## Лабораторная работа № 3

**Тема.** **Программирование разветвляющегося процесса**;

**Цель:** изучение команд условных переходов, способов организации разветвляющихся программ; исследование порядка функционирования ЭВМ при выполнении таких команд;

**Техническое обеспечение:** персональный компьютер, программная модель учебной ЭВМ;

**Основные понятия:** команды условной передачи управления.

Команды условной передачи управления

Команды, расположенные друг за другом в памяти, машина считывает последовательно. При реализации алгоритма, требующего изменение естественной последовательности выполнения команд в зависимости от результата вычислений, используют ***команды* *условной передачи управления***, т.е. «переход при условии…» по определенному адресу. Фактически, каждое условие является каким-либо состоянием флагов.

Команды условной передачи управления: **JZ, JNZ, JS, JNS, JO, JNO** (см. файл: Описание архитектуры).

**JS** – **If Асc < 0 then PC ← CR[ADR]**. Если результат операции меньше нуля (Асс < 0), т.е. в знаковом разряде единица, тогда адресная часть команды ADR заносится в регистр PC, тем самым осуществляется переход**;**

**JNS** – **If Асc > 0 then PC ← CR[ADR]**. Переход, если результат операции больше нуля Асс > 0, т.е. в знаковом разряде ноль**;**

**JZ – If Асc = 0 then PC ← CR[ADR]**. Переход, если результат операции равен нулю, Асс = 0;

**JNZ - If Асc ≠ 0 then PC ← CR[ADR]**. Если результат операции не равен нулю, Асс ≠ 0;

**JO – If |Acc| ≥ 99999 then PC ← CR[ADR]**. В данном случае, управление передается при условии переполнения разрядной сетки;

**JNO - If |Acc| ≤ 99999 then PC ← CR[ADR]**. Переход, если нет переполнения разрядной сетки.

Допустим, необходимо возвести в степень (2), положительное число ***х***. Программа должна проверить знак числа ***х***, а затем совершить переход в зависимости от полученного результата. Если число отрицательное, программа завершает работу. Если число положительное, программа должна возвести в степень число, выдать результат арифметической операции и завершить работу.

Этот алгоритм можно реализовать с помощью команды **JS,** которая осуществляет переход, если результат отрицательный. Рассмотрим код программы:

000 in // ввод числа

**001 JS 005** // переход, если число отрицательное

002 WR 10 // запись числа в ячейку памяти 010

003 mul 10 // возведение числа в степень

004 out // вывод результата

**005 hlt** // останов

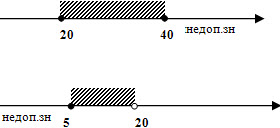
Порядок выполнения работы

(1)

*Варианты функций и допустимые пределы изменения аргумента приведены в таблице 4, варианты заданий — в таблице 5.*

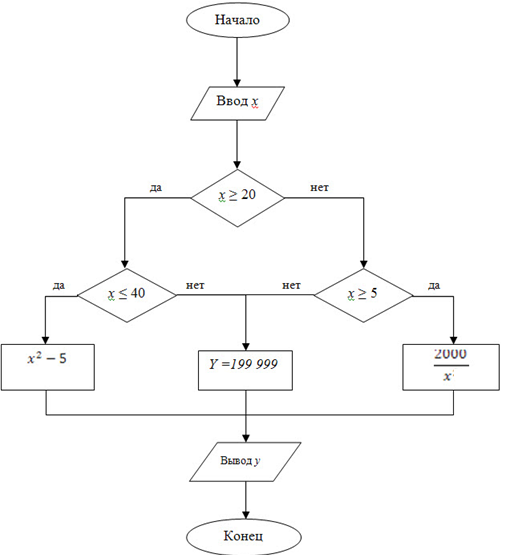
Допустим, дана следующая функция

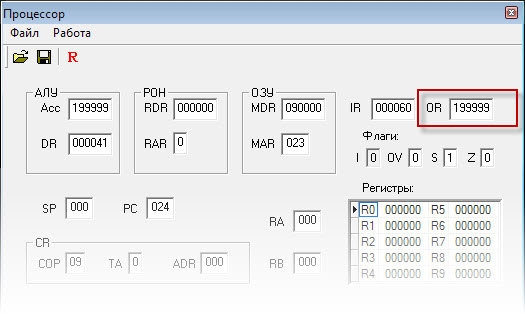
**1)** Исходя из допустимых пределов изменения аргумента функций (табл. 4) и значения параметра ***а*** для своего варианта задания (табл. 5) выделить на числовой оси ***Ох*** области, в которых функция у вычисляется по представленной в п.1 формуле, и недопустимые значения аргумента, рисунок 1.

Рисунок 1 – Область допустимых значений

**2)** Разработать блок-схему алгоритма решения задачи, рисунок 2.

Рисунок 2 – Блок-схема алгоритма программы



**3)** Разработать программу вычисления и вывода значения функции. Причем ***x***вводится с устройства ввода IR, результат ***у*** выводится на OR. На недопустимых значениях аргумента программа должна выдавать на OR максимальное отрицательное число: 199 999.

В данной лабораторной работе используются двухсловные команды, с непосредственной адресацией, позволяющие оперировать отрицательными числами и числами по модулю, превышающие 999, в качестве непосредственного операнда.

Оценив размер программы примерно в 20 команд, отведем для области данных ячейки ОЗУ, начиная с адреса 030. Составленная программа с комментариями представлена в виде таблицы 1 .

Таблица 1 – Пример программы вычисления функции

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Команда** | | **Комментарии** |
| **Мнемокод** | **Код** |
| 000 | IN | 010000 | Ввод ***х*** |
| 001 | WR 030 | 220030 | Запись ***х*** в ячейку М (30) |
| 002 | SUB #20 | 241020 | Сравнение с нижней границей (*х* - 20) |
| 003 | JS 012 | 130012 | Переход, если разность отрицательна |
| 004 | RD 030 | 210030 | Чтение ОЗУ |
| 005 | SUB #041 | 241041 | Сравнение с верхней границей (*х* - 41) |
| 006 | JNS 20 | 140020 | Переход, если разность положительна |
| 007 | RD 030 | 210030 | *Вычисление по первой формуле* |
| 008 | MUL 030 | 250030 | Умножение на содержимое ячейки (возведение *х* в степень) |
| 009 | SUB #5 | 241005 | Вычитаем 5 |
| 010 | WR 031 | 220031 | Запись в ячейку М (31) |
| 011 | JMP 22 | 100022 | Переход на вывод результата |
| 012 | RD 030 | 210030 | Чтение ОЗУ |
| 013 | SUB #5 | 241005 | Сравнение с нижней границей (*х* - 5) |
| 014 | JS 20 | 130020 | Переход к ячейке М (20) |
| 015 | RDI 002000 | 410000 002000 | *Вычисление по второй формуле* |
| 017 | DIV 030 | 260030 | Деление на содержимое ячейки 030 |
| 018 | WR 032 | 220032 | Запись результата в ячейку М (32) |
| 019 | JMP 22 | 100022 | Переход на вывод результата |
| 020 | RDI 199999 | 410000 199999 | Чтение ОЗУ (двухсловной команды, ячейки 015-016 ) |
| 022 | OUT | 020000 | Вывод результата |
| 023 | HLT | 090000 | Стоп |

**4)** Ассемблировать текст программы, при необходимости исправить синтаксические ошибки.

**5)** Отладить программу. Для этого:

а) записать в IR значение аргумента ***х > а*** (в области допустимых значений);

б) записать в PC стартовый адрес программы;

в) проверить правильность результата и адреса остановки в автоматическом режиме. В случае наличия ошибки выполнить пп.5.г и 5.д; иначе перейти к п. 5.е;

г) записать в PC стартовый адрес программы;

д) наблюдая выполнение программы в режиме **Шаг**, найти команду, являющуюся причиной ошибки; исправить ее; выполнить пп. 5.a — 5.в;

е) записать в IR значение аргумента ***х < а*** (в области допустимых значений); выполнить пп. 5.б и 5.в;

ж) записать в IR недопустимое значение аргумента ***х*** и выполнить пп. 5.б и 5.в.

**6)** Для выбранного допустимого значения аргумента ***х*** наблюдать выполнение отлаженной программы в режиме **Шаг** и записать в форме табл. 2 содержимое регистров ЭВМ перед выполнением каждой команды.

**7)** Записать изменения состояний регистров и содержимого ячеек памяти в **режиме Шаг**, для одного значения аргумента *х* = 20, таблица 3.

Таблица 2 – Состояния регистров

|  |  |  |  |
| --- | --- | --- | --- |
| **РС** | **Асс** | **М(30)** | **М(31)** |
| 000 | 000000 | 000000 | 000000 |
| 001 | **000020** | 000000 | 000000 |
| 002 | 000020 | **000020** | 000000 |
| 003 | 000000 | 000020 | 000000 |
| 004 | 000000 | 000020 | 000000 |
| 005 | 000020 | 000020 | 000000 |
| 006 | **100021** | 000020 | 000000 |
| 007 | 100021 | 000020 | 000000 |
| 008 | **000020** | 000020 | 000000 |
| 009 | **000400** | 000020 | 000000 |
| 010 | **000395** | 000020 | 000000 |
| 011 | 000395 | 000020 | **000395** |
| … | … | … | … |
| 022 | 000395 | 000020 | 000395 |
| 023 | 000395 | 000020 | 000395 |

**8)** Записать результаты выполнения программы для нескольких значений аргумента, выбранных самостоятельно в форме таблицы 3.

Таблица 3 – Значения х, у

|  |  |
| --- | --- |
| **Входные данные (*х*)** | **Выходные данные (*Y*)** |
| 20 | 395 |
| 25 | 620 |
| 41 | 199999 |
| 4 | 199999 |
| 5 | 400 |
| 15 | 133 |

Новый рисунокДана функция:

По таблице 4 выбрать вариант задания. Например, первый вариант: значение

F***i*** = 2, что соответствует номеру 2 в таблице 5; значение *F****j*** = 1, что соответствует номеру 1 в таблице 5; ***a*** = 12. В данном случае, функция для первого варианта задании:

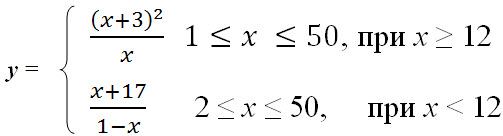


Таблица 4 – Варианты значения параметров i, j, a

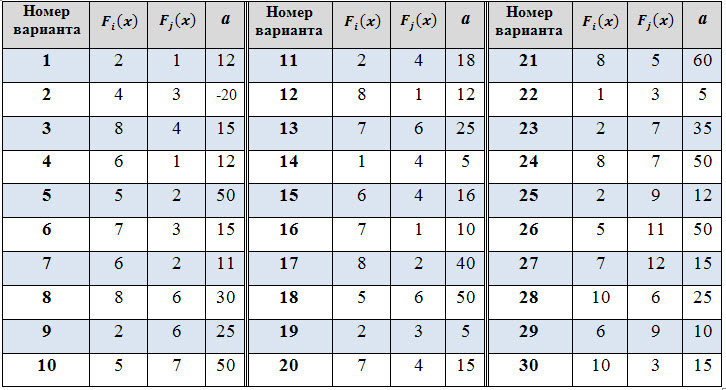
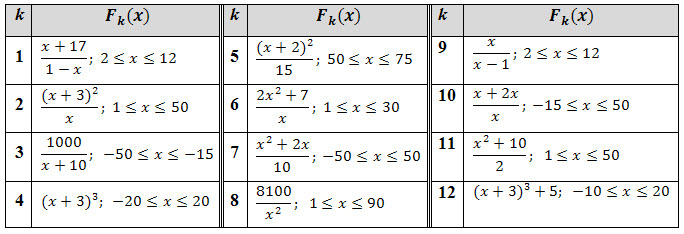


Таблица 5 – Варианты заданий функции



Форма и содержание отчета

1. Титульный лист;
2. Формулировка варианта задания;
3. Область допустимых значений;
4. Блок-схема алгоритма решения задачи;
5. Размещение данных в ОЗУ (какие ячейки памяти используются для данных, программы, констант);
6. Программа в форме таблицы 1;
7. Последовательность состояний регистров ЭВМ при выполнении программы в режиме **Шаг** для одного значения аргумента в форме таблицы 2.10;
8. Результаты выполнения программы для нескольких значений аргумента, выбранных самостоятельно в форме таблицы 3.

Контрольные вопросы

1. Какие команды условной передачи управления вы знаете?
2. При каком условии передается управление каждой из команд?
3. Если условие выполняется, то какой адрес точки перехода будет по команде JZ 25?