

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

## Лабораторная работа № 2

Программирование EDSAC

по дисциплине «Низкоуровневое программирование»

Выполнил  
студент гр. 3530901/00002

\_\_\_\_\_  
(подпись)

Тарсуков Ф.Д.

Руководитель

\_\_\_\_\_  
(подпись)

Степанов Д.С.

«\_\_» \_\_\_\_\_ 2021 г.

Санкт-Петербург  
2021

## **Задача**

Разработать универсальный алгоритм с использованием IO1 и IO2, формирующий последовательность чисел в коде Грея. Число разрядов определяется и задается пользователем.

## **Вариант задания**

Вариант: 12 - Формирование последовательности чисел в коде Грея заданной разрядности.

## **Выполнение работы**

IO1: Адрес числа разрядов - 34 строка

IO2: Адрес числа разрядов – 12 строка программы

Для получения заданного числа в коде Грея необходимо сложить по модулю два (операция “исключающее или”) текущее число с самим собой, сдвинутым на разряд вправо. Для получения последовательности необходимо в пределах заданного числа разрядов пройти циклом с шагом 1 от нуля до максимального числа. Так как операция вывода в EDSAC предусматривает лишь вывод старших пяти бит числа в десятичной системе счисления, необходимо выводить каждое число поразрядно для получения корректного результата.

Код программы Initial Orders 1 с комментариями:

[31] T 125 S

[32] O 38 S [сменили регистр на цифровой]

[33] E 41 S [в начало кода]

[34] P 1 L [число разрядов]

[35] P S [0]

[36] P L [1]

[37] & S [новая строка]

[38] # S [смена регистра на цифровой]

[39] L 0 S [шаблон оператора сдвига влево]

[40] R 0 S [шаблон оператора сдвига вправо]

[41] A 34 S [берем число разрядов на проверку >0]

[42] S 36 S [вычитаем единицу]

[43] G 124 S [при отрицательности окончание программы]

[44] T 0 S [очистили аккумулятор]

[45] A 35 S [0]

[46] T 1 S [изменяемое число]

[47] A 36 S [1]

[48] T 2 S [записываем в счетчик]

[49] A 34 S [число разрядов]

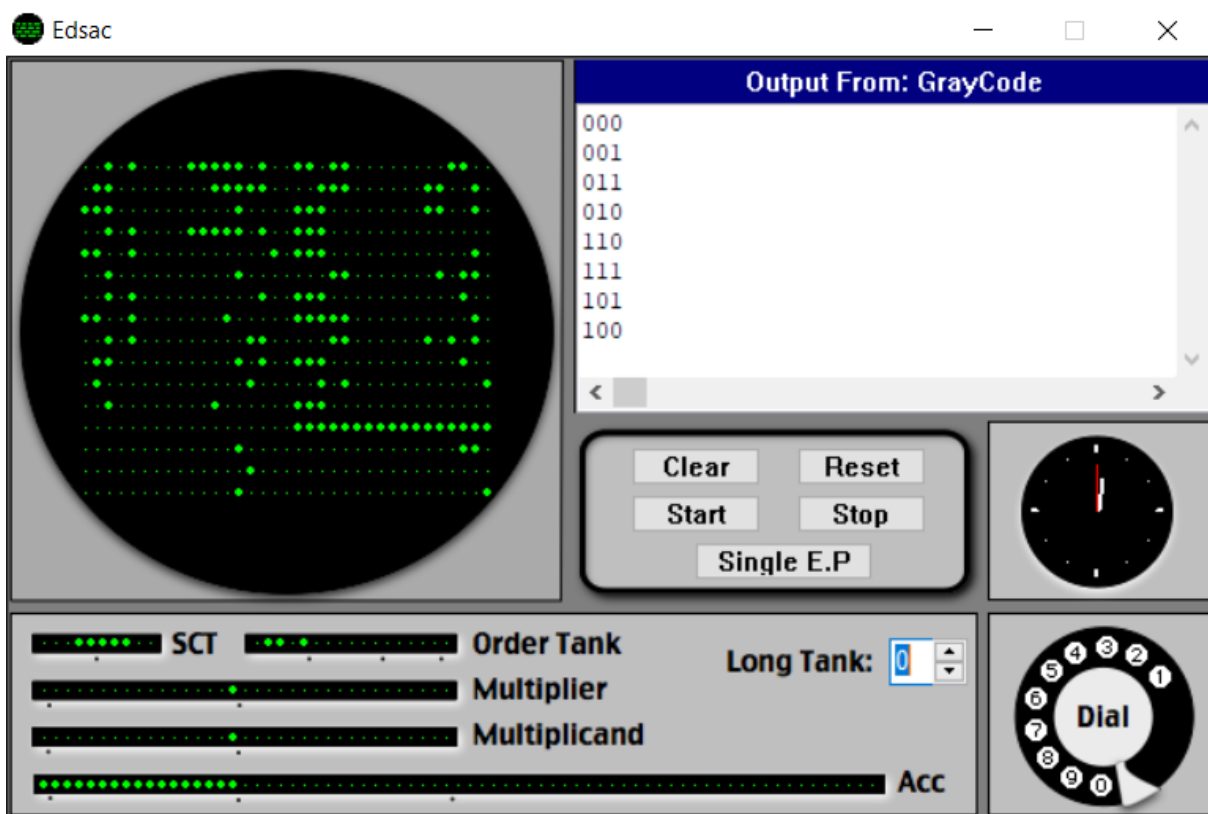
[50] S 36 S [вычитаем единицу]  
[51] T 0 S [loop 1: формируем счетчик]  
[52] A 2 S [счетчик]  
[53] L 0 L [увеличиваем в два раза на каждый разряд]  
[54] T 2 S [записываем]  
[55] A 114 S [берем оператор сдвига вправо]  
[56] S 40 S [вычитаем опкод]  
[57] L 0 L [сдвигаем]  
[58] A 40 S [возвращаем опкод]  
[59] T 114 S [записываем]  
[60] A 100 S [аналогично со сдвигом влево]  
[61] S 39 S  
[62] R 0 L  
[63] A 39 S  
[64] T 100 S  
[65] A 0 S  
[66] S 36 S [1]  
[67] E 51 S [пока локальный счетчик положителен]  
[68] T 0 S [очистили аккумулятор]

[69] A 114 S [берем оператор сдвига вправо]  
[70] S 40 S [вычитаем опкод]  
[71] R 0 L [сдвигаем]  
[72] A 40 S [возвращаем опкод]  
[73] T 114 S [записываем]

[74] A 2 S [счетчик]  
[75] U 5 S [заготовка для маски]  
[76] S 36 S [уменьшили счетчик на 1]  
[77] T 2 S [loop 2: формируем из числа число в коде Грея]  
[78] A 1 S [взяли текущее число]  
[79] R 0 L [сдвинули на разряд вправо]  
[80] T 3 S [записали]  
[81] H 1 S [добавили изначальное число в умножающий регистр]  
[82] C 3 S [умножили логически на сдвинутое]  
[83] L 0 L [умножили на два]  
[84] T 4 S [записали удвоенное логическое произведение]  
[85] A 3 S [взяли сдвинутое число]  
[86] A 1 S [прибавили изначальное]  
[87] S 4 S [вычли удвоенное логическое произведение]  
[88] T 3 S [записали в 3 ячейку]

[89] A 5 S [взяли заготовку для маски]  
[90] T 0 S [записали в нулевую ячейку для изменений]  
[91] A 34 S [взяли число разрядов]

[92] S 36 S [уменьшили на единицу]  
[93] S 36 S [loop 3: счетчик разрядов вывода]  
[94] T 6 S  
[95] A 0 S [взяли маску]  
[96] R 0 L [сдвинули на разряд вправо]  
[97] T 0 S [записали маску]  
[98] H 0 S [записали маску в умножающий регистр]  
[99] C 3 S [логически умножили на выводимое число]  
[100] L 512 S [сдвинули влево для вывода]  
[101] L 1 S [сдвинули еще левее]  
[102] T 7 S [записали]  
[103] O 7 S [вывели разряд выводимого числа]  
  
[104] A 101 S [взяли оператор сдвига]  
[105] S 39 S [вычли опкод]  
[106] L 0 L [увеличили на разряд]  
[107] A 39 S [вернули опкод]  
[108] T 101 S [перезаписали]  
[109] A 6 S  
[110] E 93 S [возвращаемся для вывода следующих разрядов]  
  
[111] T 6 S  
[112] A 101 S [взяли оператор сдвига]  
[113] S 39 S [вычли опкод]  
[114] R 0 L [вернули к изначальному значению]  
[115] A 39 S [вернули опкод]  
[116] T 101 S [перезаписали]  
[117] O 37 S [новая строка]  
[118] A 1 S [взяли изменяемое число]  
[119] A 36 S [увеличили на 1]  
[120] T 1 S [перезаписали]  
[121] A 2 S [взяли счетчик]  
[122] S 36 S [счетчик -1]  
[123] E 77 S [loop 2: выполняем, пока не кончится счетчик]  
[124] Z 0 S



Аналогичный алгоритм используется для Initial Orders 2 и в целом подпрограмма базируется на изменённой версии кода для Initial Orders 1. Также разработана вызывающая ее тестовая программа. Количество разрядов передается через ячейку памяти с фиксированным адресом (1 ячейку).

Код программы Initial Orders 2 с комментариями:

T 56 K [установка адреса загрузки]  
 GK [фиксация начального адреса подпрограммы]  
 [0] A 3 F [код инструкции возврата в аккумулятор]  
 [1] T 87 @ [запись инструкции возврата]  
  
 [2] X 0 @ [отладка]  
 [3] X 0 @ [отладка]  
  
 [4] A 1 F [берем число разрядов на проверку >0]  
 [5] S 89 @ [вычитаем единицу]  
 [6] G 87 @ [при отрицательности окончание программы]  
 [7] T 90 @ [очистили аккумулятор]  
  
 [8] A 88 @ [0]  
 [9] T 91 @ [изменяемое число]  
 [10] A 89 @ [1]  
 [11] T 92 @ [записываем в счетчик]

[12] A 1 F [число разрядов]  
[13] S 89 @ [вычитаем единицу]  
[14] T 90 @ [loop 1: формируем счетчик]  
[15] A 92 @ [счетчик]  
[16] L 0 D [увеличиваем в два раза на каждый разряд]  
[17] T 92 @ [записываем]  
[18] A 77 @ [берем оператор сдвига вправо]  
[19] S 3 F [вычитаем опкод]  
[20] L 0 D [сдвигаем]  
[21] A 3 F [возвращаем опкод]  
[22] T 77 @ [записываем]  
[23] A 63 @ [аналогично со сдвигом влево]  
[24] S 2 F  
[25] R 0 D  
[26] A 2 F  
[27] T 63 @  
[28] A 90 @  
[29] S 89 @ [1]  
[30] E 14 @ [пока локальный счетчик положителен]  
[31] T 90 @ [очистили аккумулятор]

[32] A 77 @ [берем оператор сдвига вправо]  
[33] S 3 F [вычитаем опкод]  
[34] R 0 D [сдвигаем]  
[35] A 3 F [возвращаем опкод]  
[36] T 77 @ [записываем]

[37] A 92 @ [счетчик]  
[38] U 95 @ [заготовка для маски]  
[39] S 89 @ [уменьшили счетчик на 1]  
[40] T 92 @ [loop 2: формируем из числа число в коде Грея]  
[41] A 91 @ [взяли текущее число]  
[42] R 0 D [сдвинули на разряд вправо]  
[43] T 93 @ [записали]  
[44] H 91 @ [добавили изначальное число в умножающий регистр]  
[45] C 93 @ [умножили логически на сдвинутое]  
[46] L 0 D [умножили на два]  
[47] T 94 @ [записали удвоенное логическое произведение]  
[48] A 93 @ [взяли сдвинутое число]  
[49] A 91 @ [прибавили изначальное]  
[50] S 94 @ [вычли удвоенное логическое произведение]  
[51] T 93 @ [записали в 3 ячейку]

[52] A 95 @ [взяли заготовку для маски]  
[53] T 90 @ [записали в нулевую ячейку для изменений]

[54] A 1 F [взяли число разрядов]  
[55] S 89 @ [уменьшили на единицу]  
[56] S 89 @ [loop 3: счетчик разрядов вывода]  
[57] T 96 @  
[58] A 90 @ [взяли маску]  
[59] R 0 D [сдвинули на разряд вправо]  
[60] T 90 @ [записали маску]  
[61] H 90 @ [записали маску в умножающий регистр]  
[62] C 93 @ [логически умножили на выводимое число]  
[63] L 512 F [сдвинули влево для вывода]  
[64] L 1 F [сдвинули еще левее]  
[65] T 97 @ [записали]  
[66] O 97 @ [вывели разряд выводимого числа]  
  
[67] A 64 @ [взяли оператор сдвига]  
[68] S 2 F [вычли опкод]  
[69] L 0 D [увеличили на разряд]  
[70] A 2 F [вернули опкод]  
[71] T 64 @ [перезаписали]  
[72] A 96 @  
[73] E 56 @ [возвращаемся для вывода следующих разрядов]  
  
[74] T 96 @  
[75] A 64 @ [взяли оператор сдвига]  
[76] S 2 F [вычли опкод]  
[77] R 0 D [вернули к изначальному значению]  
[78] A 2 F [вернули опкод]  
[79] T 64 @ [перезаписали]  
[80] O 4 F [новая строка]  
[81] A 91 @ [взяли изменяемое число]  
[82] A 89 @ [увеличили на 1]  
[83] T 91 @ [перезаписали]  
[84] A 92 @ [взяли счетчик]  
[85] S 89 @ [счетчик -1]  
[86] E 40 @ [loop 2: выполняем, пока не кончится счетчик]  
[87] E 0 F  
[88] P F [0]  
[89] P D [1]  
[90] P F [для записи переменных]  
[91] P F [для записи переменных]  
[92] P F [для записи переменных]  
[93] P F [для записи переменных]  
[94] P F [для записи переменных]  
[95] P F [для записи переменных]  
[96] P F [для записи переменных]

[97] P F [для записи переменных]

GK

[0] X 14 @ [сменили регистр на цифровой]

[1] A 12 @ [переписали число разрядов в 1 ячейку]

[2] T 1 F

[3] A 15 @ [переписали опкод L во 2 ячейку]

[4] T 2 F

[5] A 16 @ [переписали опкод R в 3 ячейку]

[6] T 3 F

[7] A 13 @ [переписали оператор сдвига строки]

[8] T 4 F

[9] A 7 @

[10] G 56 F

[11] Z 0 F

[12] P 2 D [число разрядов]

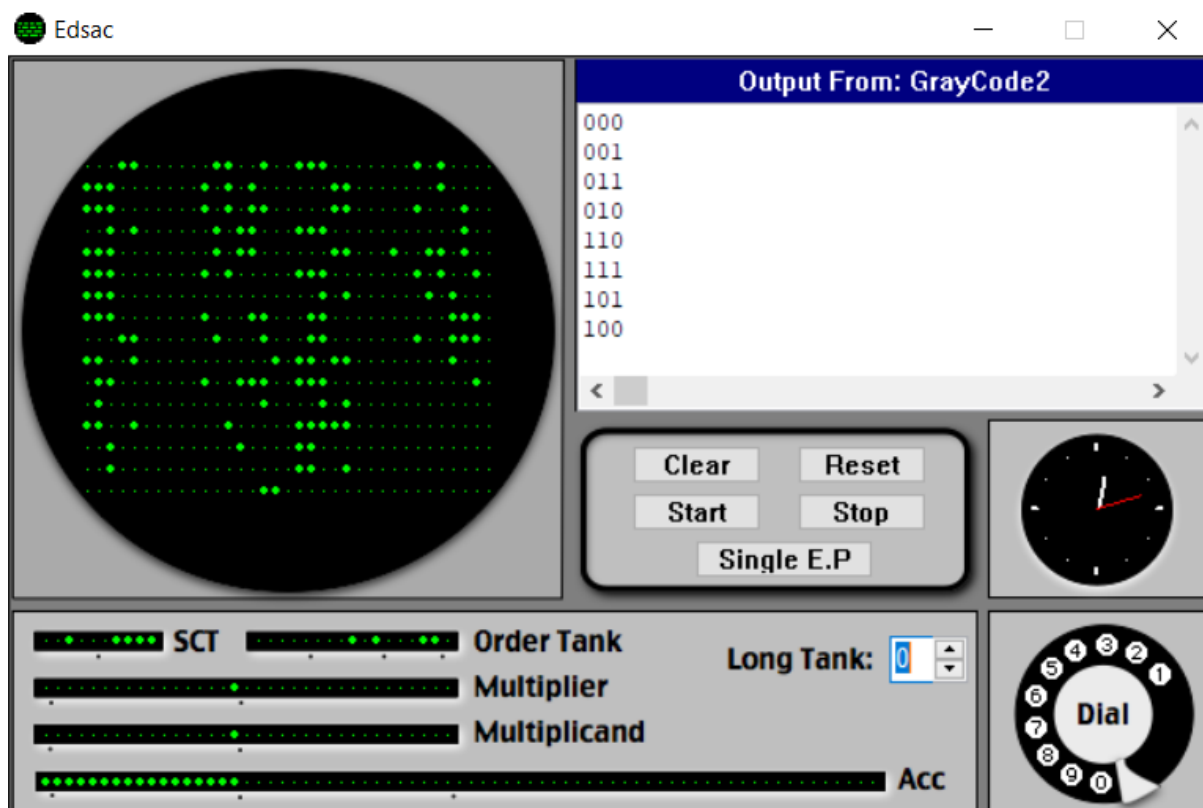
[13] & F [новая строка]

[14] # F [смена регистра на цифровой]

[15] L 0 F [шаблон оператора сдвига влево]

[16] R 0 F [шаблон оператора сдвига вправо]

[17] EZ PF





## **Вывод**

В ходе выполнения лабораторной работы была разработана программа для EDSAC, формирующая последовательность чисел в коде Грея, предполагающая загрузчики Initial Orders 1 и Initial Orders 2.