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. 3530901/90004

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Антунович П.Ю.

(подпись)

Руководитель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Алексюк А.О.

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Санкт-Петербург   
2021

**Постановка задачи**

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.

**Вариант задания:** 13) расчет значения многочлена по схеме Горнера с «длинным» результатом

**Общие сведения**

Схема Горнера для деления многочлена — это алгоритм упрощения вычисления значения многочлена f(x) при определённой величине x = x0 методом деления многочлена на одночлены (многочлены 1ой степени). Каждый одночлен включает в себя максимум один процесс умножения и один процесс сложения. Результат, полученный из одного одночлена, прибавляют к результату, полученному от следующего одночлена и так далее в аккумулятивной манере. Такой процесс деления также называют синтетическим делением.

Т.е. если задан многочлен 4-й степени:

То его можно представить в форме:

Где x = x0 – значение, при котором мы ищем значение многочлена.

Наглядно алгоритм можно представить в виде таблицы:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| счетчик | 4 | 3 | 2 | 1 | 0 |
| вычисление |  |  |  |  |  |
| результат |  |  |  |  |  |

Значение b0 и является искомым.

**Ход работы**

Напишем программу, реализующую метод Горнера с «длинным» результатом. Тестировать код будем на примере многочлена:

Код программы для загрузчика Initial Orders 1 с комментариями представлен ниже:

[31:] T 76 S

[32:] X 0 S [ сменить на Z 0 S для отладки ]

[33:] A 69 [<power>] S [ загрузка в аккумулятор степени многочлена ]

[34:] T 1 S [ запись степени многочлена в ячейку 1 ]

[35:] A 71 [<a4>] S [ загрузка старшего коэффициента в аккумулятор ]

[36:] T 2 S [ запись старшего коэффициента в рабочую ячейку 2 ]

[37:] H 68 [<c1>] S [ запись 1 в умножающий регистр ]

[38:] V 2 S [ умножение числа из ячейки 2 на число в умножающем регистре ]

[39:] R 1 S [ сдвиг числа вправо на 2 бита, для коррекции умножения ]

[40:] T 2 L [ запись старшего коэффициента в длинную ячейку ]

[41:] A 70 [<addr>] S [ загрузка в аккумулятор адреса 1-го элемента массива ]

[42:] L 0 L [ сдвиг аккумулятора, для коррекции ]

[43:] A 54 [<r1>] S [ прибавляем код инструкции с нулевым полем адреса ]

[44:] T 54 [<r1>] S [ запись сформированной инструкции, обнуление аккумулятора ]

[loop:]

[45:] A 1 S [ загружаем в аккумулятор счётчик необработанных элементов массива ]

[46:] S 68 S [ вычитание константы = 1 ]

[47:] G 66 [<exit>] S [ if (acc < 0) end ]

[48:] T 1 S [ обновляем значение счетчика и обнуляем аккумулятор ]

[49:] H 67 [<x0>] S [ запись x0 в умножающий регистр ]

[50:] V 2 L [ умножение числа из ячейки 2 на число в умножающем регистре ]

[51:] L 1024 S [ сдвиг числа влево на 12 бит, для коррекции умножения ]

[52:] L 4 S [ сдвиг числа влево на 4 бита, для коррекции умножения ]

[53:] T 2 L [ запись результата умножения в длинную ячейку ]

[54 r1:] A 0 S [ загрузка в аккумулятор значения из ячейки 0 ]

[55:] T 4 S [ запись в рабочую ячейку 4 ]

[56:] H 68 [<c1>] S [ запись 1 в умножающий регистр ]

[57:] V 4 S [ умножение числа из ячейки 4 на число в умножающем регистре ]

[58:] R 1 S [ сдвиг числа вправо на 2 бита, для коррекции умножения ]

[59:] A 2 L [ добавление числа из "длинной" ячейки 2 в аккумулятор ]

[60:] T 2 L [ запись этого значения в "длинную" ячейку 2, обнуление аккумулятора ]

[61:] A 68 [<c1>] S [ загрузка в аккумулятор значение константы = 1 ]

[62:] L 0 L [ сдвиг аккумулятора, для коррекции ]

[63:] A 54 [<r1>] S [ прибавляем код инструкции, исполненной на предыдущем шаге ]

[64:] T 54 [<r1>] S [ записываем сформированную инструкцию в память ]

[65:] E 45 [<loop>] S [ повторяем все операции; аккумулятор обнулен ]

[66 exit:] Z 0 S [ точка остановки, завершение программы ]

[67:] P 1 S [ x0 = 2 - значение, по которому мы вычисляем значение многочлена ]

[68 const 1:] P 0 L [ c1 = 1 ]

[69 power:] P 2 S [ степень многочлена ]

[70 addr:] P 36 S [ адрес 1-го элемента массива ]

[71:] P 0 L [ a4 = 1 ]

[72:] P 1 L [ a3 = 3 ]

[73:] P 2 L [ a2 = 5 ]

[74:] P 3 L [ a1 = 7 ]

[75:] P 4 L [ a0 = 9 ]

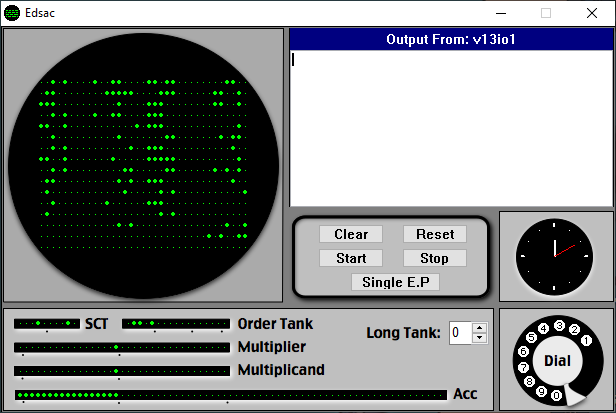


Рис. 1 результаты работы программы

Как мы видим, в «длинной» ячейке 2 находится верный ответ 83.

Код для IO2 практически идентичен коду для IO1 за несколькими отличиями, связанными с особенностями загрузчика.

T 56 K

G K

[0:] A 3 F [ пролог: формирование кода инструкции возврата в Acc]

[1:] T 31 [<ret>] @ [ пролог: запись инструкции возврата ]

[2:] H 33 [<c1>] @ [ запись 1 в умножающий регистр ]

[3:] V 2 F [ умножение числа из ячейки 2 на число в умножающем регистре ]

[4:] R 1 F [ сдвиг числа вправо на 2 бита, для коррекции умножения ]

[5:] T 2 D [ запись старшего коэффициента в длинную ячейку ]

[6:] A 1 [<addr>] F [ загрузка в аккумулятор адреса 1-го элемента массива ]

[7:] A 18 [<r1>] @ [ прибавляем код инструкции с нулевым полем адреса ]

[8:] T 18 @ [ запись сформированной инструкции, обнуление аккумулятора ]

[loop:]

[9:] A 0 F [ загружаем в аккумулятор счётчик необработанных элементов массива ]

[10:] S 33 @ [ вычитание константы = 1 ]

[11:] G 30 [<exit>] @ [ if (acc < 0) end ]

[12:] T 0 F [ обновляем значение счетчика и обнуляем аккумулятор ]

[13:] H 32 @ [ запись x0 в умножающий регистр ]

[14:] V 2 D [ умножение числа из ячейки 2 на число в умножающем регистре ]

[15:] L 1024 F [ сдвиг числа влево на 12 бит, для коррекции умножения ]

[16:] L 4 F [ сдвиг числа влево на 4 бита, для коррекции умножения ]

[17:] T 2 D [ запись результата умножения в длинную ячейку ]

[18 r1:] A 0 F [ загрузка в аккумулятор значения из ячейки 0 ]

[19:] T 4 F [ запись в рабочую ячейку 4 ]

[20:] H 33 @ [ запись 1 в умножающий регистр ]

[21:] V 4 F [ умножение числа из ячейки 4 на число в умножающем регистре ]

[22:] R 1 F [ сдвиг числа вправо на 2 бита, для коррекции умножения ]

[23:] A 2 D [ добавление числа из "длинной" ячейки 2 в аккумулятор ]

[24:] T 2 D [ запись этого значения в "длинную" ячейку 2, обнуление аккумулятора ]

[25:] A 33 @ [ загрузка в аккумулятор значение константы = 1 ]

[26:] L 0 D

[27:] A 18 [<r1>] @ [ прибавляем код инструкции, исполненной на предыдущем шаге ]

[28:] T 18 [<r1>] @ [ записываем сформированную инструкцию в память ]

[29:] E 9 [<loop>] @ [ повторяем все операции; аккумулятор обнулен ]

[30 exit:] T 0 F [ точка остановки, завершение программы ]

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

[32:] P 1 F [ x0 - значение, по которому мы вычисляем значение многочлена ]

[33 const 1:] P 0 D [ c1 = 1 ]

G K

[0:] X 0 F

[1:] A 10 [<power>] @ [ загрузка в аккумулятор степени многочлена ]

[2:] T 0 F [ запись степени многочлена в ячейку 0 ]

[3:] A 11 [<addr>] @ [ загрузка в аккумулятор адреса массива ]

[4:] T 1 F [ запись адреса массива в ячейку 1]

[5:] A 12 [<a4>] @ [ загрузка старшего коэффициента в аккумулятор ]

[6:] T 2 F [ запись старшего коэффициента в рабочую ячейку 2 ]

[7:] A 7 @ [\Запуск]

[8:] G 56 F [/подпрограммы]

[9:] Z 0 F [останов]

[10 power:] P 2 F [ степень многочлена ]

[11 addr:] P 13 @ [ адрес массива ]

[12:] P 0 D [ a4 = 1 ]

[array:]

[13:] P 1 D [ a3 = 3 ]

[14:] P 2 D [ a2 = 5 ]

[15:] P 3 D [ a1 = 7 ]

[16:] P 4 D [ a0 = 9 ]

EZ PF

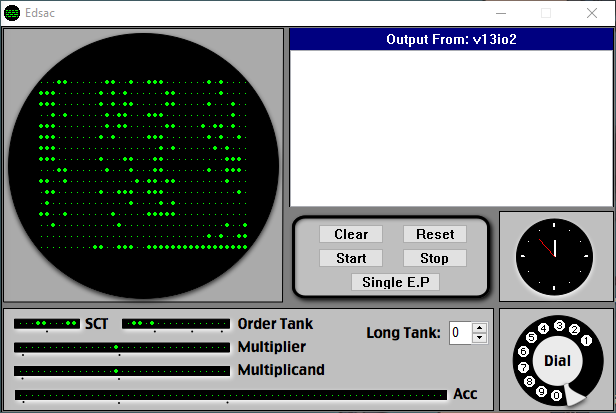


Рис. 2 результат работы программы IO2

Видим верный результат во 2 ячейке.

**Вывод:** в ходе работы были разработаны программы расчета значений многочлена по схеме Горнера с «длинным» результатом для загрузчиков IO1 и IO2 EDSAC.