Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа №4

Дисциплина: Основы профессиональной деятельности

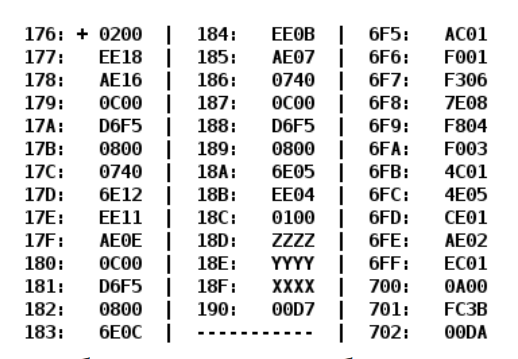
Вариант №14201

Выполнил: Васильев Никита Алексеевич, студент группы P3108

Преподаватель: Вербовой Александр Александрович

Санкт-Петербург 2024

Текст задания

По выданному преподавателем варианту восстановить текст заданного варианта программы и подпрограммы (программного комплекса), определить предназначение и составить его описание, определить область представления и область допустимых значений исходных данных и результата, выполнить трассировку программного комплекса.

Текст исходной программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Описание** | **Комментарии** |
| 176 | 0200 | CLA | Очистка аккумулятора: 0 → AC | Обнуление результата |
| 177 | EE18 | ST (IP + 24) | Прямая относительная запись: AC → 190 |
| 178 | AE16 | LD (IP + 22) | Прямая относительная загрузка: 18F → AC | Выбор переменной X |
| 179 | 0C00 | PUSH | Положить на стек: AC → −(SP) | Вызов функции  F(X)  Загрузка результата в аккумулятор |
| 17A | D6F5 | CALL 6F5 | Вызов подпрограммы: SP – 1 → SP, IP → SP, 6F5 → IP |
| 17B | 0800 | POP | Снятие со стека: (SP)+ → AC |
| 17C | 0740 | DEC | Декремент: AC – 1 → AC | Вычитание F(X) – 1  из результата  R = – (F(X) – 1) |
| 17D | 6E12 | SUB (IP + 18) | Прямое относительное вычитание: AC – 190 → AC |
| 17E | EE11 | ST (IP + 17) | Прямая относительная запись: AC → 190 |
| 17F | AE0E | LD (IP + 14) | Прямая относительная загрузка: 18E → AC | Выбор переменной Y |
| 180 | 0C00 | PUSH | Положить на стек: AC → −(SP) | Вызов функции  F(Y)  Загрузка результата в аккумулятор |
| 181 | D6F5 | CALL 6F5 | Вызов подпрограммы: SP – 1 → SP, IP → SP, 6F5 → IP |
| 182 | 0800 | POP | Снятие со стека: (SP)+ → AC |
| 183 | 6E0C | SUB (IP + 12) | Прямое относительное вычитание: AC – 190 → AC | Вычитание F(Y)  из результата  R = – F(Y) – (F(X) – 1) |
| 184 | EE0B | ST (IP + 11) | Прямая относительная запись: AC → 190 |
| 185 | AE07 | LD (IP + 7) | Прямая относительная загрузка: AC → 18D | Выбор переменной Z – 1 |
| 186 | 0740 | DEC | Декремент: AC – 1 → AC |
| 187 | 0C00 | PUSH | Положить на стек: AC → −(SP) | Вызов функции  F(Z – 1)  Загрузка результата в аккумулятор |
| 188 | D6F5 | CALL 6F5 | Вызов подпрограммы: SP – 1 → SP, IP → SP, 6F5 → IP |
| 189 | 0800 | POP | Снятие со стека: (SP)+ → AC |
| 18A | 6E05 | SUB (IP + 5) | Прямое относительное вычитание: AC – 190 → AC | Вычитание F(Z – 1)  из результата  R =  F(Z – 1) – (F(Y) – (F(X) – 1)) |
| 18B | EE04 | ST (IP + 4) | Прямая относительная запись: AC → 190 |
| 18C | 0100 | HLT | Останов |  |
| 18D | ZZZZ | --- | Переменная Z | Переменные |
| 18E | YYYY | --- | Переменная Y |
| 18F | XXXX | --- | Переменная X |
| 190 | 00D7 | --- | Результат R | Результат |

Текст подпрограммы

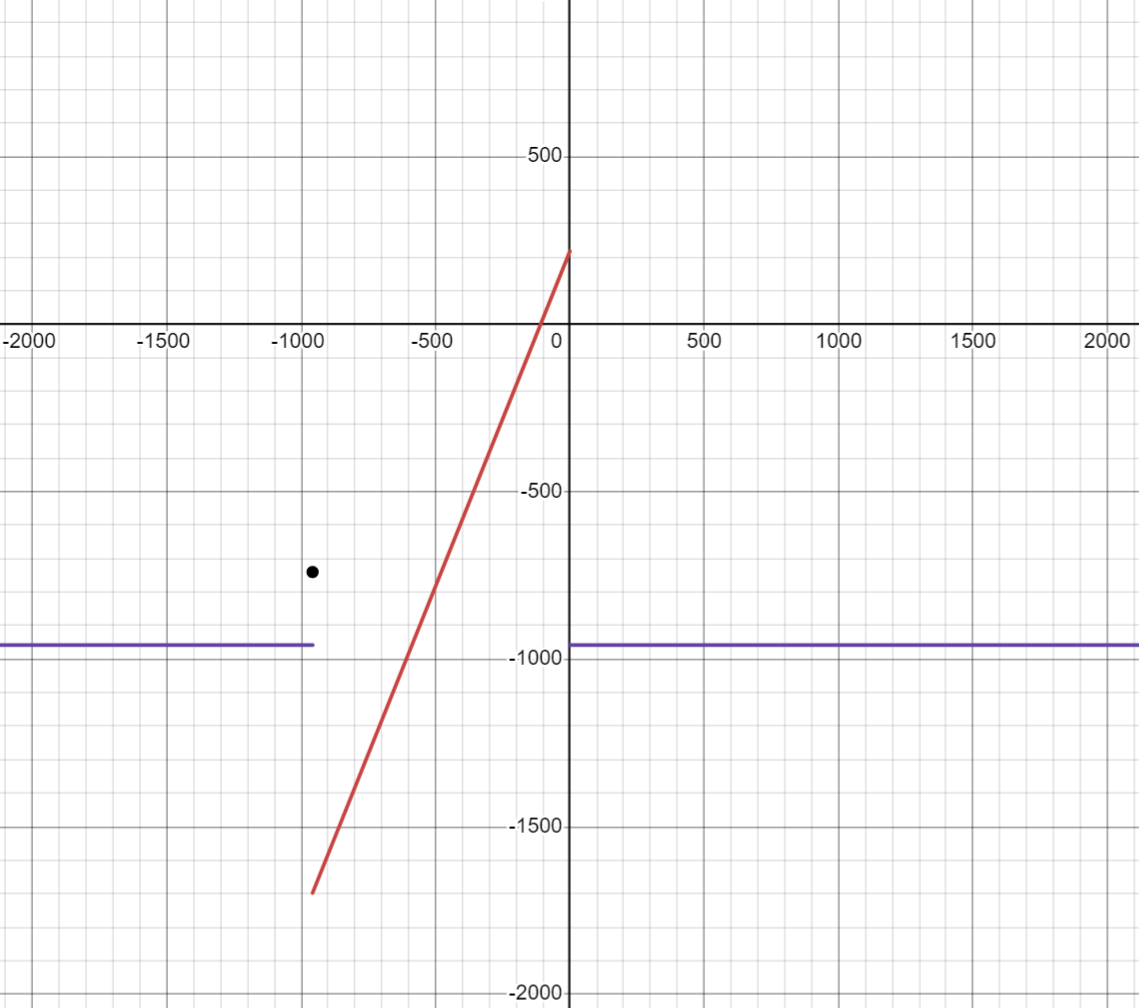
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Адрес** | **Код команды** | **Мнемоника** | **Описание** | **Комментарии** |
| 6F5 | AC01 | LD (SP + 1) | Косвенная относительная загрузка со смещением (SP) | Загрузка аргумента |
| 6F6 | F001 | BEQ 1 | Если равно (Z == 1), переход в 6F8 | Если аргумент равен 0, переход в 6F8 |
| 6F7 | F306 | BPL 6 | Если плюс (N == 0), переход в 6FE | Если аргумент положительный, переход в 6FE |
| 6F8 | 7E08 | CMP (IP + 8) | Установить флаги по результату: AC – 701 | Если аргумент меньше A, переход в 6FE, если равен A, переход в 6FC |
| 6F9 | F804 | BLT 4 | Если меньше (N ⊕ V == 1), переход в 6FE |
| 6FA | F003 | BEQ 1 | Если равно (Z == 1), переход в 6FC |
| 6FB | 4C01 | ADD (SP + 1) | Косвенное относительное сложение со смещением (SP): AC + (SP + 1) → AC | Прибавление аргумента |
| 6FC | 4E05 | ADD (IP + 5) | Прямое относительное сложение: AC + 702 → AC | Сложение с B |
| 6FD | CE01 | JUMP (IP + 1) | Прямой относительный прыжок: IP + 1 → IP | Переход в 6FF |
| 6FE | AE02 | LD (IP + 2) | Прямая относительная загрузка: 701 → AC | Загрузка A, если аргумент меньше A или положительный |
| 6FF | EC01 | ST (SP + 1) | Косвенная относительная запись со смещением (SP) | Сохранение результата |
| 700 | 0A00 | RET | Выход из подпрограммы | Возврат к основной программе |
| 701 | FC3B | --- | Константа A = −965 | Константы |
| 702 | 00DA | --- | Константа B = 218 |

# Описание программы

Формула:

Программа находит значение функции:

График:



Область представления и область допустимых значений

Область представления:

X, Y, Z, R, A, B – целые знаковые 16-ти разрядные числа.

Область допустимых значений:

Проанализируем функцию.

### Для заданных констант:

При значении аргумента функции равного *−965*, функция вернет значение *x + 218 = −965 + 218 = −747*. Переполнение не произойдёт

При значении аргумента функции в диапазоне (*−965*; *0*], функция вернет значение *2x + 218*. Это монотонная возрастающая функция, поэтому рассмотрим минимальное и максимальное значение:

Это означает, что на всем промежутке значений аргумента, результат функции будет находиться на отрезке [*−1711*; *215*].

Так как основная программа вычисляет следующее выражение:

То максимально можно получить: *216 + 1710 + 217 = 2143 < 215.*

Минимально можно получить: *−1710 − 965 − 1711 = −4386 > −215 − 1.* В обоих случаях переполнение не произойдет.

При оставшихся значениях аргумента функция вернёт значение *−965*, переполнение не произойдёт.

Тогда для заданных констант *A* и *B* ОДЗ следующее:

### В общем виде:

При значении аргумента функции равного *A*, функция вернет значение *x + B = A + B*.

Чтобы не произошло переполнение, необходимо, чтобы *−215 ≤* *A + B ≤ 215 − 1*

При значении аргумента функции в диапазоне (*A*; 0], функция вернет значение *2x + B*. Это монотонная возрастающая функция, поэтому рассмотрим минимальное и максимальное значение:

Это означает, что на всем промежутке значений аргумента, результат функции будет находиться на отрезке [*2A+B*; *B*].

Так как основная программа вычисляет следующее выражение:

То максимально можно получить: *2(A + 1) + B − 1* *+ 2(A + 1) + B + A + B = 5A + 3B + 3.*

Минимально можно получить: *B* *− 1 − B + 2 = 1.*

Чтобы не произошло переполнение, необходимо, чтобы *−215 ≤* *5A + 3B + 4 ≤ 215 − 1.*

При оставшихся значениях аргумента функция вернёт значение *A*, переполнение не произойдёт.

Расположение в памяти ЭВМ программы, исходных данных и результатов

176 – 18C – расположение команд основной программы, 6F5 – 700 – расположение команд подпрограммы;

18D – 18F – расположение переменных основной программы, 701 – 702 – расположение переменных подпрограммы;

190 – расположение результата.

Адреса первой и последней выполняемых команд

Основная программа:

176 – адрес первой выполняемой команды;

18C – адрес последней выполняемой команды.

Подпрограмма:

6F5 – адрес первой выполняемой команды;

700 – адрес последней выполняемой команды.

Таблица трассировки:

x=FEC7, y=A006, z=0000

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Выполняемая команда** | | **Содержимое регистров процессора** | | | | | | | | **Ячейка, содержимое которой изменилось после выполнения команды** | | |
| Адрес | Код | IP | CR | AR | DR | SP | BR | AC | NZVC | | Адрес | Новый код | |
| 176 | 0200 | 177 | 0200 | 176 | 0200 | 000 | 0176 | 0000 | 0100 | |  |  | |
| 177 | EE18 | 178 | EE18 | 190 | 0000 | 000 | 0018 | 0000 | 0100 | | 190 | 0000 | |
| 178 | AE16 | 179 | AE16 | 18F | FEC7 | 000 | 0016 | FEC7 | 1000 | |  |  | |
| 179 | 0C00 | 17A | 0C00 | 7FF | FEC7 | 7FF | 0179 | FEC7 | 1000 | | 7FF | FEC7 | |
| 17A | D6F5 | 6F5 | D6F5 | 7FE | 017B | 7FE | D6F5 | FEC7 | 1000 | | 7FE | 017B | |
| 6F5 | AC01 | 6F6 | AC01 | 7FF | FEC7 | 7FE | 0001 | FEC7 | 1000 | |  |  | |
| 6F6 | F001 | 6F7 | F001 | 6F6 | F001 | 7FE | 06F6 | FEC7 | 1000 | |  |  | |
| 6F7 | F306 | 6F8 | F306 | 6F7 | F306 | 7FE | 06F7 | FEC7 | 1000 | |  |  | |
| 6F8 | 7E08 | 6F9 | 7E08 | 701 | FC3B | 7FE | 0008 | FEC7 | 0001 | |  |  | |
| 6F9 | F804 | 6FA | F804 | 6F9 | F804 | 7FE | 06F9 | FEC7 | 0001 | |  |  | |
| 6FA | F003 | 6FB | F003 | 6FA | F003 | 7FE | 06FA | FEC7 | 0001 | |  |  | |
| 6FB | 4C01 | 6FC | 4C01 | 7FF | FEC7 | 7FE | 0001 | FD8E | 1001 | |  |  | |
| 6FC | 4E05 | 6FD | 4E05 | 702 | 00DA | 7FE | 0005 | FE68 | 1000 | |  |  | |
| 6FD | CE01 | 6FF | CE01 | 6FD | 06FF | 7FE | 0001 | FE68 | 1000 | |  |  | |
| 6FF | EC01 | 700 | EC01 | 7FF | FE68 | 7FE | 0001 | FE68 | 1000 | | 7FF | FE68 | |
| 700 | 0A00 | 17B | 0A00 | 7FE | 017B | 7FF | 0700 | FE68 | 1000 | |  |  | |
| 17B | 0800 | 17C | 0800 | 7FF | FE68 | 000 | 017B | FE68 | 1000 | |  |  | |
| 17C | 0740 | 17D | 0740 | 17C | 0740 | 000 | 017C | FE67 | 1001 | |  |  | |
| 17D | 6E12 | 17E | 6E12 | 190 | 0000 | 000 | 0012 | FE67 | 1001 | |  |  | |
| 17E | EE11 | 17F | EE11 | 190 | FE67 | 000 | 0011 | FE67 | 1001 | | 190 | FE67 | |
| 17F | AE0E | 180 | AE0E | 18E | A006 | 000 | 000E | A006 | 1001 | |  |  | |
| 180 | 0C00 | 181 | 0C00 | 7FF | A006 | 7FF | 0180 | A006 | 1001 | | 7FF | A006 | |
| 181 | D6F5 | 6F5 | D6F5 | 7FE | 0182 | 7FE | D6F5 | A006 | 1001 | | 7FE | 0182 | |
| 6F5 | AC01 | 6F6 | AC01 | 7FF | A006 | 7FE | 0001 | A006 | 1001 | |  |  | |
| 6F6 | F001 | 6F7 | F001 | 6F6 | F001 | 7FE | 06F6 | A006 | 1001 | |  |  | |
| 6F7 | F306 | 6F8 | F306 | 6F7 | F306 | 7FE | 06F7 | A006 | 1001 | |  |  | |
| 6F8 | 7E08 | 6F9 | 7E08 | 701 | FC3B | 7FE | 0008 | A006 | 1000 | |  |  | |
| 6F9 | F804 | 6FE | F804 | 6F9 | F804 | 7FE | 0004 | A006 | 1000 | |  |  | |
| 6FE | AE02 | 6FF | AE02 | 701 | FC3B | 7FE | 0002 | FC3B | 1000 | |  |  | |
| 6FF | EC01 | 700 | EC01 | 7FF | FC3B | 7FE | 0001 | FC3B | 1000 | | 7FF | FC3B | |
| 700 | 0A00 | 182 | 0A00 | 7FE | 0182 | 7FF | 0700 | FC3B | 1000 | |  |  | |
| 182 | 0800 | 183 | 0800 | 7FF | FC3B | 000 | 0182 | FC3B | 1000 | |  |  | |
| 183 | 6E0C | 184 | 6E0C | 190 | FE67 | 000 | 000C | FDD4 | 1000 | |  |  | |
| 184 | EE0B | 185 | EE0B | 190 | FDD4 | 000 | 000B | FDD4 | 1000 | | 190 | FDD4 | |
| 185 | AE07 | 186 | AE07 | 18D | 0000 | 000 | 0007 | 0000 | 0100 | |  |  | |
| 186 | 0740 | 187 | 0740 | 186 | 0740 | 000 | 0186 | FFFF | 1000 | |  |  | |
| 187 | 0C00 | 188 | 0C00 | 7FF | FFFF | 7FF | 0187 | FFFF | 1000 | | 7FF | FFFF | |
| 188 | D6F5 | 6F5 | D6F5 | 7FE | 0189 | 7FE | D6F5 | FFFF | 1000 | | 7FE | 0189 | |
| 6F5 | AC01 | 6F6 | AC01 | 7FF | FFFF | 7FE | 0001 | FFFF | 1000 | |  |  | |
| 6F6 | F001 | 6F7 | F001 | 6F6 | F001 | 7FE | 06F6 | FFFF | 1000 | |  |  | |
| 6F7 | F306 | 6F8 | F306 | 6F7 | F306 | 7FE | 06F7 | FFFF | 1000 | |  |  | |
| 6F8 | 7E08 | 6F9 | 7E08 | 701 | FC3B | 7FE | 0008 | FFFF | 0001 | |  |  | |
| 6F9 | F804 | 6FA | F804 | 6F9 | F804 | 7FE | 06F9 | FFFF | 0001 | |  |  | |
| 6FA | F003 | 6FB | F003 | 6FA | F003 | 7FE | 06FA | FFFF | 0001 | |  |  | |
| 6FB | 4C01 | 6FC | 4C01 | 7FF | FFFF | 7FE | 0001 | FFFE | 1001 | |  |  | |
| 6FC | 4E05 | 6FD | 4E05 | 702 | 00DA | 7FE | 0005 | 00D8 | 0001 | |  |  | |
| 6FD | CE01 | 6FF | CE01 | 6FD | 06FF | 7FE | 0001 | 00D8 | 0001 | |  |  | |
| 6FF | EC01 | 700 | EC01 | 7FF | 00D8 | 7FE | 0001 | 00D8 | 0001 | | 7FF | 00D8 | |
| 700 | 0A00 | 189 | 0A00 | 7FE | 0189 | 7FF | 0700 | 00D8 | 0001 | |  |  | |
| 189 | 0800 | 18A | 0800 | 7FF | 00D8 | 000 | 0189 | 00D8 | 0001 | |  |  | |
| 18A | 6E05 | 18B | 6E05 | 190 | FDD4 | 000 | 0005 | 0304 | 0000 | |  |  | |
| 18B | EE04 | 18C | EE04 | 190 | 0304 | 000 | 0004 | 0304 | 0000 | | 190 | 0304 | |
| 18C | 0100 | 18D | 0100 | 18C | 0100 | 000 | 018C | 0304 | 0000 | |  |  | |

При значениях x=FEC7, y=A006, z=0000 получаем результат R = (0304)16 = (772)10.

Вывод:

При выполнении данной лабораторной работы я познакомился с реализацией стека в БЭВМ, научился использовать подпрограммы и вызывать их, изучил команды POP, CALL, PUSH и RET.