляем код инструкции]
 [54:] T 29[w1] @ [запись сформированной инструкции, обнуление аккумулятора]
 [55:] A 64[c6] @ [загрузка в аккумулятор константы 6]
 [56:] L O D
 [57:] A 31[w2] @ [прибавляем код инструкции]
 [58:] T 31[w2] @ [запись сформированной инструкции, обнуление аккумулятора]
 [59:] E 18[loop1:] @ [повторяем все операции; аккумулятор обнулен]
 [exit:]
 [60:] T 0 F [обнуление аккумулятора]
 [ret:]
 [61:] E 0 F [эпилог: инструкция возврата из подпрограммы]
 [62:] [c1:] P 0 D [константа 1]
 [63:] [c2:] P 1 F [константа 2]
 [64:] [c6:] P 3 F [константа 6]
 [LEN]
 [65:] P 3 D [константа 7]
 [66:] P 1 F
 G K [директива IO2, фиксация начального адреса программы]
 [0:] Z 0 F
 [1:] A 10 [<addr>] @ [адрес конца массива]
 [2:] T 0 F [запись адреса массива в ячейку 0, обнуление аккумулятора]
 [3:] A 11 [<len>] @ [длина массива]
 [4:] T 1 F [запись длины массива в ячейку 1, 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]
 [ADDR0:]
 [10:] P 19 [<array>] @ [адрес конца массива]
 [LEN:]
 [11:] P 3 D [длина массива - 7 (0...0|0|0...101|1 = 7)]
 [COUNT:]
 [12:] P 2 D [3]
 [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]
[18:] P 3 F [6]
[19:] P 3 D [7]
EZ PF [директива IO2,переход в исполнению]

6. Работа программы Initial Orders 2



Рис. 11. Результат в ячейке 136.



Рис. 12. Результат в ячейке 137.

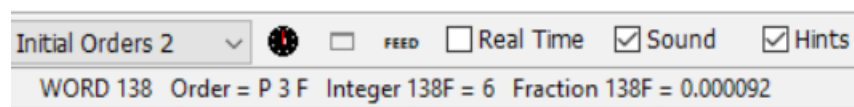


Рис. 13. Результат в ячейке 138.

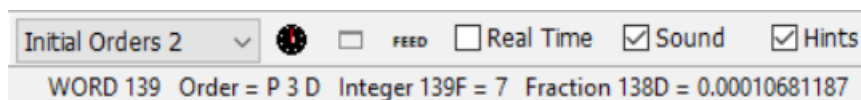


Рис. 14. Результат в ячейке 139.



Рис. 15. Результат в ячейке 140.



Рис. 16. Результат в ячейке 141.

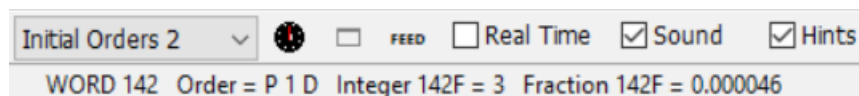


Рис. 17. Результат в ячейке 142.

Как видно из рис.11-17 программа работает верно. Исходный массив: [1, 2, 3, 4, 5, 6, 7] был сдвинут на 3 разряда влево и получен новый массив [4, 5, 6, 7, 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.