МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

федеральное государственное бюджетное образовательное учреждение высшего образования

**«Вологодский государственный университет»**

**Институт математики, естественных и компьютерных наук**

**Информатика и вычислительная техника**

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

Архитектура ЭВМ и система команд

Дисциплина: «ЭВМ и периферийные устройства»

Направление подготовки: 09.03.01. Информатика и вычислительная техника

|  |  |
| --- | --- |
| Руководитель | Павлов В.В. |
| Выполнили студенты | Старцев Д.В. |
| Группа, курс | ВМ-31 |
| Дата сдачи | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Дата защиты | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись преподавателя)* |

Вологда

2022 г.

Задание 1

1. Ознакомиться с архитектурой ЭВМ

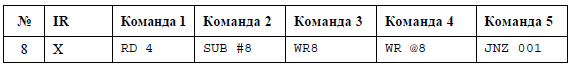
2. Записать в ОЗУ "программу", состоящую из пяти команд варианта 8. Команды разместить в последовательных ячейках памяти.

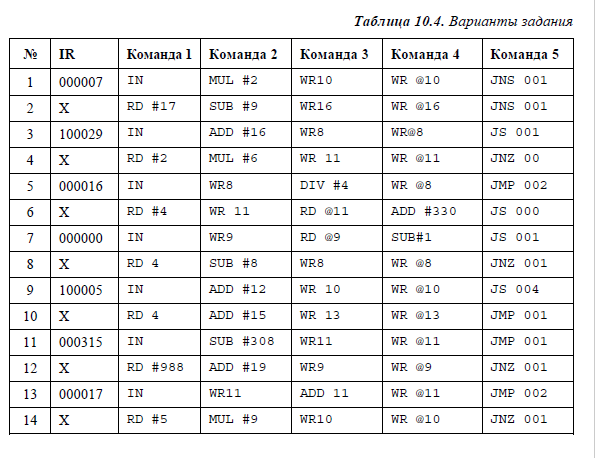
3. При необходимости установить начальное значение в устройство ввода IR.

4. Определить те программно-доступные объекты ЭВМ, которые будут изменяться при выполнении этих команд.

5. Выполнить в режиме Шаг введенную последовательность команд, фиксируя изменения значений объектов

6. Если в программе образуется цикл, необходимо просмотреть не более двух повторений каждой команды, входящей в тело цикла.

За IR x взято значение 000010



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Последовательность | Значения | | | | |
| команды | RD 004 | SUB #008 | WR 008 | WR @008 | JNZ 1 |
| коды | 210004 | 241008 | 220008 | 222008 | 120001 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PC | ACC | M 30 | M 20 | PC | ACC | M 30 | M 20 |
| 000 | 000000 | 000000 | 000000 | 001 | 120009 |  |  |
| 001 | 120001 |  |  | 002 | 120017 |  |  |
| 002 | 120009 |  |  | 003 | 120017 |  |  |
| 003 | 120009 |  |  | 004 | 120017 |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 004 | 120009 |  |  | 001 | 120017 |  |  |

8. Как записать программу в машинных кодах в память модели ЭВМ?

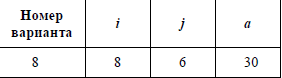
Для этого необходимо записать в текст программы коды, которые нам нужно использовать, нажать кнопку компиляции, либо через вспомогательное окно работа → компиляция. Введенные коды программа переведет в машинный код и запишет в память ЭВМ.

Можно вводить код программы сразу в ячейку памяти, он автоматически будет отображается в программе.

Задание 2.

Программирование разветвляющегося процесса

1. Разработать программу вычисления и вывода значения функции:





для вводимого из IR значения аргумента *x* . Функции и допустимые пре-

делы изменения аргумента приведены в табл. 10.6, варианты заданий —

в табл. 10.7.

2. Исходя из допустимых пределов изменения аргумента функций (табл. 10.6)

и значения параметра *a* для своего варианта задания (табл. 10.7) выделить

на числовой оси *Ox* области, в которых функция *y* вычисляется по пред-

ставленной в п. 1 формуле, и недопустимые значения аргумента. На недо-

пустимых значениях аргумента программа должна выдавать на OR мак-

симальное по модулю отрицательное число: 199 999.

3. Ввести текст программы в окно **Текст программы**, при этом возможен

набор и редактирование текста непосредственно в окне **Текст программы**

или загрузка текста из файла, подготовленного в другом редакторе.

4. Ассемблировать текст программы, при необходимости исправить синтак-

сические ошибки.

5. Отладить программу. Для этого:

а) записать в IR значение аргумента *x* *a* (в области допустимых значений);

б) записать в PC стартовый адрес программы;

в) проверить правильность выполнения программы (т. е. правильность ре-

зультата и адреса останова) в автоматическом режиме. В случае нали-

чия 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, яв-

ляющуюся причиной ошибки; исправить ее; выполнить пп. 5, *а* — 5, *в*;

е) записать в IR значение аргумента *x* *a* (в области допустимых значе-

ний); выполнить пп. 5, *б* и 5, *в*;

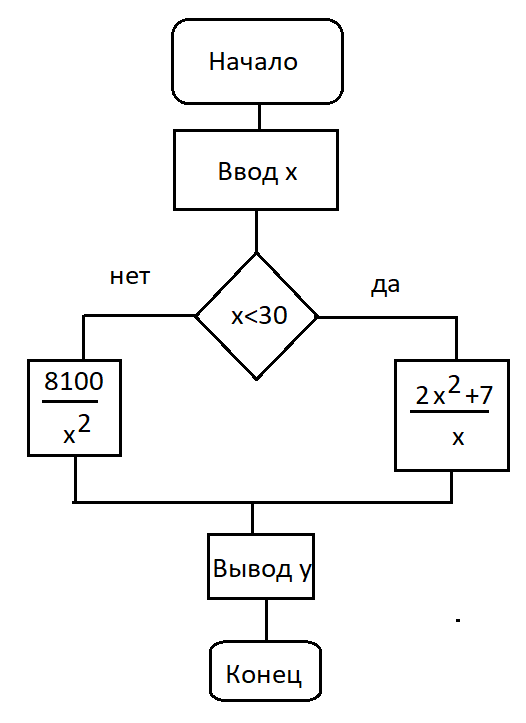
ж) записать в IR недопустимое значение аргумента *x* и выполнить пп. 5, *б*

и 5, *в*.

6. Для выбранного допустимого значения аргумента *x* наблюдать выполне-

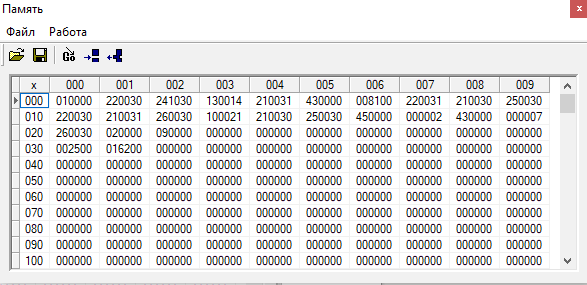
ние отлаженной программы в режиме **Шаг** и записать в форме табл. 10.3

содержимое регистров ЭВМ перед выполнением каждой команды.

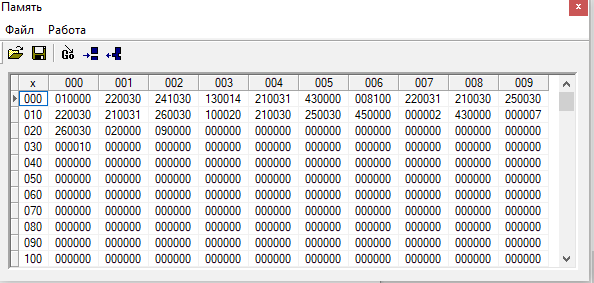


Размещение данных в ОЗУ

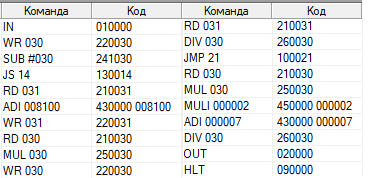
Если х>30



Если х<30



Команда и код, соответствующий ей.



Последовательность состояния регистров ЭВМ

Если х>30 Если х<30

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PC | ACC | IR | OR | PC | ACC | IR | OR |
| 000 | 000000 | 000050 | 000000 | 000 | 000000 | 000020 | 000000 |
| 001 | 000050 | 000050 | 000000 | 001 | 000020 | 000020 | 000000 |
| 002 | 000050 | 000050 | 000000 | 002 | 000020 | 000020 | 000000 |
| 003 | 000020 | 000050 | 000000 | 003 | 100010 | 000020 | 000000 |
| 004 | 000020 | 000050 | 000000 | 014 | 100010 | 000020 | 000000 |
| 005 | 000000 | 000050 | 000000 | 015 | 000020 | 000020 | 000000 |
| 007 | 008100 | 000050 | 000000 | 016 | 000400 | 000020 | 000000 |
| 008 | 008100 | 000050 | 000000 | 017 | 000800 | 000020 | 000000 |
| 009 | 000050 | 000050 | 000000 | 020 | 000807 | 000020 | 000000 |
| 010 | 002500 | 000050 | 000000 | 021 | 000040 | 000020 | 000000 |
| 011 | 002500 | 000050 | 000000 | 022 | 000040 | 000020 | 000040 |
| 012 | 008100 | 000050 | 000000 |  |  |  |  |
| 013 | 000003 | 000050 | 000000 |  |  |  |  |
| 021 | 000003 | 000050 | 000000 |  |  |  |  |
| 022 | 000003 | 000050 | 000003 |  |  |  |  |

Вводим число 60 в регистр IR для расчёта значения функции

Результат выполнения в регистре OR число 2.

Вводим число 25 в регистр IR для расчёта значения функции

Результат выполнения в регистре OR число 50.

Вводим число 55 в регистр IR для расчёта значения функции

Результат выполнения в регистре OR число 2.

Код программы

IN

WR 30

SUB #30

JS 014

RD 31

ADI 8100

WR 31

RD 30

MUL 30

WR 30

RD 31

DIV 30

JMP 21

RD 30

MUL 30

MULI 2

ADI 7

DIV 30

OUT

HLT

Ответ на контрольный вопрос

3. Как работают команды передачи управления?

Выполнение программы, записанной в памяти ЭВМ, осуществляется последовательно по командам в порядке возрастания адресов команд или в порядке, определяемом командами передачи управления.

Команды передачи управления: безусловный и шесть условных переходов, вызов подпрограммы, возврат из подпрограммы, цикл, программное прерывание, возврат из прерывания;

Команда передачи управления JMP переводит работу программы с текущего адреса в котором находится команда, на адрес указанный в команде.

Команда передачи управления JS переводит работу программы с текущего адреса в котором находится команда, на адрес указанный в команде, проверяя условие отрицательной разности входящего значения.