Санкт-Петербургский политехнический университет Петра Великого

Кафедра компьютерных систем и программных технологий

**Отчёт по лабораторной работе**

Дисциплина: Низкоуровневое программирование

Тема: Программирование EDSAC

Вариант: 13

Выполнил студент гр. 3530901/90005 А. И. Терлецкий

(подпись)

Преподаватель Д. С. Степанов

(подпись)

“ ” 2021 г.

Санкт-Петербург

2021

**Постановка задачи**

1. Разработать программу для EDSAC, реализующую определенную вариантом 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 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**

Согласно варианту 13, необходимо рассчитать значение многочлена по схеме Горнера с «длинным» результатом (переполнение игнорируется).

**Схема Горнера**

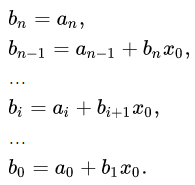
Задан многочлен:

****

Требуется вычислить значение данного многочлена при фиксированном значении x = x0{\displaystyle x=x\_{0}}. Представим многочлен {\displaystyle P(x)}P(x) в следующем виде:

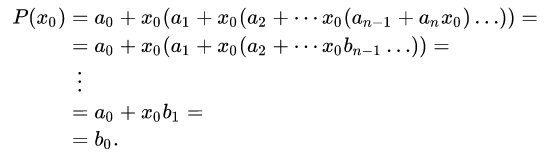
****

Определим следующую последовательность:

****

Искомое значение P(x0){\displaystyle P(x\_{0})} есть b0{\displaystyle b\_{0}}. Покажем, что это так.

В полученную форму записи *P(x)*{\displaystyle P(x)} подставим x = x0{\displaystyle x=x\_{0}} и будем вычислять значение выражения, начиная с внутренних скобок. Для этого будем заменять подвыражения через *{\displaystyle b\_{i}}bi*:

****

**Initial Orders 1**

Алгоритм был реализован следующий:

1. Преобразование старшего коэффициента an в длинный формат и запись в ячейку памяти 2-3;

2. Содержимое ячейки 2-3 умножается на x0, к результату прибавляется коэффициент an-1;

3. Шаг 2 повторяется число раз, равное степени многочлена (предполагается, что многочлен хотя бы первой степени).

**Текст программы**

[31] T 68 S

[32] A 61 S [загрузка в аккумулятор степени многочлена]

[33] T 0 S [запись в ячейку 0, очистка аккумулятора]

[34] A 63 S [загрузка в аккумулятор старшего коэффициента]

[35] T 2 S [запись в ячейку 2, очистка аккумулятора]

[36] H 60 S [запись 1 в умножающий регистр]

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

[38] R 1 S [сдвиг вправо на 2 бита]

[39] T 2 L [запись старшего коэффициента в длинную ячейку 2-3]

[loop]

[40] A 0 S [загрузка в аккумулятор счетчика]

[41] S 60 S [уменьшение счетчика на 1]

[42] G 59 S [если аккумулятор < 0, переход в ячейку 59]

[43] T 0 S [сохранение счетчика]

[44] H 62 S [запись x0 в умножающий регистр]

[45] V 2 L [умножение ячейки 2 на x0]

[46] L 1024 S [сдвиг влево на 12 бит]

[47] L 4 S [сдвиг влево на 4 бита]

[48] T 2 L [запись результата в ячейку 2]

[49] H 60 S [запись 1 в умножающий регистр]

[50] V 64 S [умножение a\_n на число в умножающем регистре]

[51] R 1 S [сдвиг вправо на 2 бита]

[52] A 2 L [прибавление числа из ячейки 2]

[53] T 2 L [запись результата в ячейку 2]

[54] A 60 S [загрузка в аккумулятор 1]

[55] L 0 L [сдвиг влево на 1]

[56] A 50 S [прибавление к аккумулятору инструкции, использующей коэффициенты]

[57] T 50 S [запись измененной инструкции]

[58] E 40 S [переход к началу цикла]

[59] Z 0 S

[60] P 0 L [1]

[61] P 2 S [степень многочлена = 4]

[62] P 1 L [x0 = 3]

[63] P 9 L [a4 = 19]

[64] P 2 S [a3 = 4]

[65] P 0 L [a2 = 1]

[66] P 5 L [a1 = 11]

[67] P 10 S [a0 = 20]

В ячейках 61-67 находятся входные данные: степень многочлена, значение x0, и массив коэффициентов. В ячейках 36-39 происходит преобразование числа в длинный формат. Для этого единица загружается в умножающий регистр, умножается на старший коэффициент и результат записывается в ячейку 2-3. В этой же ячейке будет записан результат. В ячейках 49-50 остальные коэффициенты так же преобразовываются в длинный формат.

**Пример выполнения программы на симуляторе:**

|  |
| --- |
|  |
| Рис. 1 Результат работы программы |

Входные данные: a4 = 19, a3 = 4, a2 = 1, a1 = 11, a0 = 20, x0 = 3

Решение по схеме Горнера:

P(x0) = (((19 \* 3 + 4) \* 3 + 1) \* 3 + 11) \* 3 + 20 = 1709

Как видно по рисунку, результаты совпадают.

**Initial Orders 2**

Функциональность программы, предполагающей загрузчик Initial Orders 1, была выделена в подпрограмму, в которой идут основные вычисления. Запись констант в ячейки и сами константы находятся в основной программе. Алгоритм не претерпел изменений.

**Текст программы**

T 56 K

G K [фиксация начального адреса]

[0] A 3 F [формирование кода инструкции возврата]

[1] T 24 @ [запись инструкции возврата]

[2] A 1 F [загрузка в аккумулятор адреса 1-го элемента массива]

[3] A 15 @ [прибавление к аккумулятору инструкции, использующей коэффициенты]

[4] T 15 @ [запись измененной инструкции]

[loop]

[5] A 0 F [загрузка в аккумулятор счетчика]

[6] S 25 @ [уменьшение счетчика на 1]

[7] G 24 @ [если аккумулятор < 0, переход в ячейку 24]

[8] T 0 F [сохранение счетчика]

[9] H 4 F [запись x0 в умножающий регистр]

[10] V 2 D [умножение ячейки 2 на x0]

[11] L 1024 F [сдвиг влево на 12 бит]

[12] L 4 F [сдвиг влево на 4 бита]

[13] T 2 D [запись результата в ячейку 2]

[14] H 25 @ [запись 1 в умножающий регистр]

[15] V 0 F [умножение a\_n на число в умножающем регистре]

[16] R 1 F [сдвиг вправо на 2 бита]

[17] A 2 D [прибавление числа из ячейки 2]

[18] T 2 D [запись результата в ячейку 2]

[19] A 25 @ [загрузка в аккумулятор 1]

[20] L 0 D [сдвиг влево на 1]

[21] A 15 @ [прибавление к аккумулятору инструкции, использующей коэффициенты]

[22] T 15 @ [запись измененной инструкции]

[23] E 5 @ [переход к началу цикла]

[24] E 0 F [инструкция возврата из подпрограммы]

[25] P 0 D [1]

G K [фиксация начального адреса]

[0] A 16 @ [загрузка в аккумулятор степени многочлена]

[1] T 0 F [запись в ячейку 0, очистка аккумулятора]

[2] A 17 @ [загрузка в аккумулятор адреса 2-го элемента]

[3] T 1 F [запись в ячейку 1, очистка аккумулятора]

[4] A 19 @ [загрузка в аккумулятор старшего коэффициента]

[5] T 2 F [запись в ячейку 2, очистка аккумулятора]

[6] H 15 @ [запись 1 в умножающий регистр]

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

[8] R 1 F [сдвиг вправо на 2 бита]

[9] T 2 D [запись старшего коэффициента в длинную ячейку 2-3]

[10] A 18 @ [загрузка в аккумулятор x0]

[11] T 4 F [запись в ячейку 4, очистка аккумулятора]

[12] A 12 @ [вызов]

[13] G 56 F [подпрограммы]

[14] Z 0 F

[15] P 0 D [1]

[16] P 2 F [степень многочлена = 4]

[17] P 20 @ [адрес 2-го элемента массива]

[18] P 1 D [x0 = 3]

[19] P 9 D [a4 = 19]

[20] P 2 F [a3 = 4]

[21] [P 17179869181 D] [a2 = -5]

[22] P 5 D [a1 = 11]

[23] P 10 F [a0 = 20]

EZ PF [переход к исполнению]

**Пример выполнения программы на симуляторе:**

|  |
| --- |
|  |
| Рис. 2 Результат работы программы |

Входные данные: a4 = 19, a3 = 4, a2 = -5, a1 = 11, a0 = 20, x0 = 3

Решение по схеме Горнера:

P(x0) = (((19 \* 3 + 4) \* 3 - 5) \* 3 + 11) \* 3 + 20 = 1655.

Как видно по рисунку, результаты совпадают.

|  |
| --- |
|  |
| Рис. 3 Результат работы программы |

Входные данные: a4 = -5, a3 = 4, a2 = 19, a1 = 11, a0 = 20, x0 = 3

Решение по схеме Горнера:

P(x0) = (((-5 \* 3 + 4) \* 3 + 19) \* 3 + 11) \* 3 + 20 = -73.

Получили число в доп. коде: 1.1111111111111111111111111110110111

Перевод в прямой код: 0.000000000000000000000000001001001

В десятичной системе: 73

Добавляем к числу минус в начале и получаем верный результат.

**Вывод**

В ходе выполнения лабораторной работы была реализована программа для EDSAC, реализующая расчет значения многочлена по схеме Горнера с «длинным» результатом (переполнение игнорируется) для двух

загрузчиков – Initial Orders 1 и Initial Orders 2; получены навыки по созданию алгоритмов для машины EDSAC, а также навыки работы с ее симулятором.