



**лабораторный
практикум
по электронным
цифровым
вычислительным
машинам**



КНИГА I

КАЗАНЬ

1973

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ
ПО ЭЛЕКТРОННЫМ ЦИФРОВЫМ
ВЫЧИСЛИТЕЛЬНЫМ
МАШИНАМ**

КНИГА I

КАЗАНЬ 1973

ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО ЭЛЕКТРОННЫМ ЦИФРОВЫМ ВЫЧИСЛИТЕЛЬНЫМ МАШИНАМ

В 3 КНИГАХ

Под общей редакцией
доцента Н. Н. Ливанова

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО
ОБРАЗОВАНИЯ РСФСР

КАЗАНСКИЙ ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ
АВИАЦИОННЫЙ ИНСТИТУТ им. А. Н. ТУПОЛЕВА

Кафедра электронных вычислительных машин

T. B. МАРЧЕНКО, B. И. ХАРИТОНОВ, R. C. ХАРИТОНОВА

**ЭЦВМ „НАИРИ“
УСТРОЙСТВО МАШИНЫ
И ПРОГРАММИРОВАНИЕ**

КАЗАНЬ
1973

ПРЕДИСЛОВИЕ

Студенты факультета вычислительных и управляющих систем Казанского авиационного института в числе основных профильирующих дисциплин изучают курсы "Электронные цифровые машины" (ЭЦМ) и "Алгоритмические языки и программирование". В первом рассматриваются теория и методы построения ЭЦМ, во втором - применение ЭЦМ для решения задач. Изучение материала курсов сопровождается практическими занятиями по теоретическим разделам и лабораторной практикой на реальных ЭЦВМ.

Лабораторный практикум по ЭЦМ обеспечивается вычислительной лабораторией, располагающей машинами: "Наири-С", "Сетунь", "Минск-II" и "М-220М". Эти машины относятся к различным видам ЭЦВМ первого и второго поколений, перекрываая типажом своих функциональных устройств почти все наиболее распространенные их типы.

На лабораторном практикуме по курсу ЭЦМ студенты изучают структуру машин и особенности организации и действия различных типов функциональных устройств ЭЦВМ "Сетунь", "Минск" и "Наири-С", не вникая в детали конструкции и электрических схем самих машин. Практикум включает в себя 17 лабораторных работ.

На машине "Наири", относящейся ко второму поколению машин, изучаются ее структура, пульт управления, внешние устройства, долговременное запоминающее устройство.

На машине "Минск-II" первого поколения изучаются структура машины, пульт управления, внешние устройства, устройство управления, арифметическое устройство, оперативное запоминающее устройство, накопитель на магнитной ленте.

На машине "Сетунь", работющей на магнитных элементах в троичной системе счисления, изучаются структура машины, пульт управления, система элементов, арифметическое устройство, устройство управления, оперативное запоминающее устройство, накопитель на магнитном барабане, внешние устройства.

Каждая лабораторная работа рассчитана на 4 часа и включает предварительную проверку подготовленности студента, выполнение и сдачу индивидуальных заданий. Перед каждым заданием студент должен изучить теоретическую часть и разобраться в схемах, приведенных в описании соответствующей лабораторной работы. Рекомендуется к каждому занятию проработать соответствующий раздел лекционного курса.

На лабораторных занятиях по курсу программирования студенты самостоятельно решают задачи по составленным ими программам на ЭЦВМ "Наирис-С" и "М-220М".

Настоящее учебное пособие предназначено для студентов, выполняющих лабораторные работы, а также для лиц, желавших в общих чертах изучить устройство данных машин и научиться самостоятельно решать задачи на этих машинах.

Пособие состоит из трех книг.

Данная, первая книга дает описание устройства ЭЦВМ "Наирис-С", изложенное как описание лабораторных работ, и содержит материал, необходимый для изучения программирования и решения задач на машине "Наирис".

Вторая и третья книги содержат аналогичное изложение устройства машин "Минск-II" и "Сетунь". Программирование на машинах "Минск-II" и "Сетунь" не приводится, поскольку к настоящему времени эти машины начали терять свое практическое значение.

В дальнейшем предполагается издание четвертой книги, посвященной изучению машины "М-220М".

При написании настоящего пособия использовалась техническая документация на машину "Наирис-С" и ранее составленные описания лабораторных работ.

Часть первая УСТРОЙСТВО ЭЦВМ "Наирис-С"

ОСНОВНЫЕ ТЕХНИЧЕСКИЕ ДАННЫЕ ЭДВМ "НАИРИ-С"

1. "Наири-С" является двухадресной программно-управляемой машиной с естественным порядком выполнения команд.

2. Форма представления чисел в двоичной системе счисления, с фиксированной запятой. Операции над числом с плавающей запятой выполняются при помощи подпрограмм, как псевдооперации.

3. Память машины состоит из оперативного запоминающего устройства, выполненного на ферритовых сердечниках, емкостью 1024 оперативных 36-разрядных двоичных ячеек и 5 фиксированных ячеек и долговременного запоминающего устройства на окисферах емкостью 16384 ячеек.

4. Устройство управления основано на принципе микропрограммного управления с использованием для хранения микропрограмм первых 2048 адресов долговременного запоминающего устройства, имеющих 72 двоичных разряда.

5. Операции арифметического устройства, все передачи, записи в оперативное запоминающее устройство и выдача кодов из запоминающих устройств производится параллельным способом.

6. Информация может вводиться в машину с электрической пишущей машинки "Consul -254" или с перфорированной бумажной ленты с помощью трансмиттера FS-1500. Вывод результатов производится на буквенно-цифровую печать пишущей машинки "Consul-254" или на перфоленту перфоратора ПЛ-80/8.

7. Средняя скорость вычислений с фиксированной запятой для операций типа сложения 2000 - 3000 оп./сек., для операций типа умножения - не менее 100 оп./сек.

Средняя скорость вычисления для операций с плавающей запятой 200 - 300 оп./сек.

8. Тактовая частота машины 50 кГц.
9. Машина может работать в трех режимах:
 - а) в счетном с использованием ее как клавишной счетной машины;
 - б) в универсальном с программированием в машинных командах и с возможностью автоматического перехода к другим режимам;
 - в) в режиме автоматического программирования.

ЛАБОРАТОРНАЯ РАБОТА № I

БЛОК-СХЕМА ЭЦВМ "НАИРИ-С"

I. ОПИСАНИЕ ОСНОВНЫХ УЗЛОВ МАШИНЫ

Блок-схема (рис. I) включает в себя следующие узлы:

- 1) арифметическое устройство (АУ);
- 2) устройство управления (УУ);
- 3) оперативное запоминающее устройство (ОЗУ);
- 4) долговременное запоминающее устройство (ДЗУ);
- 5) внешние устройства (ВУ);
- 6) пульт управления (ПУ).

§ I. Арифметическое устройство

Арифметическое устройство включает в себя один регистр - сумматор (СМ), который является одновременно регистром числа как для ОЗУ, так и для ДЗУ. Арифметическое устройство параллельного типа со сквозным переносом состоит из 37 разрядов, 34 из которых отводятся под мантиссу, 35 и 36-й - знаковые разряды и 37-й - дополнительный разряд. Сумматор имеет следующие связи:

1. ПУ → СМ. Код, набранный на клавиатуре набора кода пульта управления, пересыпается путем нажатия кнопки "ПУ → СМ" в сумматор.

2. ПрК 8 + 25 разр. → СМ.
3. ПрК 12 + 25 разр. → СМ.

Эти две связи применяются при выполнении операции с модификациями *п* и *к*. В этом случае двоичный код, содержащий

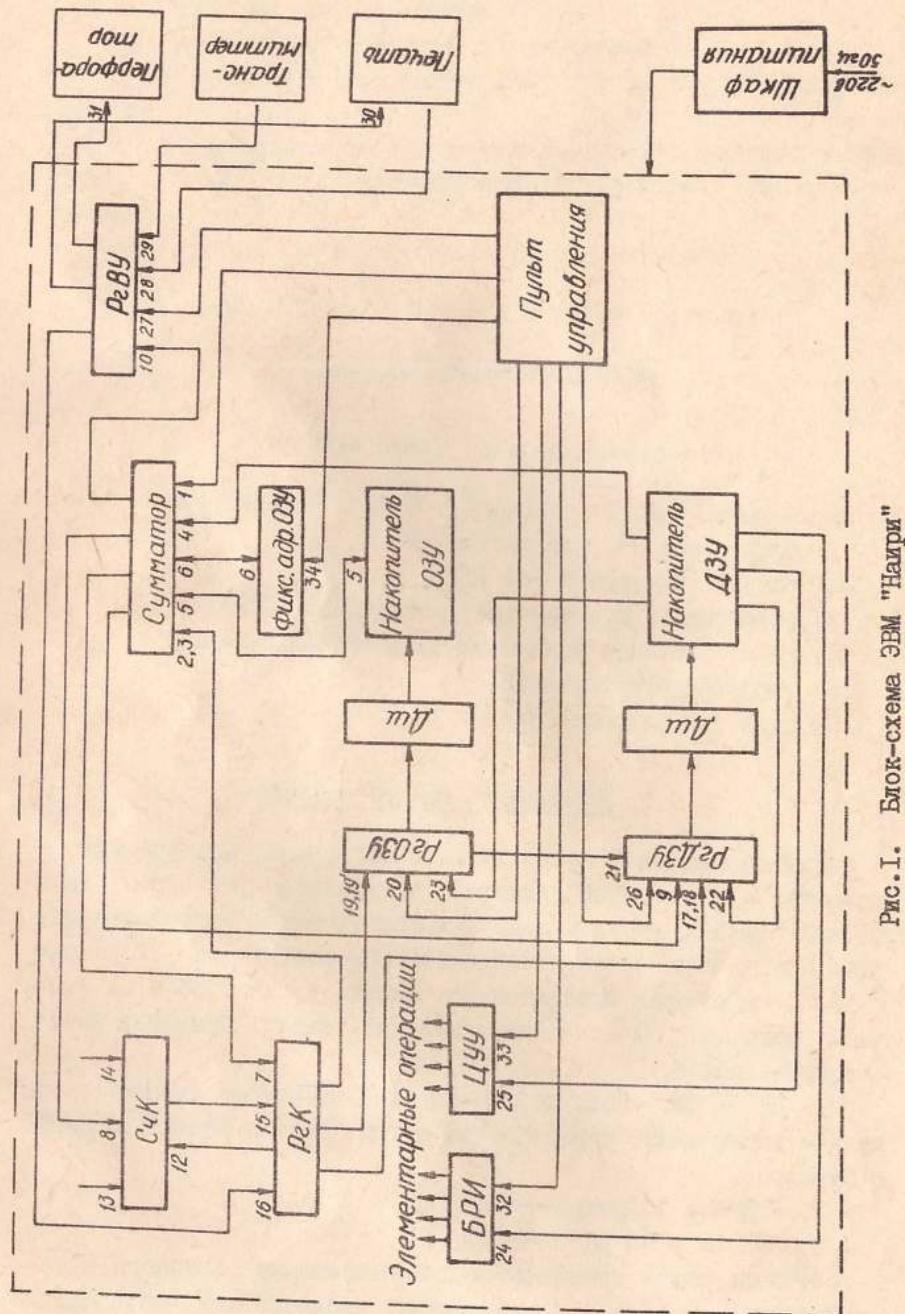


Рис. I. Блок-схема ЭМ "Нарри"

ся в 12-25-м разрядах регистра команд, пересыпается в младшие разряды сумматора и операция производится над полученным в сумматоре числом и числом, хранящимся в ячейке ЗУ, адрес которой записан в I - 7 или I - II разрядах РГК.

4. ДЗУ → СМ. Двоичный код, записанный по адресу, указанному в регистре ДЗУ, пересыпается в СМ.

5. ОЗУ → СМ. При операции "Запись" осуществляется пересылка СМ → ОЗУ; при операции "Чтение" — пересылка ОЗУ → СМ. Адрес ячейки ОЗУ, участвующей в операции, находится в регистре адреса ОЗУ.

6. Фиксированный адрес → СМ. Код, хранящийся в фиксированном адресе, пересыпается в сумматор (фиксированные адреса будут рассмотрены в § 3).

7. СМ → РГК. Пересыпается код, считанный из ОЗУ.

8. СМ I + I4 разр. → СЧК. Используется при условных и безусловных переходах и переадресациях.

9. СМ 34 + 36 разр. → РГДЗУ. Применяется для получения нужного адреса ячейки ДЗУ при различных модификациях операции.

10. СМ I + 6 разр. → РГВУ. Используется при операциях вывода на печать.

Коды с одного регистра на другой передаются мощными разрешающими импульсами, вырабатываемыми в УУ.

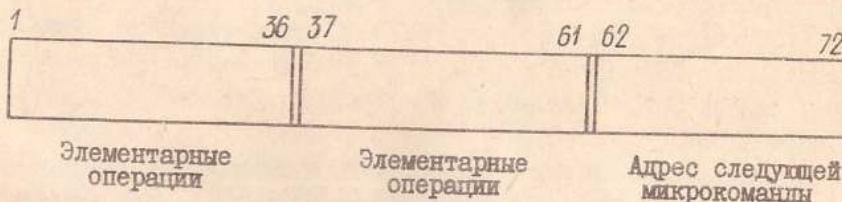
В СМ выполняются следующие операции: сложение по модулю 2; сдвиг кода влево; сдвиг кода вправо; логическое сложение.

§ 2. Устройство управления

Устройство управления предназначено для автоматического управления вводом информации в машину; автоматического выполнения введенной в машину программы; вмешательства оператора в работу машины; автоматического вывода полученных результатов.

Устройство управления микропрограммное (управление всеми процессами, протекающими в машине, осуществляется микрокомандами, прошитыми в ДЗУ). Каждая микрокоманда состоит из двух частей: первая несет в себе информацию о работе машины в данном такте, т.е. о тех элементарных операциях, которые необходимо выполнить машине в течение такта; вторая несет информа-

цию об адресе следующей микрокоманды, что позволяет осуществить последовательное выполнение микрокоманд любой микропрограммы. Микрокоманда, прошитая по какому-то адресу ДЗУ, имеет следующую структуру:



Например, по I87-му адресу ДЗУ прошита 72-разрядная микрокоманда, которая:

- 1) выполняет установку в "0" СМ (операция осуществляется I9 и 20-м разрядами);
- 2) дает разрешение на сдвиг вправо (8-й разряд);
- 3) устанавливает в "1" триггер удвоения (2-й разряд);
- 4) определяет адрес следующей микрокоманды (64 - 67 и 68-й разряды), т.е. I88-й адрес.

Устройство управления состоит из следующих блоков:

- a) счетчик команд (СЧК);
- б) регистр команд (РГК);
- в) центральное устройство управления (ЦУУ);
- г) триггер удвоения ($Tg_{удв}$);
- д) блок распределения импульсов (БРИ).

Счетчик адреса команд. I4-разрядный СЧК предназначен для счета и хранения адреса команды. Адрес следующей команды получается путем добавления единицы в СЧК. Он используется также для подсчета количества циклов при выполнении операций умножения, деления и т.д. СЧК имеет следующие связи с другими устройствами машины.

8. СМ I + I4 разр. —> СЧК *I).

II. ПУ —> СЧК. Код с клавиатуры пульта управления нажатием кнопки "ПУ —> СЧК" пересыпается в счетчик команд.

¹⁾ Здесь и далее значок "*" показывает, что пересылка рассматривалась ранее.

12. РГК 27 + 32 разр. —> СЧК I + 6,9,13 разр. Содержимое 27 - 32-го разрядов регистра команд пересыпается в младшие разряды СЧК с добавлением единиц в 9 и 13-й разрядах. В результате образуется адрес ДЗУ, с которого начинается программа выполнения операции, определяемой 27 - 32-м разрядами РГК.

15. СЧК —> РГК I2 + 25 разр. Применяется при выполнении операции "Переход с возвратом" для формирования команды возврата из подпрограммы на основную программу.

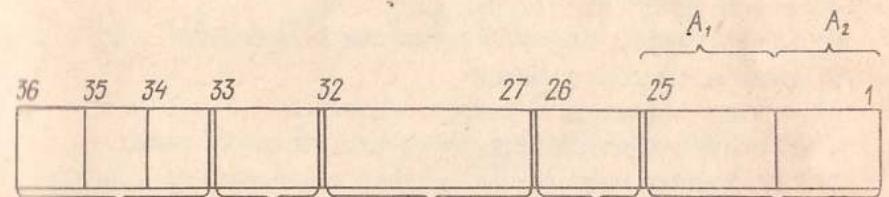
В СЧК приходят также импульсы:

I3. Уст."0" СЧК.

I4. + I СЧК.

Первый импульс может приходить с ПУ при нажатии кнопки "Уст."0" СЧК" или с УУ после выполнения пересылки "СЧК —> РГК I2 + 25 разр.". Второй импульс приходит из УУ после выполнения пересылки "СМ —> РГК".

Регистр команд. 36-разрядный регистр команд предназначен для хранения команд во время их исполнения. РГК имеет следующее разбиение разрядной сетки:



Модификация	Признак псевдо-операции	КОП	Признак переадресации	Разбиение в зависимости от модификации
-------------	-------------------------	-----	-----------------------	--

Например, при модификации Н I - 10-й разряд РГК служат для хранения второго адреса команды. По этому адресу выбирается число из ОЗУ.

II-й разряд не используется, поскольку для нумерации всех адресов ОЗУ хватает 10 двоичных разрядов, так как $2^{10} - 1 = 1023$. I2 - 25-й разряды служат для хранения первого адреса команды. (Остальные модификации подробно рассмотрены во второй части).

РГК имеет следующие связи:

2. ПрК \rightarrow СМ*.
7. СМ \rightarrow ПрК*.
- I2. ПрК 27 + 32 разр. \rightarrow СЧК I + 6, 9, I3 разр.*.
- I5. СЧК \rightarrow ПрК I2 + 25 разр.*.

I6. ПрВУ \rightarrow ПрК I2 + I8, 23 разр. При вводе информации с аппаратов внешних устройств 6-позиционный код из регистра ВУ пересыпается в I2 - I8-й разряды регистра команд с добавлением единицы в 23-й разряд; тем самым образуется адрес ДЗУ, в котором хранится внутренний код вводимого символа.

I7. ПрК 34 + 36 разр. \rightarrow ПрДЗУ I + 3 разр. Применяется при формировании адреса ДЗУ, к которому необходимо перейти в зависимости от модификации, определяемой содержимым 34 - 36-го разрядов.

I8. ПрК I2 + 25 разр. \rightarrow ПрДЗУ.

I9. ПрК I2 + 25 разр. \rightarrow ПрОЗУ. Код, содержащийся в I2 - 25-м разрядах ПрК, пересыпается в ПрДЗУ, если код адреса $A_1 > 2048$ и в ПрОЗУ, если адрес $A_1 < 2048$, где A_1 - I-й адрес команды.

Центральное устройство управления обеспечивает:

- а) чтение из ДЗУ микрокоманд и чисел;
- б) засыпку адреса следующей микрокоманды в ПрДЗУ и ПрОЗУ;
- в) работу триггера удвоения;
- г) условные переходы в микропрограммах.

В ЦУУ вырабатываются следующие управляющие сигналы:

I) "Разрешение микрокоманд" - дает разрешение на прохождение считываемых из ДЗУ сигналов;

2) "Разрешение прием СМ" - разрешает прием считанных с ДЗУ сигналов в СМ;

3) ПрК I2 + 25 разр. \rightarrow ПрЗУ. Содержимое I2 - 25-го разрядов ПрК определяет первый адрес команды A_1 . По этому адресу может выбираться число либо из ОЗУ, либо из ДЗУ. Эти адреса будут отличаться наличием единицы в одном из старших разрядов A_1 , т.е. в разрядах 23 - 25 ПрК. Если в одном из них окажется единица, то это означает, что $A_1 > 2048$; при отсутствии единицы в этих разрядах $A_1 < 2048$. При $A_1 > 2048$ элементарная операция "ПрК I2 + 25 разр. \rightarrow ПрЗУ" образует передачу "ПрК I2 + 25 разр. \rightarrow ПрДЗУ", при $A_1 < 2048$ - "ПрК I2 + 25 разр. \rightarrow ПрОЗУ".

При передаче "ПрК I2 + 25 разр. \rightarrow ПрДЗУ" адрес следующей микрокоманды засыпается в предварительно очищенный ПрОЗУ. После чтения из ДЗУ осуществляется передача "ПрОЗУ \rightarrow ПрДЗУ", чем восстанавливается прерванная последовательность выполнения команд (микрокоманд).

Триггер удвоения. Применение триггера удвоения позволяет без увеличения разрядности ДЗУ увеличить количество элементарных операций. Принцип работы триггера удвоения показан на рис.2.

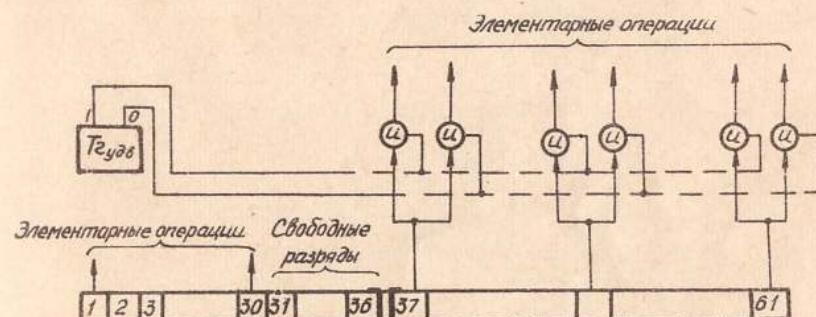


Рис.2. Распределение элементарных операций по разрядам ДЗУ

I - 30-й разряд ячейки ДЗУ (рис.2) являются независимыми элементарными операциями; I и 2-й разряды служат для управления триггером удвоения; 3I - 36-й разряды - резервные на случай увеличения количества элементарных операций. Разряды 37 - 6I-й вырабатывают удвоенное количество элементарных операций, т.е. на 25 разрядах получается 50 элементарных операций. Действительно, как видно из рисунка, выход каждого из разрядов поступает на схему совпадения двух независимых усилителей, и в зависимости от состояния триггера удвоения импульс появится на выходе одного из этих усилителей.

Если в какой-либо микрокоманде используется элементарная операция, зависящая от триггера удвоения, то установка триг-

гера в единичное или нулевое состояние происходит на один такт раньше того такта, в котором будет использована эта элементарная операция.

Блок распределения импульсов состоит из отдельных узлов, каждый из которых вырабатывает какую-то определенную операцию. Схемы узлов запускаются разрядами ДЗУ. Например, элементарная операция пересылки "ПрК 12 + 25 разр. → СМ I + I4 разр." образуется по схеме, приведенной на рис.3.

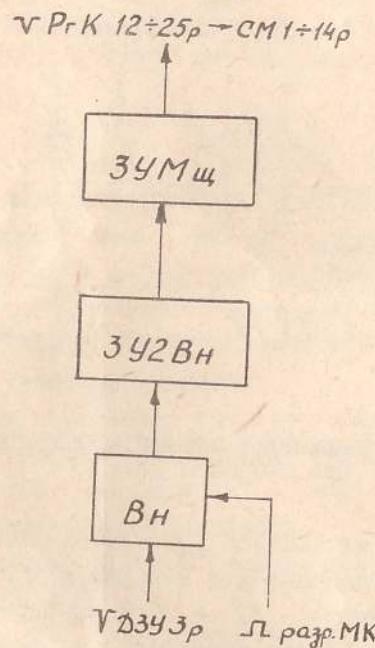


Рис.3.

ЗУМщ - мощный усилитель;

ЗУ2Вн - усилитель-формирователь;

Вн - схема совпадения, входящая в ячейку ЗУ2Вн;

✓ ДЗУ Зр - импульс с 3-го разряда ДЗУ - поступает на импульсный вход Вн;

Г разр.МК - импульс разрешения микрокоманд, вырабатываемый в ЦУУ - поступает на потенциальный вход Вн

§ 3. Оперативное запоминающее устройство

Оперативное запоминающее устройство служит для приема, хранения и выдачи информации в виде программ, подпрограмм, чисел и результатов вычислений.

Емкость ОЗУ - 1029 36-разрядных чисел, пять из которых являются фиксированными адресами. Возможна одновременная запись по двум фиксированным адресам. Время обращения к ОЗУ 24 мксек.

ОЗУ построено по принципу Σ на ферритовых сердечниках с прямоугольной петлей гистерезиса.

Блок-схема ОЗУ приведена на рис.4.

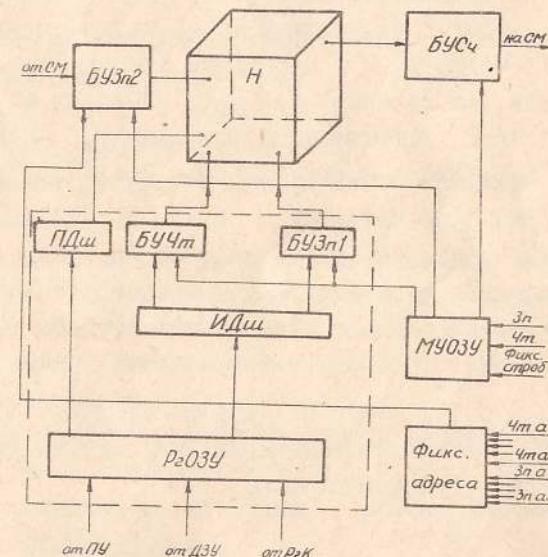


Рис.4. Блок-схема ОЗУ:

Н - накопитель;

БУЗп2 - блок усилителей записи 2;

БУСч - блок усилителей считывания;

ПДш - потенциальный дешифратор;

БУЧт - блок усилителей чтения;

БУЗп1 - блок усилителей записи 1;

ИДш - импульсный дешифратор;

МУОЗУ - местное устройство управления ОЗУ;

РэОЗУ - регистр адреса ОЗУ;

Фикс.адреса - фиксированные адреса

Выборка адреса из накопителя производится с помощью ШИ на 64 выхода и ИШ на 16 выходов.

Регистр адреса ОЗУ состоит из II разрядов, 4 первых из которых связаны с ИДШ, 6 следующих - с ПДШ, а II-й разряд используется в устройстве управления.

РгОЗУ имеет следующие связи:

I9. РгК I + II разр. → РгОЗУ. Содержимое I - II-го разрядов РгК, т.е. адрес A_1 команды, находящейся в РгК, пересыпается в регистр адреса ОЗУ для последующего чтения из этого адреса.

I9'. РгК I2 + 22 разр. → РгОЗУ. Пересылка осуществляется для чтения содержимого адреса A_1 .

20. ПУ → РгОЗУ. Занесение в РгОЗУ кода с клавиатуры ПУ.

21. РгОЗУ → РгДЗУ. Эта пересылка применяется, когда адрес A_1 команды, находящейся в РгК, > 2048, т.е. является адресом ДЗУ. Этой пересылке предшествуют микрооперации "Уст. "0" РгОЗУ", "РгДЗУ → РгОЗУ", "РгК I2 + 25 разр. → РгДЗУ" и "Чтение ДЗУ".

Импульсный дешифратор - двухступенчатый, вырабатывает мощные импульсы "Запись I" и "Чтение". Окончательным звеном ИШ являются блоки усилителей "Запись I" и "Чтение".

Потенциальный дешифратор - двухступенчатый, вырабатывает потенциальные сигналы для клапанирования адресных трансформаторов.

Блок усилителей Зп2 предназначен для записи в ОЗУ кода, находящегося в сумматоре при поступлении сигнала "Запись 2" из МУОЗУ.

Блок считывания предназначен для усиления считанных с ОЗУ сигналов с передачей их на кодовые шины сумматора.

Местное управление ОЗУ предназначено для формирования сигналов "Чтение", "Запись" и "Строб" в соответствии с поступающими из ЦУУ управляющими сигналами "Чтение", "Запись".

Блок фиксированных адресов. Фиксированные адреса служат в качестве самостоятельных регистров. Они используются при выполнении различных машинных операций. Например, при выполнении операции умножения используются четыре фиксированных адреса а1 - а4:

в а1 располагается множитель, т.е. содержимое ячейки с адресом A_1 ;

в а2 располагается множимое, т.е. содержимое ячейки с адресом A_2 ;

в а3 посылается сумма частичных произведений (СЧП);

в а4 хранится множимое с обратным знаком.

При выполнении операции "Чтение" (ввод) используются также четыре фиксированных адреса а1 - а3 и а5:

в а1 располагается содержимое A_1 ;

в а2 располагается содержимое A_2 ;

в а3 хранится результат предыдущей операции;

в а5 посылается "0" и "1" в I8-й разряд в зависимости от регистра вводимого символа (I8-й разряд служит для образования адреса ячейки ДЗУ, в которой проходит внутренний код вводимого символа).

С ЦУУ на каждый фиксированный адрес поступают свои импульсы "Чтение" или "Запись".

При чтении из фиксированного адреса одновременно с сигналом "Чтение" должен подаваться сигнал "Строб фиксированный", иначе будет происходить не чтение, а просто стирание содержимого фиксированного адреса.

§ 4. Долговременное запоминающее устройство

Долговременное запоминающее устройство предназначено для хранения и выдачи информации в виде программирующей программы, микропрограмм, подпрограмм, констант и т.д.

ДЗУ имеет следующие связи:

22. Накопитель ДЗУ 62 + 72 разр. → РгДЗУ I + II разр. Пересылка осуществляется через линию задержки для образования адреса следующей микрокоманды при выполнении микропрограммы.

23. Накопитель ДЗУ → РгОЗУ. Применяется при чтении из ДЗУ какой-либо константы.

24. Накопитель ДЗУ → БРИ. Применяется для образования сигналов элементарных операций (см. описание БРИ).

25. Накопитель ДЗУ → ЦУУ. Пересылка применяется для образования управляющих сигналов.

26. ПУ → РгДЗУ. Для засылки кода в РгДЗУ с пульта.

4. Накопитель ДЗУ → СМ.*

§ 5. Внешние устройства

Внешние устройства предназначены для ввода в ЭВМ и вывода из нее информации.

Для связи внешних устройств с машиной служит регистр внешних устройств (РгВУ):

16. РгВУ → РгК I2 + 18,23 разр.*.

10. СМ I + 6 разр. → РгВУ I + 6 разр.*.

27. ПУ → РгВУ I + 6 разр.

28. Печатающее устройство → РгВУ I + 6 разр.

29. Трансмиттер → РгВУ I + 6 разр.

30. РгВУ I + 6 разр. → печатающее устройство.

31. РгВУ I + 6 разр. → перфоратор.

§ 6. Пульт управления

Пульт управления предназначен для включения и выключения машины, выбора режима ее работы, световой сигнализации состояний регистров всех устройств машины, для оперативного вмешательства в ее работу, а также для профилактических и наладочных работ.

Пульт управления имеет:

а) управляющие связи

32. ПУ → БРИ.

33. ПУ → ЦУУ.

б) связи занесения кодов в регистры устройств

36. ПУ → РгДЗУ.

II. ПУ → СЧК.

20. ПУ → РгОЗУ.

34. ПУ → фикс.адр.ОЗУ.

I. ПУ → СМ.

27. ПУ → РгВУ.

II. РАБОЧИЙ ЦИКЛ МАШИНЫ

Каждая операция в машине выполняется за некоторое время, называемое рабочим циклом машины. Рабочий цикл включает в себя 5 основных этапов, осуществляемых своими микроPROGRAMMAMI, прошитыми в ДЗУ:

- 1) выборку команды;
- 2) выборку I-го числа в зависимости от модификации команды;
- 3) анализ n, S ;
- 4) выполнение операции;
- 5) запись результатов.

В качестве примера рассмотрим блок-схему выполнения операции "Сложение", которая представляется во внешнем коде символом "с", а во внутреннем коде единицей в 27-м разряде РгК.

Описание блок-схемы операции "Сложение" (рис.5)

1. Первому пункту блок-схемы предшествуют две микрокоманды: "Уст."0" ТГ_{удб}" и "Уст."0" РгК I2 + 25 разр.". В этом же пункте осуществляется микрокоманда "Уст."0" РгОЗУ".

2. Анализ 23 - 25-го разрядов РгК. Если в одном из этих разрядов имеется единица, то, следовательно, это адрес ДЗУ. Тогда схемным путем осуществляется пересылка "РгДЗУ → РгОЗУ", и машина переходит к 4 и 6-му пунктам блок-схемы. Если же в 23 - 25-м разрядах РгК нули, то осуществляется переход к 3 и 5-му пунктам.

3 и 4. В эти пунктах, кроме пересылки "РгК I2 + 25 разр. → РгЗУ", осуществляются микрооперации "Уст."0" СМ", "Уст."0" РгК I + II разр." и "Уст."0" РгК 26 + 36 разр.".

5 и 6. Здесь осуществляется чтение в СМ из ЗУ.

7. Анализ 26-го разряда сумматора, т.е. анализируется признак переадресации. При наличии "1" в 26-м разряде СМ осуществляется переход к пункту 8. При 26 разр. = 0 машина от пункта 7 переходит к 9-му пункту, минуя 8-й.

8. Происходит суммирование содержимого СМ и содержимого первой ячейки ОЗУ, которая используется в машине в качестве индексного регистра. Выполняется это следующими микрооперациями:

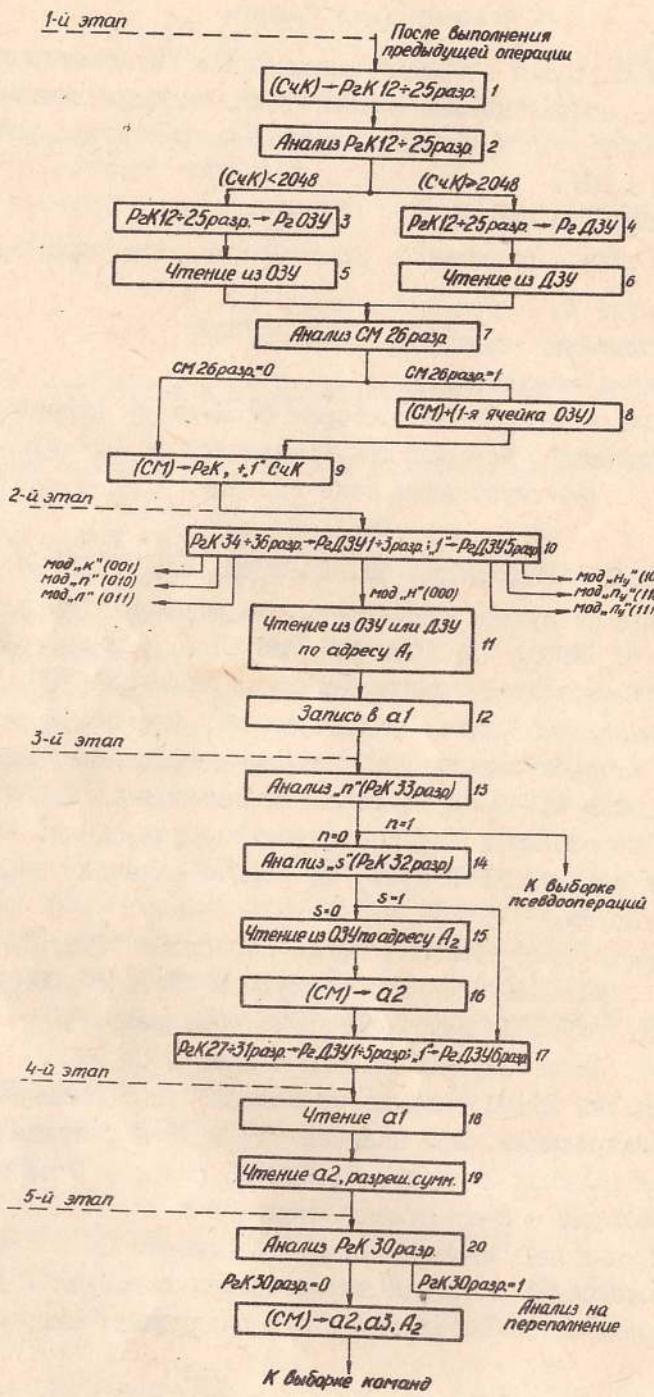


Рис.5. Блок-схема операции "Сложение"

а) "Чтение аI" (поскольку строб фиксированный не подается, то это очистка аI);

б) "Запись аI" (команда, находящаяся в СМ, записывается в аI);

в) "Уст."0" РrOЗУ";

г) "Уст."I" ТГудб";

д) "Уст."I" РrOЗУ";

е) "Уст."0" СМ";

ж) "Чтение ОЗУ" (в СМ читается содержимое первой ячейки ОЗУ, так как в РrOЗУ находится "I");

з) "Чтение аI" со стробом фиксированным, одновременно подается сигнал "Разрешение суммирования";

и) "Уст."0" ТГудб".

9. Здесь осуществляется пересылка команды из СМ в РrK, устанавливается в "0" РrOЗУ и затем добавляется единица в С4К.

10. Осуществляется "Уст."0" СМ" и пересылка содержимого 34 - 36-го разрядов РrK в I - 3-й разряды РrDЗУ, т.е. пересыпается содержимое разрядов, отведенных под модификацию. В результате получается адрес микрокоманды, к которой должна перейти машина при данной модификации. Наличие "I" в 5-м разряде РrDЗУ указывает, что адреса всех модификаций находятся в пределах I6 - 23-й ячеек (модификация Н в ячейке I6, модификация к в ячейке I7 и т.д.).

11. При модификации Н (000) происходит пересылка "РrK I2 + 25 разр. → РrDЗУ" и "Чтение аI", затем чтение из ОЗУ или ДЗУ в зависимости от 23 - 25-го разрядов РrK.

12. Содержимое сумматора, т.е. прочитанный из ЗУ код, записывается в фиксированный адрес аI. Устанавливается в "I" триггер удвоения.

13. Анализируется 33-й разряд РrK, т.е. признак псевдоопераций, и осуществляются микрооперации: "Уст."0" СМ" и "Уст."0" РrOЗУ". При n = 1 машина переходит к анализу и выработке начального адреса микропрограммы данной псевдооперации; при n = 0 переходит к пункту 14.

14. Анализируется 32-й разряд РrK, т.е. признак модификации кода операции (C; C₁; C₂ или C₃) и происходит "Уст."0" ТГудб".

При S = 0 осуществляются 15 и 16-й пункты блок-схемы, при S = 1 происходит переход от 14-го сразу к 17-му пункту.

15. а) "Чтение a2", т.е. очистка a2;
б) "ПрК I + II разр. → ПрОЗУ", т.е. в ПрОЗУ посылается второй адрес (A_2) команды;
в) "Чтение ОЗУ".

16. "Запись a2", т.е. "СМ → a2" и "Уст."0" ПрОЗУ".

17. Выполняется "Уст."0" СМ" и содержимое 27 - 31-го разрядов ПрК, т.е. код операции, посыпается в младшие разряды ПрДЗУ с добавлением "1" в 6-й разряд ПрДЗУ. Тем самым образуется адрес ячейки ДЗУ, с которого начинается микропрограмма, осуществляющая данную операцию.

18. Чтение из фиксированного адреса a1 со стробом фиксированным, т.е. в a1 в данном случае находится код, который был записан по адресу A, выполняемой команды.

19. Чтение из фиксированного адреса a2 со стробом, одновременно подается сигнал "Разрешение суммирования".

20. Осуществляется анализ 30-го разряда ПрК для определения модификации кода операции. Если ПрК 30 разр. = 0, то код операции C_1 , C_2 и анализа результата на переполнение не происходит. Если ПрК 30 разр. ≠ 0, то, следовательно, код операции C_2 или C_3 и машина анализирует результат на переполнение.

21. В этом пункте происходит запись результата в a2, a3; в ПрОЗУ засыпается содержимое ПрК I + II разр., т.е. адрес A_2 , и дается сигнал "Запись".

Этим заканчивается рабочий цикл и машина уходит опять к микропрограмме "Выборка команд", т.е. к I-му этапу.

Контрольные вопросы

1. Технические данные машины.
2. Технические данные ОЗУ.
3. Микропрограммный способ управления.
4. Структура микрокоманд.
5. Работа триггера удвоения.
6. Блок-схема выполнения операций.

Рабочее задание

1. Выполнить с пульта операцию "Сложение" в тактовом режиме.(Внешний код команды, адреса ячеек и коды чисел, записанных в ячейках, даются преподавателем).

2. Состояние регистров за каждый такт выполнения операции занести в таблицу.

Номер такта	Регистр				
	ОЗУ	ДЗУ	СЧК	ПрК	СМ
I					
2					
3					

3. При выполнении I и 2-го пунктов рабочего задания проследить по блок-схеме порядок выполнения операции.

4. Составить микропрограмму выполнения операции, заданной преподавателем.

Содержание отчета

1. Блок-схема и ее описание.
2. Таблица состояния регистров при выполнении операции, заданной преподавателем, с пояснением одного из тактов.
3. Микропрограмма выполнения заданной операции.

ЛАБОРАТОРНАЯ РАБОТА №2

ВНЕШНИЕ УСТРОЙСТВА И ПУЛЬТ УПРАВЛЕНИЯ

§ I. Внешние устройства

Назначение и основные технические данные. Внешние устройства предназначены для ввода информации в машину и вывода результатов вычисления.

Ввод информации может осуществляться:

- а) с печатающего устройства (электрическая пишущая машина "Consul-254") в виде букв русского и латинского алфавита и десятичных цифр (клавиатура печатающего устройства показана на рис. I);

б) с фототрансмиттера (F5-I500) в виде 6-позиционных кодов, пробитых предварительно на бумажной ленте (коды всех символов приведены в приложении I).

Вывод результатов осуществляется:

- а) на печатающее устройство ("Consul-254") в виде букв и цифр;
 б) на перфоратор ("ПЛ-80/8") в виде 6-позиционных кодов.
 Скорость ввода: с печатающего устройства 7 - 8 зн./сек.;
 с фототрансмиттера 100 - 200 зн./сек.
 Скорость вывода: на печатающее устройство 8 - 10 зн./сек.,
 на перфоратор 70 зн./сек.

Блок-схема ВУ приведена на рис.2. Кроме трех перечисленных ранее аппаратов, в состав ВУ входит:

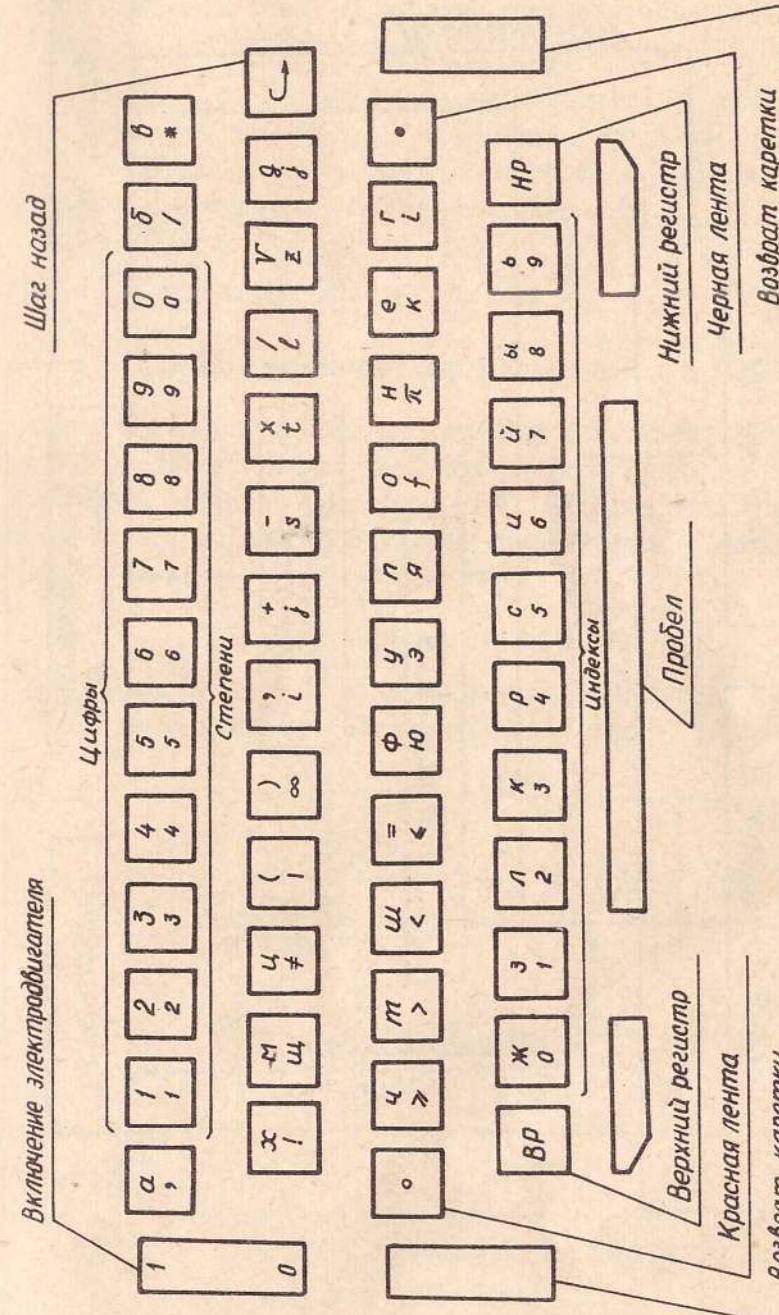


Рис. I. Клавиатура печатающего устройства

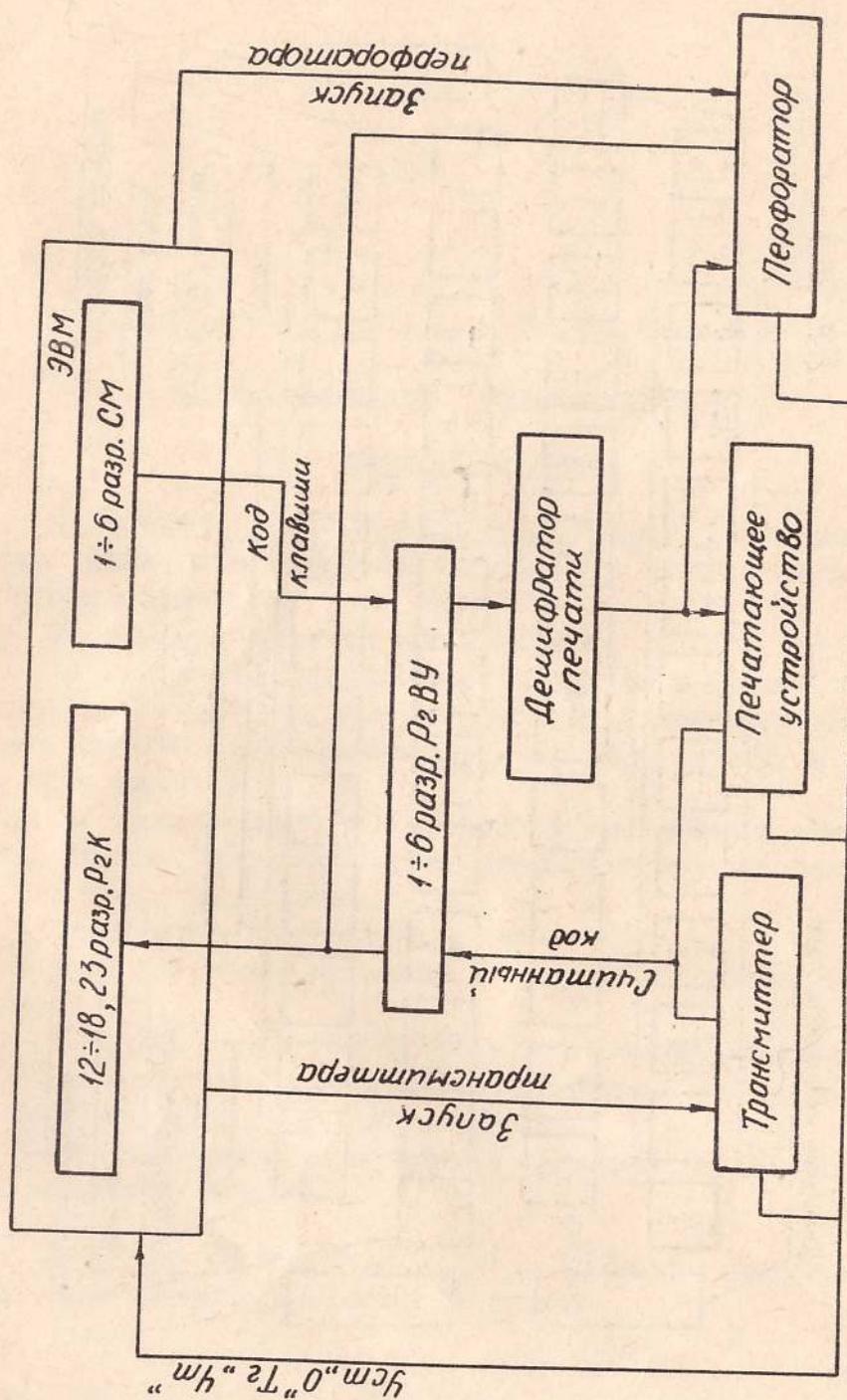


Рис.2. Блок-схема ВУ

регистр внешних устройств (РгВУ);
электрическая схема управления печатью (ЭСУПЧ);
электрическая схема управления трансмиттером (ЭСУТ);
электрическая схема управления перфоратором (ЭСУПФ).

Регистр ВУ. 6-разрядный регистр ВУ является буферным регистром как между ВУ и машиной, так и между аппаратами ВУ. РгВУ собран на 6 триггерных ячейках. Каждый символ может быть представлен в РгВУ 6-разрядным кодом.

Электрическая схема управления печатью состоит из двух частей:

I. Схема образования кодов и запись их в РгВУ

При нажатии любой клавиши комбинатор печатающего устройства образует 6-разрядный код, который поступает на потенциальные входы вентилей триггеров РгВУ. Одновременно образуется импульс записи в РгВУ, поступающий на импульсные входы тех же вентилей (рис.3).

2. Схема автоматической печати

В печатающей машинке каждому символу клавиатуры соответствует свой электромагнит. При поступлении кода символа в РгВУ

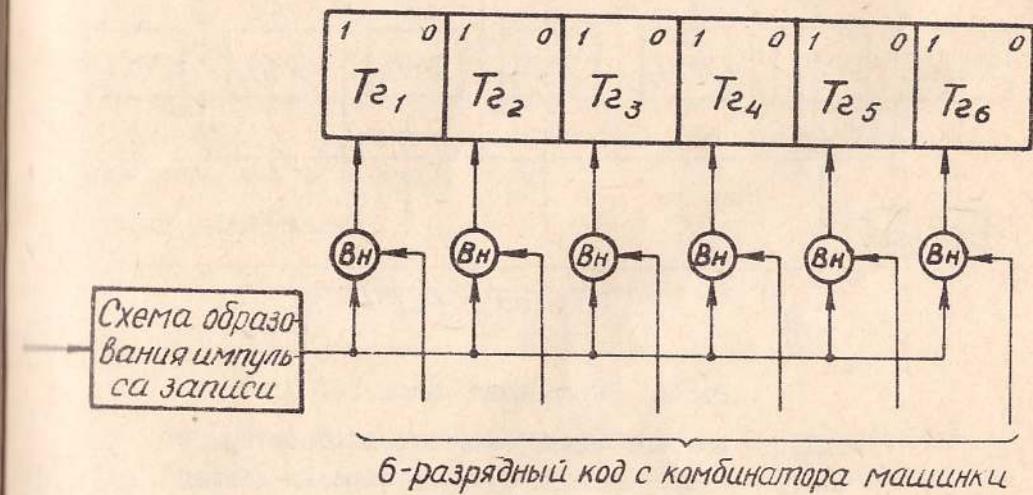


Рис.3. Схема образования кодов

осуществляется выборка электромагнита, соответствующего данному символу. Выборка происходит при помощи дешифратора, расположенного в машинке. В результате анализа I - 3-го разрядов РгВУ выбирается одна из вертикальных шин дешифратора, а анализа 4 - 6-го разрядов - одна из горизонтальных шин. Срабатывает выбранный электромагнит и пропечатается символ, соответствующий поступившему в РгВУ коду.

Электрическая схема управления трансмиттером. При считывании информации с перфоленты на выходе трансмиттера появляются отрицательные импульсы соответственно пробитому на перфоленте символу. Эти импульсы через инверторы поступают на потенциальные схемы вентиляй триггеров РгВУ. На импульсные входы этих вентиляй поступает импульс записи, т.е. получается схема, аналогичная схеме ЭСУПЧ.

Электрическая схема управления перфоратором. Вся работа схемы (рис. 4) осуществляется сигналами от датчиков перфоратора. При приходе импульса "Вывод" мотор перфоратора включается и сигналы с датчиков начинают нарастать. Как только их амплитуда достиг-

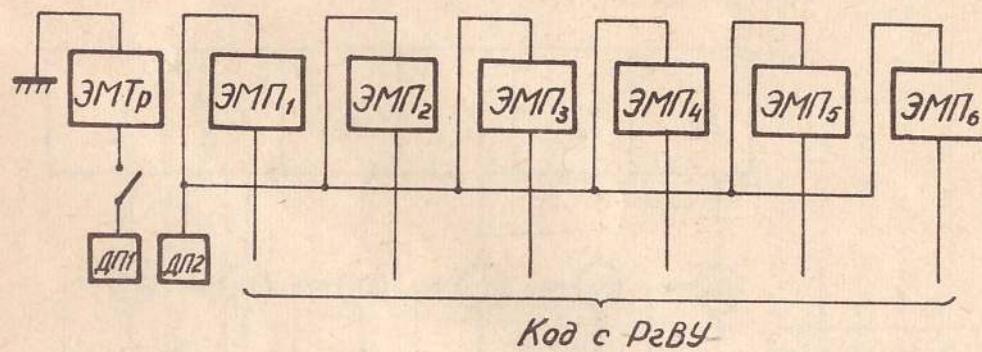


Рис.4. Упрощенная схема ЭСУПФ:

ЭМП₁₋₆ - кодовые электромагниты перфоратора;
ЭМТр - электромагнит транспортировки ленты;
ДП1-ДП3 - датчики перфоратора

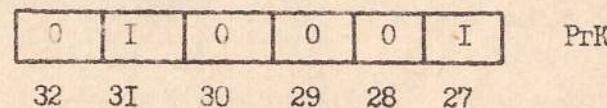
нет определенной величины, от датчика ДП2 поступает разрешение на пробивку кода, находящегося в РгВУ; одновременно пробивается отверстие синхродорожки.

По сигналу с датчика ДП1 происходит выключение кодовых электромагнитов перфоратора (ЭМП₁₋₆) и включается транспортировка ленты, отключение которой осуществляется по сигналу с датчика ДП3.

Образование кодов и запись их в ЭВМ. При нажатии любой из клавиш печатающего устройства с комбинатора печатающего устройства на вход РгВУ поступает внешний код данной клавиши, а в ЭВМ посылается импульс, устанавливающий в "0" триггер "Чтение" (Тг "Чт."), что позволяет выполнить микропрограмму "Чтение" с микрокомандой, осуществляющей передачу РгВУ → РгК (12 - 18, 23 разр.). Единица в 23-й разряд РгК добавляется для формирования адреса внутреннего кода данного знака, заранее прошитого в ДЗУ.

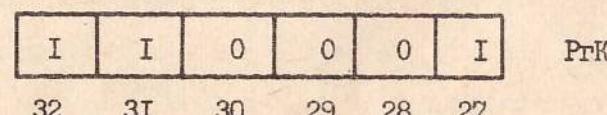
Вывод информации из ЭВМ осуществляется микропрограммой операции "Обращение", которая имеет две модификации:

Первая модификация



При нулевом значении 32-го разряда РгК выводу подлежит содержимое ячейки а1 (фиксированная ячейка), где представлен непосредственно код выводимого знака.

Вторая модификация



При единичном значении 32-го разряда РгК в а2 представлен промежуточный адрес, к которому микропрограммой "Обращение" добавляется число ($2^{\text{11}} + 2^7$), тем самым и формируется адрес выводимого знака, записанного в ДЗУ.

§ 2. Пульт управления

В состав пульта управления входят панель сигнализации (ПС) и панель управления (ПУ).

Панель сигнализации (рис.5) предназначена для включения и выключения питания, пуска и останова машины, выбора режима работы и для световой сигнализации. На панели сигнализации расположены клавиши:

- I. "Универсальный - счетный" - для работы машины в универсальном (клавиша не нажата) или в счетном режиме (клавиша нажата).
 2. "Выдача памяти" - для вызова подпрограммы "Выдача памяти".
 3. "Пуск I" - для вызова начала подпрограммы универсального или счетного режимов.
 4. "Останов" - для останова машины.
 5. "Пуск 2" - для пуска машины с текущего адреса СЧК (и адреса ДЗУ).
 6. "Печать и перфорация с трансмиттера".
 7. "Ввод с трансмиттера".
 8. "Вывод на печать".
 9. "Вывод на перфорацию".
 10. "Связь - автономный".
 - II. "Перфорация с трансмиттера".
 12. "Печать, перфорация".
 13. "Печать с трансмиттера".

На ПС расположено также 127 неоновых ламп, которые показывают состояние соответствующих триггеров следующих регистров: СМ (36 разрядов и дополнительный разряд), РтК (36 разрядов), СЧК (14 разрядов), РгДЗУ (14 разрядов), Рг0ЗУ (II разрядов).

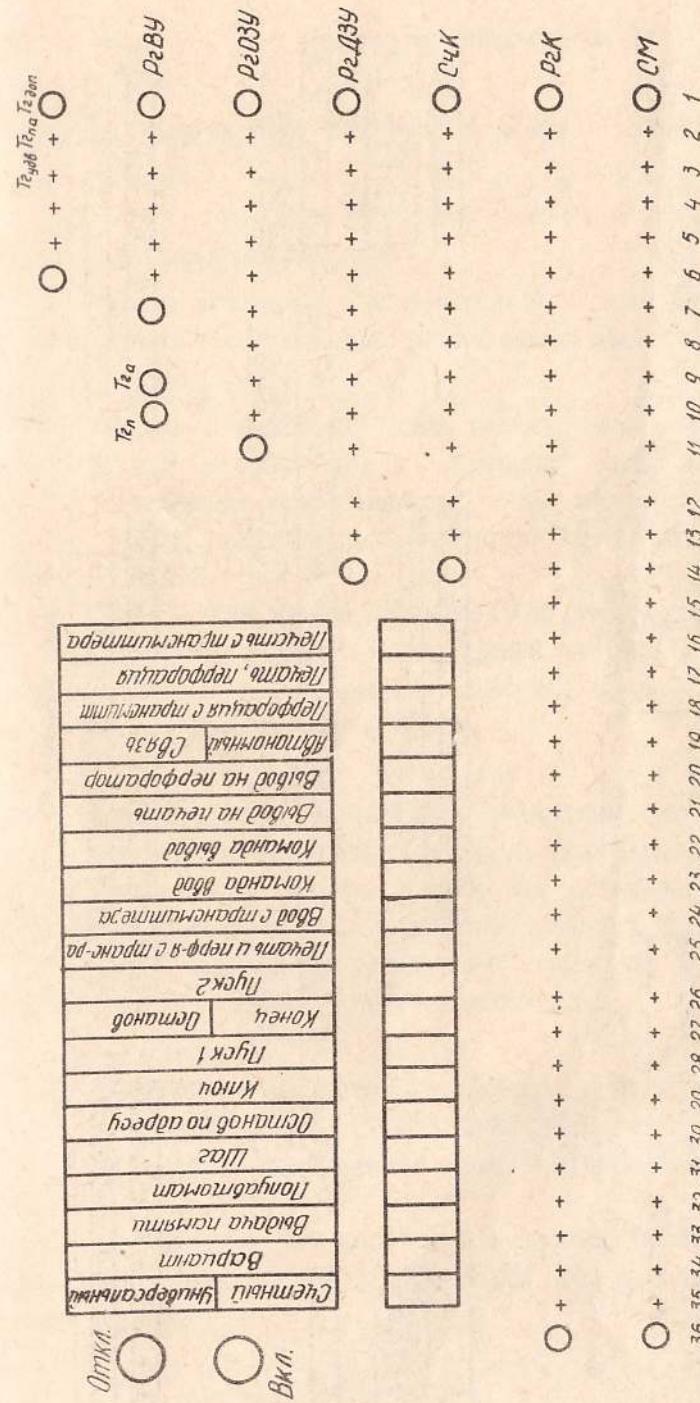


Рис. 5. Панель синтаксизаций

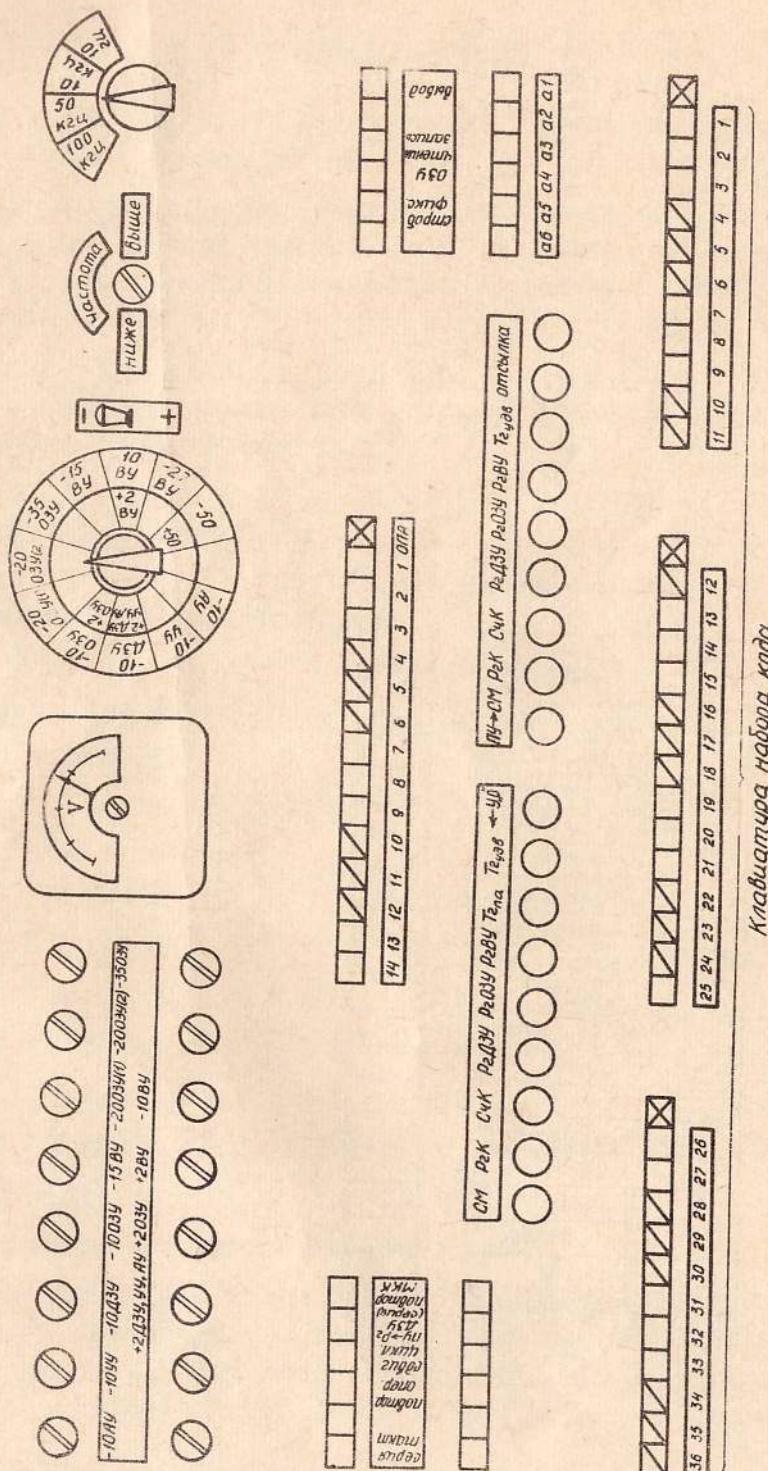


Рис.6. Панель управления

РгВУ (6 разрядов) и триггеров "Полуавтомат" (ТгПа), "Нечать" (ТгП) и Тг_{удб}.

Панель управления. На ПУ расположены (рис.6):

Переменные сопротивления, вольтметр и переключатели - для измерения и регулировки питающих напряжений.

Переменные сопротивления и переключатель - для регулировки частоты.

Клавиши:

- 1) "Серия - тактовый" - для работы машины в режиме "Серия" (клавиша не нажата) или "Тактовый" (клавиша нажата);
- 2) "Повторение микрокоманды" - для повторного выполнения одной и той же микрокоманды, прошитой по адресу, зафиксированному в РгДЗУ;
- 3) "ОЗУ" - для чтения и записи ОЗУ;
- 4) "Чтение - запись" - для чтения из ОЗУ (клавиша не нажата) или записи в ОЗУ (клавиша нажата) кода, находящегося СМ по адресу, указанному в РгОЗУ.

Клавиатура набора кода (КНК) - для записи кода в различные регистры. Нажатием одной клавиши подается разрешающий потенциал на вход триггеров данного разряда всех регистров. Засылка кода в тот или другой регистр обуславливается нажатием кнопки соответствующей передачи.

По цвету клавиатура разбита на группы по 3 клавиши для удобства набора числа в восьмеричном коде.

Кнопки:

- 1) "Уст."0" - для сброса в нулевое состояние соответствующим СМ, РгК, СЧК, РгДЗУ, РгОЗУ, РгВУ, ТгПа и Тг_{удб};
- 2) "Посылка" - для записи кода с ПУ соответственно в СМ, РгК, СЧК и т.д.;
- 3) "Отсылка" - для получения импульса "Отсылка", который используется для записи и чтения ОЗУ или одного из фиксированных адресов.

§ 3. Инструкция по работе на машине

Счетный режим. В счетном режиме ЭВМ используется как усовершенствованный арифмометр, т.е. вычисления производятся без предварительного программирования. При работе в этом режиме необходимо соблюдать следующие правила:

1. Всем функциям, корню ($\sqrt{}$ или $\sqrt[n]{}$) и переходу от градусной меры к радианной предшествует пробел, например:

пробел $\cos x =$ (при вычислении $\cos x$);

пробел \sin пробел $\cos x =$ (при вычислении $\sin \cos x$);

пробел $30^\circ 21' 16'' =$ (при переводе градусной меры в радианную);

пробел $\sqrt[3]{17}, 082 =$ (при вычислении корней);

пробел $\arccos x =$ (при вычислении $\arccos x$).

Примечание. При печатании функции необходимо после пробела отпечатать только первую букву, обозначающую функцию: для вычисления выражения $\sin \cos x$ необходимо напечатать "пробел δ пробел $\sin x$ "; для вычисления $\arccos x$ напечатать "пробел $\arccos x$ ".

2. Для присвоения букве численного значения печатается буква, знак " $=$ " и число, после чего дается возврат каретки (вк): $a = 5,017$ вк; $b = 0,455$ вк.

3. Для присвоения букве результата предыдущей операции печатается знак " $=$ " и выбранная буква; это присвоение используется для запоминания промежуточных результатов, так как в счетном режиме отсутствуют скобки.

4. Показатель, относящийся к функции, печатается непосредственно после символа функции; $(\sin x)^2$ печатается так: "пробел $\sin^2 x$ ".

Пример вычислений в счетном режиме. Необходимо вычислить выражение

$$\frac{x^5 - \sin^2(a-x^3)}{\sqrt{(a \operatorname{ctg} x)^2 - a^3}},$$

где $x = 0,17$; $a = 0,818$.

I) присвоение

$$x = 0,17 \text{ вк};$$

$$a = 0,818 \text{ вк};$$

2) вычисление знаменателя

$$\begin{aligned} \text{пробел } & at^2 x - a^3 = \\ & = p \end{aligned}$$

$$\begin{aligned} \text{пробел } & \sqrt[p]{} = \\ & = p \end{aligned}$$

3) вычисление числителя

$$\begin{aligned} a - x^3 & = \\ & = b \end{aligned}$$

$$x^3 - \text{пробел } 5b =$$

4) вычисление выражения

$$/p = .$$

Перед вводом каждого символа необходимо соблюдать правила регистров.

Универсальный режим. В этом режиме информация вводится в машину двумя способами.

Ввод информации с машинки. На панели сигнализации должна быть нажата клавиша "Вывод на печать". Однократным нажатием на клавишу "Пуск I" заставить машинку отработать (машинка должна осуществить возврат каретки и перевод строки), после чего можно вводить информацию.

В зависимости от вида вводимой информации на машинке набирается:

Mt - для целых чисел;

Mp - для чисел с плавающей запятой;

Mch - для чисел с фиксированной запятой;

Md - для длинных чисел;

MK - для команд.

Здесь M - начальный адрес вводимого массива.

Каждому виду массива должна предшествовать буква " m ". Например, необходимо ввести в машину следующую информацию:

массив чисел с плавающей запятой начиная с ячейки 70;

массив команд начиная с 80-й ячейки;

дать команду на исполнение программы.

Ввод осуществляется так: нажать клавишу "Вывод на печать", "Пуск I" и набрать на машине:

70п (возврат каретки осуществляется автоматически);

3,75 вк

0,814 вк

7,102 вк

6,03 вк

} ввод массива чисел;

ш (возврат каретки осуществляется автоматически);

80к

п70н 74 вк

уп71н 74 вк

сп74н 74 вк

дп72н 73 вк

бл73н 74 вк

пп73н 6 вк

02270н вк

пп74н 6 вк

02274н вк

к0н вк

ш

80и - команда на исполнение программы.

Примечание. При вводе какого-либо символа необходимо ввести вначале регистр этого символа, если он отличается от регистра предыдущего символа. Например, перед вводом команды сп74н 74 необходимо набрать нижний регистр (НР), так как последний символ в предыдущей команде был в верхнем регистре (ВР), а символ с в НР. После напечатания символа с необходимо нажать ВР, поскольку символ п в верхнем регистре.

Исправление ошибок, допущенных при вводе информации или при составлении программы, осуществляется следующим образом: нажимается клавиша "Пуск I"; печатается номер ячейки, содержимое которой необходимо исправить, и признак информации (т, з, п, ф или к); печатается вводимая информация. Если ошибка замечена при вводе, то необходимо, не давая возврата каретки, напечатать символ *, являющийся признаком ошибки, и снова напечатать данную команду или число.

Ввод информации с фототрансмиттера:

1) нажать на панели пишущей машинки клавишу "Вкл.трансмиттера";

2) заправить перфоационную ленту в FS-I500; при заправке ленту необходимо вставить узкой частью наружу (лента делится на две части синхродорожкой);

3) на панели сигнализации нажать клавишу "Ввод с трансмиттера" и "Пуск I".

Автономный режим. В автономном режиме можно осуществить набивку информации на перфоленту без связи с ЭВМ, которая в это время производит вычисление по введенной в нее ранее программе, без вывода на печать.

Для перехода в автономный режим необходимо: нажать на ПС клавишу "Связь - автономный"; нажать клавишу на ПС "Печать - перфорация"; нажать клавишу "Работа перфоратора" на панели печатающей машинки.

Перед пробивкой первого символа необходимо несколько раз (6 - 8) пробить признак ВР. Это делается для удобства ввода перфоленты с FS-I500.

В автономном режиме можно также осуществлять проверку отперфорированной ленты путем вывода на печать пробитых на ней символов. Для этого необходимо нажать клавишу "Связь - автономный"; включить трансмиттер; заправить ленту; нажать клавишу "Печать с трансмиттера".

Режим "Выдача памяти". Для перехода в этот режим на ПС должны быть нажаты клавиши "Вывод на печать" и "Выдача памяти". После этого машинка отпечатает признак режима и содержимое СЧК, например,

bb

157.

После этого необходимо напечатать MiN в случае вывода на печать или MiNpk в случае вывода на перфорацию и дать возврат каретки.

Здесь M - начальный адрес массива ячеек, содержимое которых необходимо вывести;

i - признак вида информации (т,п,з,ф,к);

N - количество ячеек, входящих в выводимый массив;
лк - признак вывода на перфорацию.

Пример: 1. Вывести на печать массив чисел с плавающей запятой из 30 ячеек начиная со 107-й. (На машинке печатается 107п 30 вк).

2. Вывести на перфоратор массив команд из 50 ячеек начиная с 253-й. (Печатается 253к 50лк, после чего на ПС нажимается клавиша "Вывод на перфорацию", отжимается клавиша "Вывод на печать" и дается возврат каретки).

Засылка кодов в регистры машины и "Запись - чтение" в ОЗУ.

Запись кодов в ОЗУ. Набрать на КНК нужный код. Нажать кнопку "Уст."0" СМ". Заслать набранный код в СМ путем нажатия кнопки "ПУ-->СМ". Набрать на КНК адрес ячейки ОЗУ. Очистить РгОЗУ нажатием кнопки "Уст."0" РгОЗУ". Заслать набранный адрес в РгОЗУ, нажав кнопку "ПУ-->РгОЗУ".

Нажать на ПУ клавиши "Чтение - запись" и "ОЗУ". Нажатием кнопки "Отсылка" код, находящийся в СМ, перепишется в ОЗУ по адресу, указанному в РгОЗУ.

Чтение из ОЗУ. Очистить СМ и РгОЗУ, как при записи кодов. Заслать в РгОЗУ нужный адрес, нажать клавишу "ОЗУ" и нажатием кнопки "Отсылка" прочитать в СМ код, хранящийся по данному адресу ОЗУ.

Засылка кодов в регистры осуществляется так же, как и засылка кода в РгОЗУ, необходимо только использовать соответствующие выбранному регистру кнопки "Уст."0" и "ПУ-->Рг".

Контрольные вопросы

1. Назначение, состав и основные технические данные ВУ.
2. Электрические схемы управления внешними устройствами.
3. Назначение и состав пульта управления.
4. Режимы работы машины.

Рабочее задание

1. Произвести вычисление алгебраического выражения в счетном режиме.
2. Отперфорировать на ленту заданную программу в автономном режиме.
3. Ввести программу с трансмиттера и решить задачу.
4. Вывести программу и результаты вычислений на печать в режиме "Выдача памяти".
5. Произвести засылку заданных кодов в регистры машины и в ОЗУ.

Образец рабочего задания

$$I. C = \frac{x^2 - e^y + xyz^5 - \sqrt[5]{y^x}}{\sin x + \ln z},$$

где $x = 2,03;$
 $y = 14,102;$ $z = 0,763.$

2. 275п 4,075 (вк)

6,025 (вк)

2,1 (вк)

5,5 (вк)

ш280к п 275 н 279 (вк)

сл 276 н 279 (вк)

ун 277 н 279 (вк)

дп 278 н 279 (вк)

о 2270 н (вк)

пп 279 н 7 (вк)

о 2274 н (вк)

к О н (вк)

ш 280и

3. Ввести программу с трансмиттера и решить задачу.
4. Вывести программу и результаты вычислений на печать в режиме "Выдача памяти".
5. Заслать код II0706 в РтК (IO); записать в 207-ю ячейку ОЗУ число I70334 и прочитать его в СМ; прочитать из 279-й ячейки ОЗУ содержимое в СМ.

Содержание отчета

1. Блок-схема ВУ и ее описание.
2. Результаты вычисления выражения, заданного преподавателем.

ЛАБОРАТОРНАЯ РАБОТА № 3

ДЛГОВРЕМЕННОЕ ЗАПОМИНАЮЩЕЕ УСТРОЙСТВО

§ I. Основные технические данные ДЗУ

ДЗУ предназначено для постоянного хранения микропрограмм, библиотеки подпрограмм, констант и т.д.

ДЗУ входит в состав главного шкафа машины.

Емкость ДЗУ: 14336 36-разрядных чисел и 2048 72-разрядных чисел (команд микропрограмм).

Время обращения к ДЗУ - 20 мксек.

Считывающий ток в выбранной числовой линейке I00 - I25 ма.

Параметры считанного (выходного) сигнала с ЯН:

полярность - отрицательная;

амплитуда - 5,5 - 8 в;

длительность - 3 - 4 мксек;

длительность переднего фронта - 0,8 - 1,3 мксек;

длительность заднего фронта - 0,8 - 1,2 мксек.

В ЯН1 и ЯН2 записаны микропрограммы, в ЯН3 - ЯН8 - подпрограммы, в ЯН-9 - запасная (без прошивки).

Основным функциональным узлом ДЗУ является накопитель, в котором в качестве элементов запоминания используются импульсные трансформаторы на оксиферовых сердечниках. Первичные одновитковые обмотки служат для записи информации. Применяется принцип запоминания на одном сердечнике одного разряда нескольких чисел (до 256). Запись (запоминание) "1" осуществляется пронизыванием первичным (адресным) проводом сердечника, запись "0" - обходом сердечника (рис. I). Вторичные обмотки сердечников служат для считывания информации по разрядам чисел.

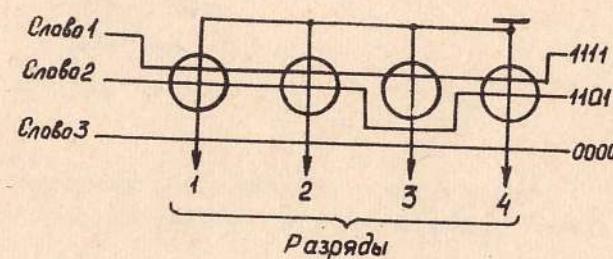


Рис.1. Принцип хранения информации в ДЗУ

§ 2. Блок-схема ДЗУ (рис.2)

Выбор числа из ДЗУ осуществляется путем подачи адреса данного числа в РгДЗУ и соответствующих импульсов обращения (из ЦУУ).

Согласно расшифрованному адресу с помощью триодных переключателей выбирается нужное число.

Усилители считывания служат для усиления и формирования выходных импульсов с накопителя; МУУ - для формирования считывающих и стробирующих сигналов; блок задержек - для временной задержки сигналов 62 - 72-го разрядов ячейки накопителя на время выполнения микрокоманды и подачи их на входы первых 11 разрядов РгДЗУ как адреса следующей микрокоманды.

§ 3. Описание функциональной схемы

Накопитель ДЗУ на 16384 адреса выполнен на 9 сменимых ячейках накопителя (ЯН). Каждая ЯН представляет диодно-трансформаторную матрицу. На схеме рис.3 показано размещение оксидеровых сердечников на одной плате ЯН: 8 горизонтальных рядов сердечников, по 36 оксидеров в каждом ряду, что определяет разрядность запоминаемых чисел.

Каждый адресный провод последовательно проходит через 8 рядов одной ЯН, т.е. запоминает код восьми 36-разрядных чисел.

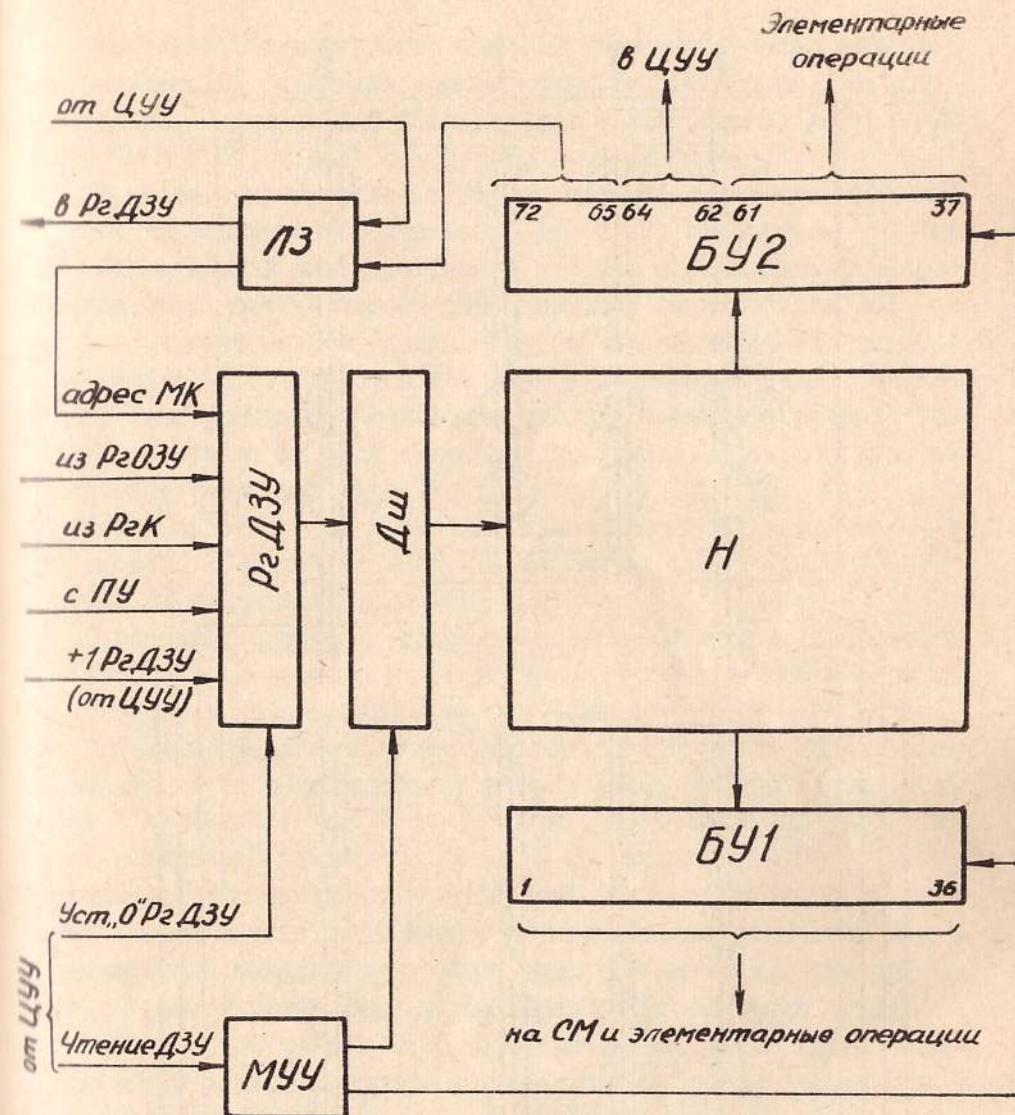


Рис.2. Блок-схема ДЗУ:

РгДЗУ - 14-разрядный регистр адреса;

Дш - дешифратор;

Н - накопитель;

БУ1, БУ2 - блоки усилителей считывания;

ЛЗ - линии задержек;

МУУ - местное устройство управления

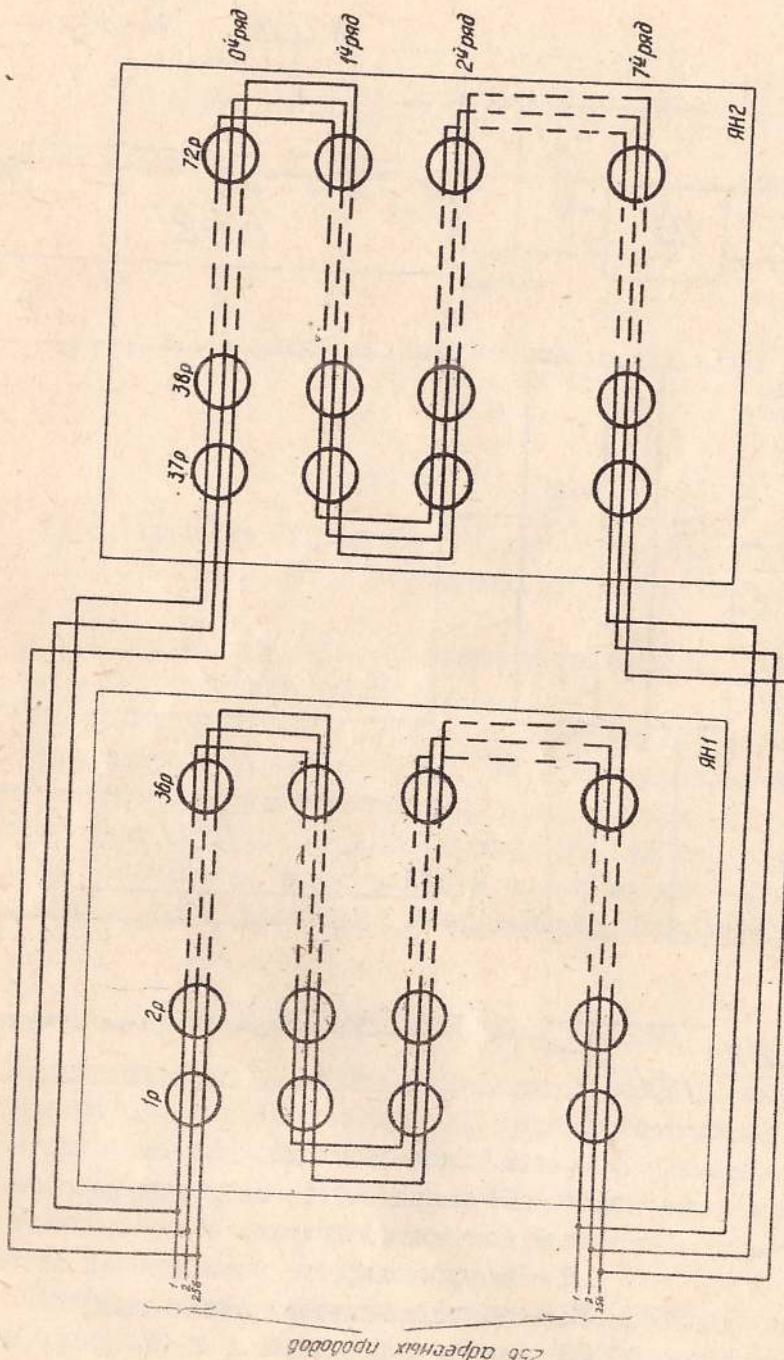


Рис.3. Принцип образования 72-разрядных кодов

Максимальное количество адресных проводов, проходящих через оксидеровый сердечник данного ряда, взято равным 256, следовательно, максимальный объем хранимой информации каждой ЯН равен (8×256), т.е. 2048 36-разрядным числам.

В связи с применением в машине микропрограммного принципа управления возникла необходимость увеличить разрядность части ДЗУ, запоминающей микропрограмму, так как в коде каждой микрокоманды указываются элементарные операции (а их больше 36) и адрес следующей микрокоманды. Информация с двух ячеек накопителя (ЯН1 и ЯН2), служащих для хранения микропрограмм, считывается одновременно, и считанные сигналы с каждого разряда каждой ЯН поступают на свой разрядный усилитель. Этим достигается увеличение разрядности до 72 (рис.4).

Таким образом, при 9 ячейках накопителя общая емкость ДЗУ составляет: $2048 \times 7 = 14336$ 36-разрядных чисел (ЯН3 – ЯН9) и 2048 72-разрядных чисел (ЯН1 – ЯН2).

В пределах одной ЯН однополярные концы выходных обмоток трансформаторов каждого из 36 вертикальных рядов объединяются через диоды в общую разрядную шину (рис.4).

Одноименные разряды ячеек накопителя ЯН2 – ЯН9 объединяются вместе через развязывающие диоды и поступают на входы усилителей считывания I – 36-го разрядов, а 36 разрядов ЯН1 – на входы усилителей считывания 37 – 72-го разрядов. Другие (потенциальные) концы выходных обмоток каждого горизонтального ряда объединяются вместе. Одноименные горизонтальные ряды всех 9 ячеек накопителя объединены в общие шины, каждая из которых связана с соответствующим выходом дешифратора выбора ряда (ДШI).

256 проводов каждой ЯН (рис.5) разделены на 16 групп по 16 проводов в каждой группе. Провода каждой группы через разделительные диоды (см. Д₁ – Д₁₆ для I-й группы, Д₁₇ – Д₃₂ для 2-й группы любой ЯН) объединены в общую шину. 16 таких шин ЯН1 соединены с соответствующими выходами триодного переключателя импульсов (ШПI для I-й шины, ..., ШП16 для 16-й шины), на общий вход которого с усилителя мощности поступает считающий сигнал "2 Чтение ДЗУ" отрицательной полярности.

16 шин, объединяющих через разделительные диоды по 16 проводов остальных 8 ЯН (ЯН2 – ЯН9), выведены на общие шины. 16 таких шин подведены к соответствующим выходам три-

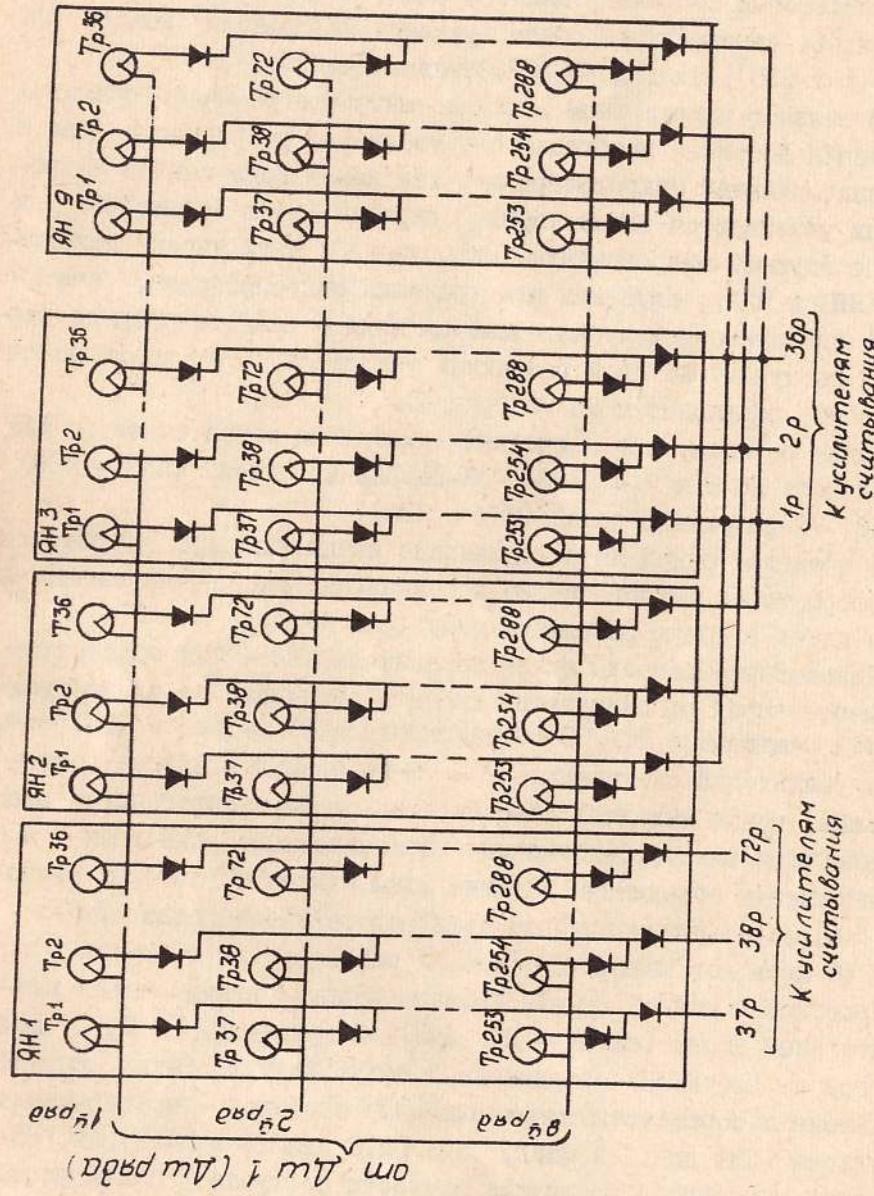


Рис.4. Выходная дешифрация

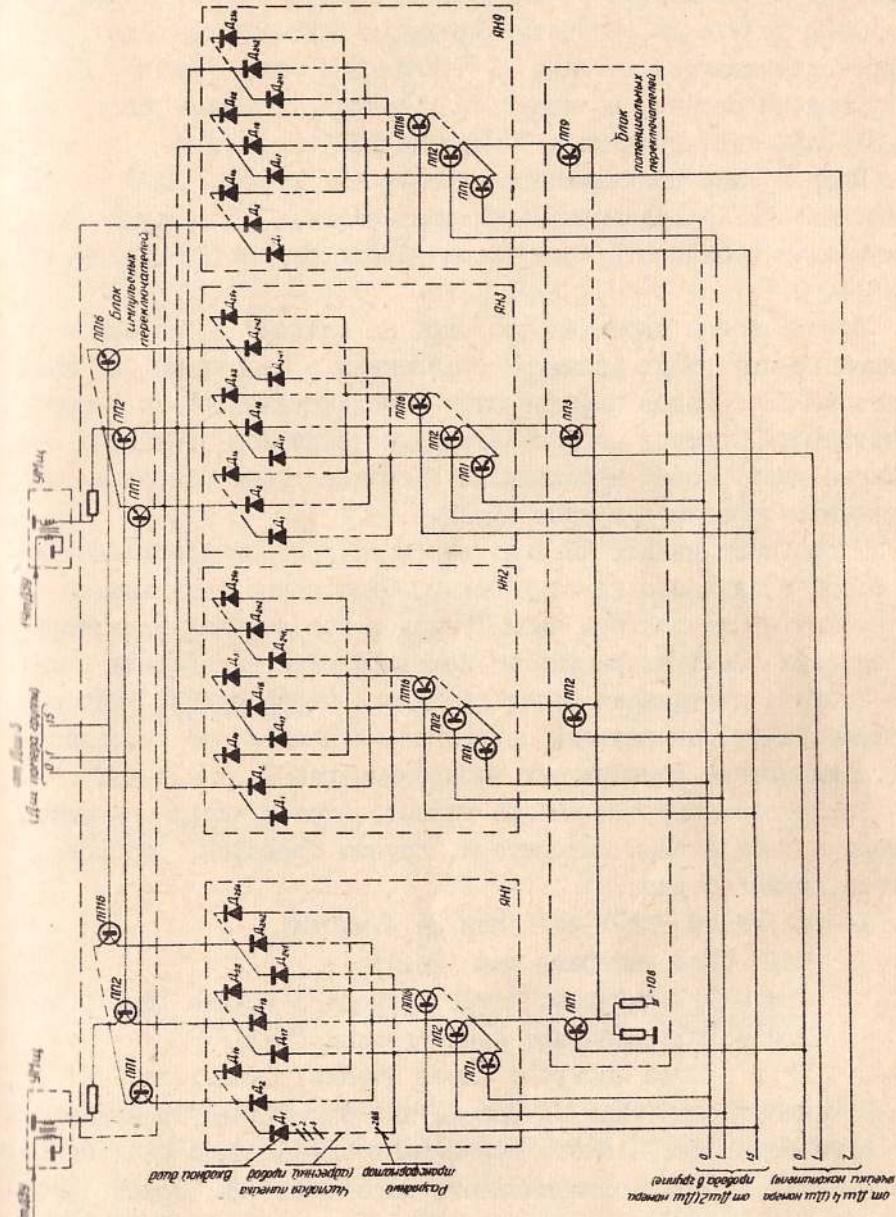


Рис.5. Входная дешифрация

одного переключателя импульсов (на схеме рис.5 правый блок импульсных переключателей). На общий вход триодов переключателя импульсов поступает считающий импульс ("I Чтение ДЗУ") с ячейки усилителя мощности. Ячейка усилителя мощности формирует считающие сигналы "I Чтение ДЗУ" и "2 Чтение ДЗУ" при наличии сигнала на импульсных входах, который поступает из МУУ при подаче сигнала "Чтение ДЗУ".

Прохождение считывающего сигнала ("1 Чтение ДЗУ" или "2 Чтение ДЗУ") обеспечивается разрешающими (нулевыми) потенциалами соответствующих выходов дешифратора выбора группы (Дш3).

Другие концы адресных проводов на каждой ЯН, по одному с каждой группы, в пределах ЯН объединены и подведены к коллекторам 16 триодов переключателей потенциалов. Базы соответствующих триодов переключателей потенциалов всех 9 ЯН соединены между собой и подведены к выходам дешифратора выбора адресного провода в группе (Д2).

В пределах каждой ЯН эмиттеры всех триодов соединены между собой и выведены на общую шину. Эмиттерные шины каждой ЯН поступают на коллекторы своих триодов согласующих инверторов на выходах дешифратора выбора номера ячейки накопителя (Л1).

Импульс считывания будет проходить через ту ЯН, которой соответствует разрешающий потенциал на коллекторе согласующего инвертора дешифратора выбора номера ЯН.

Таким образом, для выбора нужного адреса числа задаются номера нужной ячейки накопителя, группы проводов, провода в группе, нужного ряда.

14-разрядный РГДЗУ разделен на 4 части:

ТГ1 - ТГ3 выбирают ряд (ДИТ):

Тг4 - Тг7 выбирают номер провода в группе (и-2)

Tr8 - TrII выбирают номер группы (Пт3).

Тр12 - Тр14 выбирают номер ячейки, находящейся

В каждом конкретном случае сигнал считывания проходит через один триод импульсного переключателя, входной диод, числовую линейку, триод потенциального переключателя, триод выбора ячейки накопителя и через делитель на землю (рис.5,6). При этом возбуждаются все пронизанные адресным проводом оксиферовые сердечники во всех 8 рядах выбранной ЯН.

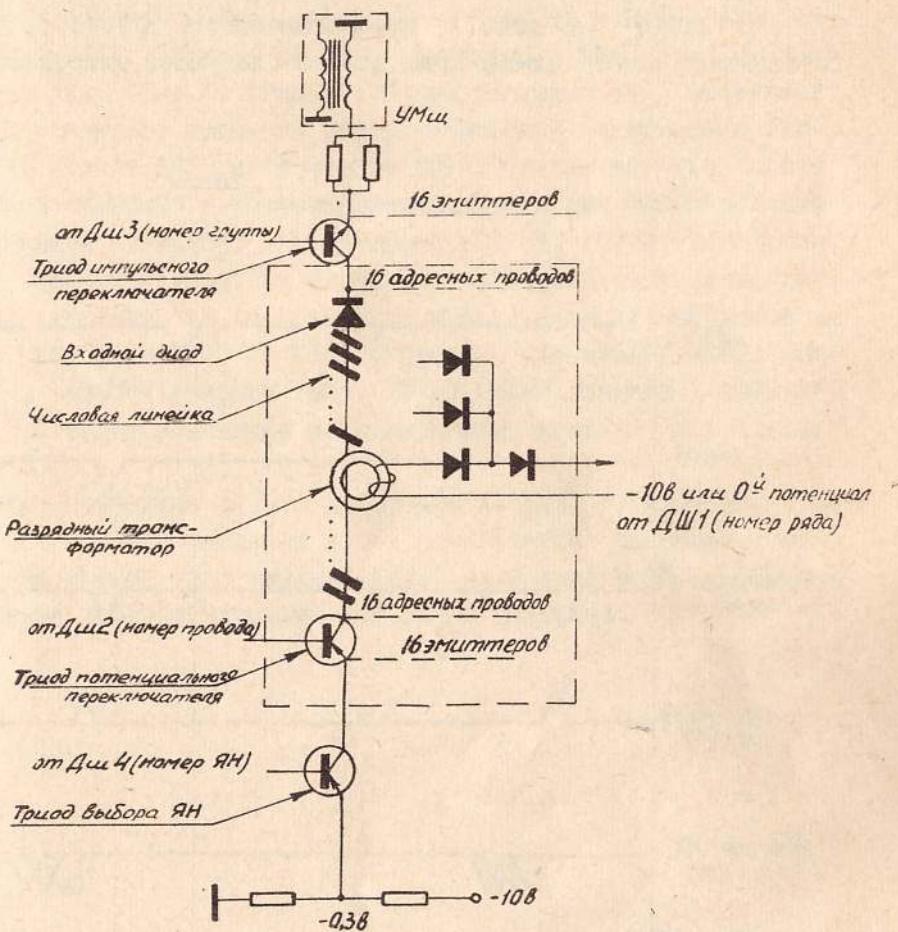


Рис.6. Схема прохождения сигнала считывания

Подачей нулевого потенциала на потенциальные концы трансформаторов (рис.6) дешифратор выбора ряда выбирает один из 8 рядов ЯН. На потенциальные концы остальных 7 рядов подается запирающий потенциал - 10в. Код считанного числа через развязывающие диоды поступает на входы усилителей считывания.

§ 4. Цикл обращения к ДЗУ

Цикл обращения к ДЗУ (при поступлении на РгДЗУ адреса следующей микрокоманды из ДЗУ) начинается с подачи команды "Уст."0" РгДЗУ" (см.рис.7), вырабатываемой в ЦУУ, которая подготовливает его к приему кода адреса следующей микрокоманды.

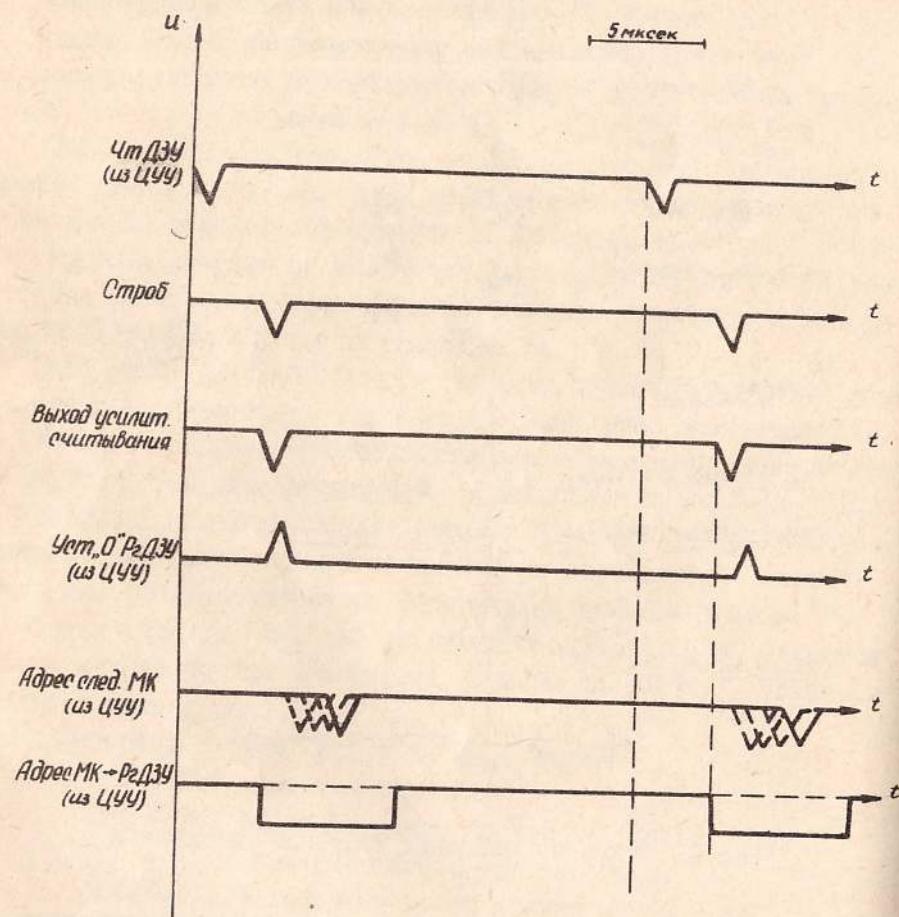


Рис.7. Временная диаграмма работы ДЗУ

Через 3,1 мкsec после "Уст."0" РгДЗУ" на импульсные входы клапанов приема адреса поступает код адреса следующей микрокоманды (из блока задержек). Прохождение их на входы "Уст."1" триггеров обеспечивается потенциалом "Адр.МК в ДЗУ", вырабатываемым в ЦУУ. Сигнал "Чт.ДЗУ" поступает в ДЗУ через 16 мкsec после команды "Уст."0" РгДЗУ". После усиления он поступает на вход усилителя мощности для формирования считывающих сигналов "1 Чтение ДЗУ" и "2 Чтение ДЗУ". Тот же импульс после временной задержки и усиления используется для формирования стробирующих сигналов. Код считанного с ЯН1 (37 - 72-го разрядов) числа поступает на входы 36 усилителей считывания, где сигналы стробируются во времени и после усиления поступают на другие узлы машины: 37 - 61-й разряды - как элементарные операции, а старшие разряды (62 - 72-й) после временной задержки на 3,6 мкsec как адрес следующей микрокоманды в регистр адреса ДЗУ.

Коды, считанные с ЯН2, поступают на входы усилителей считывания I - 36-го разрядов и, как элементарные операции, передаются на другие узлы машины. Коды, считанные с ЯН3 - ЯН9, поступают на входы усилителей I - 36-го разрядов.

Контрольные вопросы

1. Назначение и состав ДЗУ ЭПВМ "Наири".
2. Технические данные ДЗУ.
3. Принцип построения ДЗУ.
4. Конструкция накопителя ДЗУ.
5. Принцип входной дешифрации.
6. Принцип выходной дешифрации.
7. Цепь прохождения сигнала считывания.
8. Временная диаграмма работы ДЗУ.

Рабочее задание

1. По заданному адресу ДЗУ определить номера ячейки, группы, провода в группе и номер ряда.
2. Найти заданный провод в ячейке.
3. Зациклить чтение заданного адреса ДЗУ. Просмотром разрядов ДЗУ (выводы усилителей считывания), используя систему команд, определить машинную операцию, прошитую по данному адресу ДЗУ.

Содержание отчета

1. Блок-схема ДЗУ и ее описание.
2. Расшифровка заданного преподавателем адреса ДЗУ.
3. Структура ячейки ДЗУ, осуществляющей зацикленную микрокоманду.
4. Осциллограммы импульсов на выходах усилителей считывания ДЗУ.

Часть вторая ПРОГРАММИРОВАНИЕ

Программирование на ЭВМ "Наири-С" обладает некоторой спецификой. Это обусловлено наличием в машине нескольких режимов работы и достаточно высокой степенью автоматизации программирования. В этой машине впервые в отечественной вычислительной технике была применена программирующая программа, жестко записанная в ДЗУ (режим автоматического программирования). Она дала возможность работать на машине лицам, не имеющим специальной подготовки по методам вычислений.

I. СЧЕТНЫЙ РЕЖИМ

В счетном режиме имеется возможность использовать машину для вычисления алгебраических выражений, если известны конкретные числовые значения всех параметров, входящих в них.

Для перевода машины в счетный режим нужно нажать клавиши "Вывод на печать" и "Универсальный - счетный" на панели сигнализации, после чего нажать клавишу "Пуск I", и печатающая машинка отпечатает сочетание *CC* (условный шифр счетного режима), а на панели сигнализации загорится лампочка "Команда ввод". Для работы печатающей машинки нужно нажать клавишу "Вкл. печати" на пульте управления внешними устройствами, а тумблер выключения машинки поставить в положение "I". Ввод формул осуществляется с печатающей машинки с одновременной печатью их на бумажной ленте. Формулы должны быть записаны на входном алгоритмическом языке счетного режима, который близок к обычно-му математическому языку формул и содержит:

- а) знаки арифметических действий +, -, х, / ;
- б) знак запятой ",";

- в) цифры 0, 1, ..., 9;
- г) целые положительные показатели 0, 1, ..., 9;
- д) буквы *c, b, y, d, l, m, z, n, x, a, b, t, ch, u, o, e, l, p, m, w, s, t, n, zh, z, y, b, f, c, zh, f, ya, ju, z, i, j, k, n*, которым можно придавать те или иные численные значения;
- е) знак факториала "!";
- ж) число $\pi = 3,141592651$;
- з) знак равенства "=";
- и) символы элементарных функций $\sin, \cos, \tg, \arcsin, \operatorname{acos}, \arctg, \exp, \ln, \lg, \sqrt{ }, \sqrt[n]{ }$;
- к) специальные символы пробела и возврата каретки.

С помощью этого алгоритмического языка можно в счетном режиме производить следующие вычисления:

- 1) вычислить значения элементарных функций $\sin, \cos, \tg, \arcsin, \operatorname{acos}, \arctg, \exp, \ln, \lg, \sqrt{ }, \sqrt[n]{ } \quad (\text{где } n > 2)$;
- 2) по заданной градусной мере угла вычислить его радианную меру и наоборот;
- 3) присвоить буквам любые численные значения, а также результат предыдущего действия;
- 4) выдать на печать значение любой буквы;
- 5) вычислить факториал для чисел от 1 до 12.

Для выполнения этих вычислений необходимо соблюдать некоторые правила:

1. При вводе всем функциям, включая $\sqrt{ }, \sqrt[n]{ }$ и переход от градусной меры к радианной, предшествует пробел, например:

пробел $\cos x =$

пробел $\sqrt{16} =$

пробел $30^\circ =$

2. Для перехода от радианной меры к градусной печатается при вводе число и знак равенства, например, $5,643 =$. После знака равенства машинка отпечатает это число в градусах.

3. Для присвоения букве численного значения набираются

буква

знак =

число

возврат каретки

Пример: $a = 5$ (вк)

$b = 4$ и т.д.

Одно и то же значение можно присвоить нескольким буквам одновременно, тогда для экономии времени следует печатать

$a = 5$ (вк)

$= b$ (возврат каретки осуществляется машиной автоматически)

$= c$ и т.д.

4. Для присвоения букве результата предыдущего действия печатается

знак =

выбранная буква.

Пример: $ax^2 + bx + c =$ (машина печатает результат и делает возврат каретки с переводом строки). Далее нужно напечатать $= c$, после чего машина присваивает результат букве c .

5. Можно запросить у машины значение любой буквы. Для этого печатается

буква

знак =

пробел.

6. Чтобы вычислить $\sqrt[n]{ }$ из какого-либо выражения, печатается

пробел

показатель n

$\sqrt{ }$

выражение

знак равенства.

В подкоренных выражениях не разрешается использовать никаких символов арифметических действий, кроме произведения букв.

Пример: пробел $\sqrt[4]{46} =$ (машина печатает ответ).

7. Показатель, относящийся к функции, печатается непосредственно после символа функции, т.е. $(\sin x)^2$ нужно вводить так:

пробел

\sin

показатель 2

аргумент x

знак равенства =

8. Тригонометрическая функция от угла в градусной мере печатается следующим образом:

пробел
функция
пробел
угол в градусах

Пример: $\sin 30^0$
пробел
 \sin
пробел
 $30.$

Сложную функцию от угла в градусной мере вычислить нельзя, т.е. нельзя, например, вычислить $\sin \cos 50^0$.

9. Если выражение начинается знаком "-", то вначале надо напечатать 0, а затем само выражение.

Для перехода из счетного режима в универсальный нужно отжать клавишу "Универсальный - счетный" на режимной панели и нажать "Пуск I". После этого машина будет готова для вычислений в универсальном режиме.

II. УНИВЕРСАЛЬНЫЙ РЕЖИМ

В универсальном режиме машина работает по программе, введенной в ее память. Программа может быть составлена с использованием автоматического программирования (*ап*) в специальном алгоритмическом языке (в этом случае программа в машинных кодах составляется самой машиной, без участия программиста) и с использованием системы команд машины. Возможно составление и комбинированной программы, т.е. программы, включающей в себя как элементы автоматического программирования, так и систему команд. Ряд типовых задач можно решать по стандартным программам, жестко закоммутированным в машине.

§ I. Автоматическое программирование

Для того, чтобы машина работала в универсальном режиме, нужно нажать на режимной панели клавишу "Вывод на печать", включить печатающую машинку нажатием клавиши "Вкл.печати" и

установкой тумблера включения в положение "I", нажать клавишу "Пуск I". После этого на режимной панели загорается лампочка "Команда ввод"; это означает, что машина готова для вычислений в универсальном режиме. Чтобы машина считала в режиме *ап*, нужно перед вводом информации на печатающей машине набрать сочетание букв *ап* (признак обращения к программирующей программе).

Для составления программ в *ап* используется специальный алгоритмический язык, который содержит:

а) буквы $c, b, y, d, l, m, z, n, x, a, b, c, m, o, e, \ell, r, t, s, \pi, w$, которые используются для обозначения переменных;

б) знак запятой "," ;

в) цифры 0, 1, ..., 9, из которых можно образовать любое число $Q (|Q| < 10^{19})$ и любой положительный показатель;

г) число $\pi = 3,141592651$;

д) знаки арифметических действий +, -, / (знак умножения опускается);

е) знаки соотношений =, ≠, <, <=, >, ≥;

ж) символы элементарных функций $\sin, \cos, \tg, \arcsin, \arccos, \arctg, \exp, \ln, \lg, \sqrt{\cdot}$;

з) скобки ();

и) знак модуля || ;

к) буквы i, j, k только для обозначения индексов и целочисленных переменных;

л) букву p для обозначения целочисленных переменных;

м) 17 операторов;

н) специальные символы возврата каретки и пробела.

Особого объяснения в алгоритмическом языке *ап* требуют операторы.

Оператор - это некоторое указание, написанное в виде слова или группы слов, данное вычислительной машине для обработки информации, поступающей вслед за этим указанием.

Операторы программируются в любой последовательности, какую потребует решаемая задача, а нумеруются при этом строго по порядку. В *ап* используются 17 операторов, которые условно можно разбить на несколько групп.

Вычислительные операторы.

1. Вычислим $x = X$ - для организации вычислений по формулам, составленным из символов алгоритмического языка АП.

Пример: 1. вычислим $y = (ac^2 + \sin x)/x$.

2. Допустим $x = y$ - для присвоения переменным конкретных числовых значений, а также значений других переменных и их модулей.

Пример: 1. допустим $x = 10$
2. допустим $a = 0,2378$
3. допустим $r = -|y|$.

Знак " - " перед переменной без модуля в записи данного оператора употреблять нельзя. Буквам i, j, k, p разрешается присваивать только целочисленные значения.

3. Вставим - служит для изменения переменных без индексов на заданную величину a - положительную или отрицательную, причем значения переменных i, j, k, p можно изменять только на целое число.

Пример: 1. вставим $x = x + 0,5$
2. вставим $K = K - 1$.

4. Введем - для ввода в машину значений переменных в процессе вычислений.

Пример: 1. введем a .

Если машина встречает такой оператор, то вычисления прерываются, на режимной панели загорается лампочка "Команда ввод". Далее на печатающей машинке нужно набрать численное значение, соответствующее a , и возврат каретки, после чего машина продолжает вычисления. С помощью этого оператора нельзя вводить значения букв i, j, k, p .

В записи рассмотренных операторов может быть сколько угодно параметров, которые отделяются друг от друга пробелами. Исключение составляет оператор "Вычислим"; в его записи должно быть не более одной формулы.

Пример: 1. введем a пробел b пробел c
2. вставим $y = y + 3,5$ пробел $i = i + 1$
3. вычислим $y = x^3 + 8x$.

Операторы печати.

1. Печатаем - для печати значений букв с заданным числом знаков после запятой.

Пример: 1. печатаем с N знаками $x y z$.

В записи этого оператора не рекомендуется использовать более трех переменных ввиду ограниченной ширины бумажной ленты на печатающей машинке. Количество знаков $N < 9$.

2. Интервал - для пропуска указанного количества строк при печати.

Пример: 1. интервал K .

Встречая такой оператор, машина переводит каретку печатающей машинки на K строк.

3. Спросим - дает возможность контролировать значения переменных в процессе вычислений. Оператор "Спросим" работает только при нажатой клавише "Вариант" на режимной панели. Встретив оператор "Спросим" при нажатой клавише "Вариант", машина останавливается, и на режимной панели загорается лампочка "Команда ввод". После этого нужно на печатающей машинке набрать интересующую переменную и знак " = ", машина автоматически печатает численное значение этой переменной. Если переменная индексированная, то ее нужно набирать с цифровым индексом. У двухиндексной переменной нужно делать пробел между индексами. Для пуска машины после вывода численных значений запрошенных переменных нужно отжать клавишу "Вариант" и сделать возврат каретки.

Операторы перехода. Программа решения какой-либо задачи в АП представляет собой последовательность операторов. Если операторы выполняются в той же последовательности, в какой они пронумерованы, то такой порядок выполнения операторов

называется естественным. Для прерывания естественного порядка выполнения операторов служат операторы перехода.

1. Идти к N - для безусловной передачи управления. Встретив этот оператор, машина переходит к исполнению оператора, имеющего номер N .

2. Если $\alpha \beta Z$ идти к N - оператор условной передачи управления, где $\alpha \beta$ - есть какое-либо условие (α, β могут быть любыми буквами, Z кроме того, может быть любым числом, а Z - один из знаков соотношений $=, \neq, >, <, >, \leq$). В случае, если условие $\alpha \beta Z$ выполняется, то машина переходит к выполнению оператора с номером N , в противном случае выполняется следующий по порядку оператор.

Служебные операторы.

1. Кончаем - указывает на конец задачи. После этого оператора можно лишь начать решение задачи сначала.

2. Останов - для останова вычислений. Продолжать вычисления можно после нажатия клавиши "Пуск 2" на режимной панели.

3. Исполним - для передачи управления на начало программы. Этот оператор стоит в конце любой программы и употребляется без номера.

Операторы печати графиков.

1. Храним - означает, что машина должна запомнить заданное количество значений какой-либо переменной.

Пример: 1. храним 12 у
2. храним 5 х.

2. Начертим - с помощью этого оператора машина вычерчивает на бумажной ленте печатающей машинки графики функций. Одновременно можно вычерчивать графики двух функций.

Пример: начертим 1 гр.
начертим 2 гр.

Перед оператором "Начертим" всегда должен работать оператор "Храним", для того чтобы накопить массив значений, по которым затем строится график.

Операторы подключения программ.

1. Решим - предусмотрен только для двух задач:

1) решения системы линейных алгебраических уравнений порядка $n \leq 28$. В этом случае оператор задается в следующем виде: решим $S_{10}x = 0$, где S - признак того, что решается система линейных алгебраических уравнений, n - порядок системы, a - буква, обозначающая коэффициенты системы, x - неизвестное, $= 0$ машина добавляет автоматически.

Пример: 1. решим $S_{10}x = 0$. Это означает, что требуется решить систему линейных алгебраических уравнений 10-го порядка с коэффициентами c_{ij} ($i, j = 0, 1, \dots, 9$);

2) решения алгебраических уравнений степени $m \leq 38$. В этом случае оператор задается в виде: решим $ra z^m = 0$, где r означает признак решения алгебраических уравнений, a - коэффициенты уравнения, z - неизвестное, m - степень уравнения.

После оператора "Решим" продолжать вычисления в *ап* нельзя. Поэтому оператор "Решим", как правило, стоит в конце операторной программы. Оператор "Кончаем" в этом случае употреблять не нужно.

2. Программа. В логических задачах обычно встречаются операции специальной арифметики (логический и арифметический сдвиги, поразрядное сложение и т.д.), программу для которых нельзя записать в символах *ап*. Однако эти операции могут быть запрограммированы командным путем, т.е. составлением небольших подпрограмм, реализующих эти операции. Обращение к этим подпрограммам из *ап* ведется с помощью оператора "Программа". Таким образом, оператор "Программа n " означает переход к нестандартной программе, начинающейся с ячейки n . Выход в *ап* из подпрограммы осуществляется по 66-й ячейке, т.е. в конце нестандартной программы должна стоять команда *и 66 н*.

Использование оператора "Программа" бывает затруднено, поскольку программирующая программа *ап* распределяет память машины по отдельным рабочим массивам автоматически для каждой задачи. Поэтому прежде чем поместить в память машины нестандартную программу, нужно рассчитать количество ячеек, отведенных программой под рабочие массивы. Нестандартную программу можно разместить только в свободном участке памяти так, чтобы не портить рабочие массивы программы *ап*. Распределение памяти на рабочие массивы программой *ап* дано в приложении 2.

Специальные операторы.

Массив - служебный оператор, контролирует размещение программы данной задачи в оперативной памяти машины. Оператор "Массив" употребляется без номера и бывает двух видов: "Массив *K*" - выдает на печать количество составленных *ап* команд до момента ввода оператора; "Массив *B*" - печатает в командах операторную программу, составленную в кодах *ап*. Номера команд не указываются, так как начало этой программы строго фиксировано адресом 370.

Правила использования языка *ап*. При записи задачи на языке *ап* следует соблюдать некоторые правила:

1. Особенностью программирования в *ап* является использование так называемой шапки программы. Шапка - это некоторая информация, относящаяся к решению задачи, которая вводится без порядкового номера до решения задачи. В шапке обязательно следует указать индексы и переменные, к которым они относятся, если в решении задачи используются переменные с индексами, а также максимальные значения этих индексов. Кроме того, в шапке можно записать одну из формул (лучше длинную), которая будет вычисляться в программе. В простых задачах шапку можно не использовать.

2. Переменные в решении задачи могут сопровождаться одним или двумя индексами. Причем, если у переменных *a* и *b* имеется один общий индекс, то в шапке должно быть записано: *i=5ab*; это значит, в задаче используются переменные

$a_0 \ a_1 \dots a_5 \ b_0 \ b_1 \dots b_5$.

Если у переменных разные индексы, т.е. в шапке будет записано *i=10 j=15 x y*, то в задаче используются переменные

$x_{0,0} \ x_{0,1} \dots x_{0,15} \ x_{10} \dots x_{1,15} \dots x_{10,0} \ x_{15}$

$y_{0,0} \ y_{0,1} \dots y_{0,15} \ y_{1,0} \dots y_{10,0} \ y_{15}$.

Индексированные буквы в данной задаче уже не могут употребляться без индексов.

3. Аргументом функции является все то, что следует за символом функции до знака "+", "-" или "Возврат каретки", если аргумент не заключен в скобки.

4. Подкоренное выражение нужно заключать в скобки, если в него входят знаки арифметических действий "+", "-".

Примеры:

I. Вычислить выражение $y = \frac{4x^2}{\sqrt{1+x^3}}$, в интервале $[-1,1]$ с шагом 0,1.

Согласно правилам записи решения задачи на языке *ап* программа решения будет следующей:

ап

1. допустим $x = -1$ (вк)
 2. вычислим $y = 4x^2 / (\sqrt{1+x^3})$ (вк)
 3. печатаем с 9 знаками y (вк)
 4. вставим $x = x + 0,1$ (вк)
 5. если $x < 1$ идти к 2 (вк)
 6. кончаем
- исполним I.

Для решения этого примера можно использовать также шапку. Программа с использованием шапки такова:

ап

$$y = 4x^2 / (\sqrt{1+x^3})$$

1. допустим $x = -1$ (вк)
 2. вычислим y (вк)
 3. печатаем с 9 знаками y (вк)
 4. вставим $x = x + 0,1$ (вк)
 5. если $x < 1$ идти к 2 (вк)
 6. кончаем
- исполним I.

После каждой строки нужно делать возврат каретки. Исключение составляют операторы "Кончаем", "Останов", "Начертим", которые выполняют возврат каретки автоматически.

2. Построить график функции $y = x + e^x \sin \pi x$ в интервале $[2, 4]$ с шагом 0,05.

ап

1. допустим $x = 2$ (вк)
 2. вычислим $y = x + (\exp x) \sin \pi x$ (вк)
 3. печатаем с 9 знаками y (вк)
 4. храним 4I y (вк)
 5. вставим $x = x + 0,05$ (вк)
 6. если $x < 4$ идти к 2 (вк)
 7. начертим I гр
 8. кончаем
- исполним 1.

3. Вычислить значения функции y в интервале $[0, 2]$ с шагом 0,1

$$y = \begin{cases} \sin x & 1 > x > 0 \\ e^x & 1 < x \leq 2 \end{cases}$$

При $x = 1$ вычислить и отпечатать выражение $a = 1 - \sin x$.

ап

1. допустим $x = 0$ (вк)
 2. вычислим $y = \sin x$ (вк)
 3. вставим $x = x + 0,1$ (вк)
 4. если $x < 1$ идти к 2 (вк)
 5. вычислим $a = 1 - y$ (вк)
 6. печатаем с 9 знаками a (вк)
 7. вычислим $y = e^x$ (вк)
 8. вставим $x = x + 0,1$ (вк)
 9. если $x < 2$ идти к 7 (вк)
10. кончаем
- исполним 1.

Ввод информации. Ввод операторной программы в машину осуществляется двумя путями: с печатающей машинки и с перфоленты.

При вводе с печатающей машинки нужно набрать сочетание букв *ап*, затем вводить шапку, если она есть, далее – программу. Названия операторов подобраны так, что первые две

буквы определяют весь оператор. При вводе печатается сначала порядковый номер оператора, затем пробел, затем две буквы, определяющие оператор, остальные буквы печатающая машинка добавляет автоматически. Нужно также помнить, что каждой элементарной функции (за исключением квадратного корня) при вводе предшествует пробел.

Программа, пробитая на перфоленте, вводится через трансмиттер. Для этого перед нажатием клавиши "Пуск I" нужно нажать клавишу "Ввод с трансмиттера" на панели сигнализации и "Вкл. трансмиттера" на пульте управления печатающей машинкой.

Перфорация ленты осуществляется с помощью перфоратора в автономном режиме, при этом нужно придерживаться следующих правил:

1. Для каждого оператора перфорируются: его порядковый номер, пробел, первые две буквы оператора, информация, идущая после названия оператора, возврат каретки.

2. После пробивки признака *ап*, а также после операторов "Кончаем", "Останов" и "Начертим" возврат каретки не перфорируется.

3. Перед каждым перфорируемым символом следует набирать его регистр.

4. Оператор "Исполним" перфорируется без номера и пробела.

5. Оператор условного перехода перфорируется в такой последовательности: порядковый номер оператора, пробел, две первые буквы *ес*, условие, пробел, информация, идущая после слова "идти к", возврат каретки.

6. Оператор "Печатаем с *N* знаками" перфорируется в такой последовательности: порядковый номер оператора, пробел, первые две буквы *пе*, число знаков после запятой, переменные, которые должны выводиться на печать, с пробелами между ними, возврат каретки.

Для облегчения исправления ошибок на перфоленте рекомендуется после перфорации каждого оператора несколько раз пробивать символ нижнего регистра – тогда операторы будут отделяться друг от друга участками чистой ленты.

Исправление ошибок. Для составления программы в *ап* можно использовать только символы, предусмотренные в языке *ап*. Если в программе использован символ, не предусмотренный в *ап*,

то машина его рассматривает как неверную информацию, печатает знак * (признак неверной информации) и делает возврат каретки. При этом программа, составленная для предыдущих операторов, сохраняется.

Если же при введении оператора сделана ошибка, то программист может сам ввести *, что дает возможность заново ввести этот оператор. Если ошибочный оператор замечен после ввода, то для исправления его нужно ввести заново с тем же порядковым номером. Если ошибочный оператор замечен во время решения задачи, то решение следует прервать нажатием клавиши "Останов" на режимной панели, нажать клавишу "Пуск I", набрать на печатающей машинке сочетание букв па и ввести его правильно с тем же порядковым номером. Продолжить решение можно, набрав оператор "Исполним N" и возврат каретки.

Примеры составления программ в режиме ал.

I. Программа вычисления функции

$$y = \frac{m^2(1 - \sin \pi x)^2}{(m-1)^3}$$

при $x = 0,9 + 1,9$; $\Delta x = 0,05$; $m = 0,5 + 2$; $\Delta m = 0,5$.

ал

1. допустим $m = 0,85$ (вк)

2. интервал 3 (вк)

3. печатаем с 2 знаками m (вк)

4. допустим $x = 0,9$ (вк)

5. вычислим $y = (m^2(1 - \sin \pi x)^2)/(m-1)^3$ (вк)

6. печатаем с 5 знаками xy (вк)

7. вставим $x = x + 0,05$ (вк)

8. если $x < 1,9$ идти к 5 (вк)

9. вставим $m = m + 0,5$ (вк)

10. если $m < 2$ идти к 2 (вк)

II. кончаем (вк)

исполним I

2. Программа решения нелинейного уравнения $x = ax^2(\exp a^2 x) \times \frac{1}{x^2-1}$ методом простой итерации с точностью $\epsilon = 0,0001$, где $a = 0,75$. Задаемся произвольным значением x_0 и, подставив его в правую часть уравнения, вычислим новое значение x_1 . Сравниваем x_0 с x_1 и, если разница больше ϵ , находим следующее

значение x_2 , подставив x_1 в правую часть, и т.д. до тех пор, пока не удовлетворится неравенство $|x_n - x_{n-1}| < \epsilon$.

ал

1. допустим $a = 0,75$ $\epsilon = 0,0001$ (вк)

2. вычислим $y = (ax^2(\exp a^2 x))/(x^2-1)$ (вк)

3. вычислим $p = y - x$ (вк)

4. допустим $x = y$ $p = |p|$ (вк)

5. если $p > \epsilon$ идти к 2 (вк)

6. печатаем с 7 знаками x (вк)

7. кончаем (вк)

исполним I.

3. Программа вычисления функции

$$y = \frac{e^{x/8} \sin x}{a} \quad \text{при } x > 1,5;$$

$$y = ae^{x/4} \sin x \quad \text{при } x < 1,5,$$

где $x = 0,1 + 3$; $\Delta x = 0,1$, и построения графиков этой функции при трех значениях a ($a = 1 + 3$; $\Delta a = 1$).

ал

1. допустим $a = 1$ (вк)

2. печатаем с 1 знаками a (вк)

3. допустим $x = 0$ (вк)

4. вставим $x = x + 0,1$ (вк)

5. если $x < 1,5$ идти к 8 (вк)

6. вычислим $y = ((\sin x)(\exp x/8))/a$ (вк)

7. идти к 9 (вк)

8. вычислим $y = a(\sin x)(\exp x/4)$ (вк)

9. печатаем с 7 знаками xy (вк)

10. храним 30 y (вк)

II. если $x < 3$ идти к 4 (вк)

12. начертим I гр (вк)

13. интервал 4 (вк)

14. вставим $a = a + 1$ (вк)

15. если $a < 3$ идти к 2 (вк)

16. кончаем (вк)

исполним I.

Для индексированных переменных можно употреблять только буквенные индексы i, j, k , которым предварительно должны быть присвоены целочисленные значения с помощью операторов "Допустим" или "Вставим".

Если алгоритм содержит индексированные переменные, то перед набором первого оператора необходимо ввести в машину максимальные значения индексов и буквы, которые употребляются в данной задаче с индексами.

За начальное значение индексов, как правило, принимается нуль. В случае двухиндексных переменных количество столбцов матрицы должно быть задано буквой n .

4. Программа нахождения корней полинома

$$a_0 x^3 + a_1 x^2 + a_2 x + a_3 = 0,$$

коэффициенты которого определяются по формулам: $a_0 = 1,08$;
 $a_1 = e^{t-1}$; $a_2 = \sin t + \cos t / 2$; $a_3 = (\ln t) - 1$, где $t = 1,25$.

При построении алгоритма используем стандартную программу, к которой будем обращаться с помощью оператора "Решим".

- ан
 $i = 3a$
 I. допустим $t = 1,25$ $i = 0$ $a_i = 1,08$ (вк)
 2. вставим $i = i + 1$ (вк)
 3. вычислим $a_i = \exp(t - 1)$ (вк)
 4. вставим $i = i + 1$ (вк)
 5. вычислим $a_i = \sin t + \cos t / 2$ (вк)
 6. вставим $i = i + 1$ (вк)
 7. вычислим $a_i = \ln t - 1$ (вк)
 8. решим $px^3 = 0$ (вк)
 исполним I.

5. Программа умножения квадратных матриц 12-го порядка

$$\begin{vmatrix} a_{00} & a_{01} & \dots & a_{0n} \\ a_{10} & a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots & \dots \\ a_{n0} & a_{n1} & \dots & a_{nn} \end{vmatrix} \times \begin{vmatrix} b_{00} & b_{01} & \dots & b_{0n} \\ b_{10} & b_{11} & \dots & b_{1n} \\ \dots & \dots & \dots & \dots \\ b_{n0} & b_{n1} & \dots & b_{nn} \end{vmatrix}$$

Результат должен представлять матрицу вида

$$\begin{vmatrix} c_{00} & c_{01} & \dots & c_{0n} \\ c_{10} & c_{11} & \dots & c_{1n} \\ \dots & \dots & \dots & \dots \\ c_{n0} & c_{n1} & \dots & c_{nn} \end{vmatrix}$$

Алгоритм должен содержать индексированные переменные a, b и c с двумя индексами i, j , максимальные значения которых равны 12. Количество столбцов задается буквой $n = 12$. Вычислительный процесс будет состоять в получении элементов матрицы

$|c_{ij}|$ по формуле

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj} .$$

- ан
 $i = 12$ $j = 12$ $a b c$
 I. допустим $i = 0$ $n = 12$ (вк)
 2. допустим $j = 0$ (вк)
 3. допустим $c_{ij} = 0$ (вк)
 4. введем $a_{ij} b_{ij}$ (вк)
 5. вставим $j = j + 1$ (вк)
 6. если $j < n$ идти к 3 (вк)
 7. вставим $i = i + 1$ (вк)
 8. если $i < n$ идти к 2 (вк)
 9. допустим $i = 0$ (вк)
 10. допустим $j = 0$ (вк)
 II. допустим $k = 0$ (вк)
 12. вычислим $c_{ij} = c_{ij} + a_{ik} b_{kj}$ (вк)
 13. вставим $k = k + 1$ (вк)
 14. если $k < n$ идти к 12 (вк)
 15. печатаем с 9 знаками c_{ij} (вк)
 16. вставим $j = j + 1$ (вк)
 17. если $j < n$ идти к II (вк)
 18. интервал 4 (вк)

19. вставим $i = i + I$ (вк)
 20. если $i < n$ идти к 10 (вк)
 21. кончаем (вк)
 исполним I.

6. Программа нахождения коэффициентов степенного полинома наилучшего приближения точечной функции способом наименьших квадратов.

Пусть точечная функция задана таблично:

x	x_1	x_2	\dots	x_m
y	y_1	y_2	\dots	y_m

Требуется аппроксимировать ее полиномом третьей степени

$$y = a_0 x^3 + a_1 x^2 + a_2 x + a_3. \quad (1)$$

При этом коэффициенты полинома a_0, a_1, a_2, a_3 должны быть выбраны так, чтобы невязка

$$\mathcal{S} = \sum_{i=1}^m (y_i - (a_0 x_i^3 + a_1 x_i^2 + a_2 x_i + a_3))^2 \quad (2)$$

была минимальной.

При подстановке координат точек (x_i, y_i) в уравнение (1) получаем m уравнений с 4 неизвестными:

$$\begin{cases} a_0 x_1^3 + a_1 x_1^2 + a_2 x_1 + a_3 = y_1; \\ a_0 x_2^3 + a_1 x_2^2 + a_2 x_2 + a_3 = y_2; \\ \dots \\ a_0 x_m^3 + a_1 x_m^2 + a_2 x_m + a_3 = y_m. \end{cases} \quad (3)$$

Согласно принципу Лежандра система линейных условных уравнений (3), где число неизвестных меньше числа уравнений, сводится к системе линейных уравнений:

$$\begin{cases} a_0 x_1^3 + a_1 x_1^2 + a_2 x_1 + a_3 = y_1; \\ a_0 x_2^3 + a_1 x_2^2 + a_2 x_2 + a_3 = y_2; \\ \dots \\ a_0 x_m^3 + a_1 x_m^2 + a_2 x_m + a_3 = y_m; \end{cases} \quad (4)$$

$$\begin{aligned} a_3 \sum_{i=1}^m x_i^3 + a_2 \sum_{i=1}^m x_i^2 + a_1 \sum_{i=1}^m x_i + a_0 \sum_{i=1}^m 1 &= \sum_{i=1}^m y_i x_i^2; \\ a_3 \sum_{i=1}^m x_i^5 + a_2 \sum_{i=1}^m x_i^4 + a_1 \sum_{i=1}^m x_i^3 + a_0 \sum_{i=1}^m x_i^2 &= \sum_{i=1}^m y_i x_i^3, \end{aligned} \quad (4)$$

где число неизвестных равно числу уравнений. При этом обеспечивается минимум невязки \mathcal{S} . Система (4) может быть решена одним из известных способов.

ан

$$i = 3 \quad j = 4 \quad b$$

- I. допустим $i = 0 \quad K = 0 \quad n = 5$ (вк)
2. допустим $j = 0$ (вк)
3. допустим $b_{ij} = 0$ (вк)
4. вставим $j = j + I$ (вк)
5. если $j \leq 4$ идти к 3 (вк)
6. вставим $i = i + I$ (вк)
7. если $i \leq 3$ идти к 2 (вк)
8. допустим $a = 0 \quad j = 0$ (вк)
9. введем $x \ y$ (вк)
10. допустим $m = 0 \quad i = 0$ (вк)
- II. вычислим $b_{ij} = b_{ij} + \exp(a + m / \ln x)$ (вк)
12. вставим $i = i + I \quad m = m + I$ (вк)
13. если $i \leq 3$ идти к II (вк)
14. вставим $j = j + I \quad a = a + I$ (вк)
15. если $j \leq 3$ идти к 10 (вк)
16. допустим $m = 0 \quad i = 0$ (вк)
17. вычислим $b_{ij} = b_{ij} + y \exp m \ln x$ (вк)
18. вставим $i = i + I \quad m = m + I$ (вк)
19. если $i \leq 3$ идти к 17 (вк)
20. вставим $K = K + I$ (вк)
21. если $K < 10$ идти к 8. (вк)
22. допустим $K = 0$ (вк)
23. допустим $i = 0$ (вк)
24. допустим $j = 0$ (вк)
25. вычислим $c = b_{ik}$ (вк)
26. вычислим $b_{ij} = b_{ij} / c$ (вк)
27. вставим $j = j + I$ (вк)

28. если $j \leq 4$ идти к 26 (вк)
 29. вставим $i = i + I$ (вк)
 30. если $i \leq 3$ идти к 24 (вк)
 31. допустим $i = 0$ (вк)
 32. допустим $j = 0$ (вк)
 33. если $i = K$ идти к 37 (вк)
 34. вычислим $B_{ij} = B_{ij} - B_{kj}$ (вк)
 35. вставим $j = j + I$ (вк)
 36. если $j \leq 4$ идти к 34 (вк)
 37. вставим $i = i + I$ (вк)
 38. если $i \leq 3$ идти к 32 (вк)
 39. вставим $K = K + I$ (вк)
 40. если $K \leq 3$ идти к 23 (вк)
 41. допустим $i = 0$ (вк)
 42. вычислим $a = B_{ik} / B_{jj}$ (вк)
 43. печатаем с 9 знаками a (вк)
 44. вставим $i = i + I$ (вк)
 45. если $i \leq 3$ идти к 42 (вк)
 46. кончаем (вк)
 исполним I.

Здесь I – 2I-й операторы реализуют формирование коэффициентов системы (4) на основе вводимых с помощью 9-го оператора значений x_i , y_i , причем в машине для экономии памяти одновременно хранится не более одной пары значений x_i , y_i , в связи с чем отпадает необходимость в их индексации.

После формирования системы (4) 22 – 45-й операторы осуществляют решение ее методом диагональных элементов и печать результатов.

При аппроксимации точечной функции, заданной другим числом точек (т.е. пар значений x_i , y_i), нужно изменить чи-ло в записи 2I-го оператора.

(Исправление программы оператора "Печатаем" см. в приложении 3).

Примеры индивидуальных заданий по программированию.

I. Вычислить значение функции

$$y = \sqrt{x} \sin \pi (a_i^2 + x^2) \quad \text{при } x \leq 0,45;$$

$$y = \sqrt{x} \cos \pi (a_1^2 + x^2) \quad \text{при } x > 0,45,$$

где $x = 0,05 + 0,75$; $\Delta x = 0,05$; $a_1 = 0,25$; $a_2 = 0,2$; $a_3 = 0,27$; $a_4 = 0,09$, и построить график функции y при $a = 0,2$.

2. Вычислить корни полинома

$$P(x) = 0,725\beta^2 x^5 + 2\sin\beta x^3 - 0,125\beta^5 x = 0,5$$

при $\beta = 0,615$.

3. Дана квадратная матрица A порядка n . Определить номер строки и столбца минимального элемента.

4. Вычислить функцию

$$T = \frac{\operatorname{sh}(x^2 + y)}{xy - 1};$$

где $x = 0,15 + 0,75$; $\Delta x = 0,025$; $y = 0 + 0,6$; $\Delta y = 0,1$; $\varepsilon = 0,0001$.

При вычислении использовать формулы

$$\operatorname{sh} X \approx \sum_{k=1}^{\infty} U_k; \quad U_1 = X; \quad U_{k+1} = \frac{X^2}{2k(2k+1)} U_k.$$

§ 2. Командное программирование

Представление чисел и команд в ячейках памяти машины. ЭВМ "Наири-С" работает с числами, представленными в форме с фиксированной запятой в дополнительном коде. Кроме того, с помощью соответствующих псевдоопераций (специальных команд и микропрограмм) машина оперирует числами с плавающей запятой, целыми и длинными числами (т.е. числами с двойной длиной).

Разрядная сетка ячейки машины состоит из 36 двоичных разрядов, пронумерованных справа налево:

α_{36}	α_{35}	α_{34}	d_2	d_1
---------------	---------------	---------------	-------	-------

I. Числа с фиксированной запятой имеют вид

$$\alpha_{36} \cdot 2^1 + \alpha_{35} \cdot 2^0 + \alpha_{34} \cdot 2^{-1} + \dots + \alpha_1 \cdot 2^{-34},$$

где α_i принимает значения 0 или 1, $i = 1 + 36$; дополнительный код числа $[x]_{\text{доп}} = x + 2^2$ по $\text{mod } 2^2$.

Диапазон чисел с фиксированной запятой

$$\begin{aligned} -2 < x < -2^{-34}, & \text{ если } x < 0; \\ 2^{-34} < x < 2(1 - 2^{-35}), & \text{ если } x > 0; \\ x = 0, & \text{ если } x = 0. \end{aligned}$$

Каждый разряд ячейки имеет свое определенное назначение:

α_{36} – знаковый разряд. Если $\alpha_{36} = 0$, то число, записанное в ячейку, положительное, в противном случае – число отрицательное. Запятая фиксирована между разрядами α_{35} и α_{34} .

2. Числа с плавающей запятой представляются в виде $x = M \cdot 2^p$, где M – мантисса числа, p – порядок числа.

В этом случае разрядная сетка ячейки разбивается следующим образом:

α_{36}	α_{35}	α_8	α_7	α_6	α_1
---------------	---------------	------------	------------	------------	------------

Разряды α_1 – α_7 отводятся для порядка числа, причем α_1 – α_6 занимает величина порядка, а α_7 – знак порядка; разряды α_8 – α_{34} отведены для величины мантиссы числа, а разряды α_{35} , α_{36} – для знака мантиссы. Мантисса будет отрицательной только тогда, когда $\alpha_{35} = 1$ и $\alpha_{36} = 1$ одновременно, во всех остальных случаях мантисса воспринимается машиной как положительная.

Мантисса и порядок числа задаются в дополнительном коде.

3. Целое число представляется в машине в форме

$$\alpha_{36} \cdot 2^{35} + \alpha_{35} \cdot 2^{34} + \dots + \alpha_1 \cdot 2^0$$

$$[x]_{\text{доп}} = x + 2^{36} \text{ по } \text{mod } 2^{36}.$$

4. Машина "Наири-С" может работать с числами, записанными в двух последовательных ячейках, при этом целая часть записывается в первой ячейке, а дробная – в следующей. Такое число условно называется длинным. Если обозначить через α_i двоичные разряды целой части, а через β_i – двоичные разряды дробной части, то длинное число можно представить в виде

$$\alpha_{36} 2^{35} + \alpha_{35} 2^{34} + \dots + \alpha_1 2^0 + \beta_{34} 2^{-1} + \beta_{33} 2^{-2} + \dots + \beta_1 2^{-34}.$$

В этом случае число также записывается в дополнительном коде

$$[x]_{\text{доп}} = x \cdot 2^{36} \text{ по } \text{mod } 2^{36}.$$

5. Числа называются нормализованными, для которых мантисса лежит в интервале

$$\begin{aligned} \frac{1}{2} < M < 1, & \text{ если } M > 0; \\ -1 < M < -\left(\frac{1}{2} + 2^{-27}\right), & \text{ если } M < 0. \end{aligned}$$

Диапазон нормализованных чисел следующий:

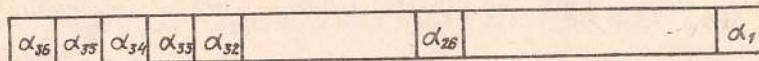
$$\begin{aligned} -2^{63} < x < -2^{64} \left(\frac{1}{2} + 2^{-27}\right) & \text{ при } x < 0; \\ \frac{1}{2} \cdot 2^{-64} < x < 2^{63} \left(1 - 2^{-27}\right) & \text{ при } x > 0. \end{aligned}$$

6. Нуль представляется как число, все разряды ячейки которого равны нулю.

7. Особенностью программирования на "Наири-С" является то, что команды могут быть записаны как во внешнем (буквенном) коде, так и во внутреннем (цифровом) коде. Программистами, как правило, используется внешний код команд. Команда во внешнем коде имеет вид $K A_1 \theta A_2$, где K – код операции; A_1 – первый адрес; A_2 – второй адрес и θ – модификация команды. Код операции указывает на производимое действие, т.е. это условное обозначение какой-либо арифметической или логической операции (сложение, вычитание, логический сдвиг и т.д.). В первом и втором адресах помещаются номера ячеек памяти машины,

в которых записаны числа, над которыми необходимо произвести указанное в коде действие. Модификация команд - это пояснение, как именно должно быть выполнено данное действие.

В машине каждая команда записывается в одну ячейку. Внутренний код команды представляет собой набор нулей и единиц. Команда занимает 36 разрядов ячейки; все разряды имеют определенное назначение



Разряды α_1 - α_{25} отведены для A_1 и A_2 , причем в зависимости от модификации количество разрядов для каждого адреса различно. Разряды α_{27} - α_{31} занимает код операции, $\alpha_{26} = \Phi$ употребляется для видоизменения команды. Если $\Phi = 0$, то команда выполняется без изменения так, как она написана. При $\Phi = I$ к данной команде прибавляется содержимое I-й ячейки памяти и выполняется полученная команда. Этим путем можно изменять адреса и код операции.

Разряд $\alpha_{33} = P$ является служебным, определяющим принадлежность команды к машинным операциям или же к псевдооперациям, α_{32} служит для модификации кода операций, а α_{36} , α_{35} , α_{34} отведены для модификации команды.

В машине "Наира-С" предусмотрено 7 различных модификаций, которые во внешнем коде команды обозначаются различными буквами, а во внутреннем коде - определенным сочетанием нулей и единиц в разрядах α_{36} , α_{35} , α_{34} ячейки.

Модификация Н (накопитель). Во внутреннем коде в разрядах α_{36} , α_{35} , α_{34} записаны нули.

При выполнении команды с модификацией Н адреса A_1 и A_2 воспринимаются как номера ячеек памяти машины, и операция выполняется над числами, записанными в этих ячейках.

Следующие три модификации дают возможность экономить ячейки памяти машины, поскольку первый адрес команды воспринимается не как номер ячейки памяти, а как число, и операция выполняется над этим числом и содержимым второго адреса.

Модификация К (правый короткий параметр). Во внутреннем коде в разрядах, отведенных для модификации, записано $\alpha_{36} = 0$; $\alpha_{35} = 0$; $\alpha_{34} = I$.

При выполнении команды с модификацией К первый адрес A_1 , занимающий разряды $\alpha_{12} - \alpha_{25}$, выделяется, сдвигается вправо на II разрядов (т.е. число, записанное по первому адресу, сдвигается в младшие разряды ячейки) и операция выполняется над полученным числом и содержимым второго адреса или фиксированной рабочей ячейки (a2).

Пример: Пусть в 100-й ячейке записана команда сложения двух чисел, находящихся в ячейках 110, и III, т.е.
100) CII0nIII.

Допустим, нужно изменить (увеличить) на 5 единиц второй адрес этой команды. Для этого можно использовать команду с модификацией K

101) C 5k100.

После выполнения этой команды в 100-й ячейке будет команда CII0nII6, т.е. второй адрес увеличится на 5 единиц.

С помощью этой модификации может быть изменена любая ячейка памяти (ОЗУ), т.е. A_2 может быть номером любой ячейки с I-й по 1023-ю. A_1 может быть не больше чем $2^{14} - I = 16383$, т.е. число, записанное по первому адресу, должно занимать не больше разрядов, чем отведено при этой модификации под A_1 .

Модификация P (правый длинный параметр). Во внутреннем коде в разрядах, отведенных для модификации, стоят $\alpha_{36} = 0$; $\alpha_{35} = I$; $\alpha_{34} = 0$. Команды с модификацией P выполняются аналогично командам с модификацией K, т.е. выделяется A_1 , сдвигается вправо на 7 разрядов и операция выполняется над полученным числом и содержимым второго адреса или (a2). Отличие этих модификаций состоит в том, что второй адрес занимает всего 7 младших разрядов ячейки (α_1 - α_7). Это значит, что на месте второго адреса может стоять только номер ячеек с I-й по 127-ю. Первый адрес занимает разряды α_8 - α_{25} и максимальное число, которое может быть записано на месте A_1 , равно $2^{18} - I = 262143$.

Модификация L (левый параметр). Во внутреннем коде разряды, отведенные для модификации, имеют значения $\alpha_{36} = 0$; $\alpha_{35} = I$; $\alpha_{34} = I$.

При выполнении команды с модификацией L число, записанное на месте A_1 , выделяется, сдвигается влево на II разрядов

и операция выполняется над полученным числом и содержимым второго адреса или (a2). Эта модификация выполняется аналогично модификации p , только сдвиг числа, записанного по A_1 , происходит влево. С помощью этой модификации удобно изменить внутренний код команды.

Пример: $M\ 124\ 116$.

Это значит, что надо сложить по модулю 2 левым параметром 124 с числом, записанным в ячейке 116 (т.е. число 124 будет записываться в разрядах ячейки, начиная с 19-го).

Модификации k , p , l можно использовать не только при формировании команд, но и при операциях с целыми числами.

Примеры:

1. Послать 1 по первому адресу в ячейку I можно с помощью команды: $p\ 2048\ k\ 1$.

2. Вычислить целое число 10 из содержимого ячейки 100 можно с помощью следующих команд: $B\ 10\ p\ 100$ или $B\ 10\ k\ 100$.

3. К содержимому 150-й ячейки прибавить целое число 20 можно только с помощью модификации k , так как модификация l не допускает, чтобы второй адрес команды был больше 127: $c20\ k\ 150$.

Следующие 3 модификации используются только для условных команд (см. "Условные команды") и выполняются аналогично безусловным модификациям.

Модификация H_y . Во внутреннем коде в разрядах, отведенных для модификации, $\alpha_{36} = 1$; $\alpha_{35} = 0$; $\alpha_{34} = 0$.

При выполнении условной команды с модификацией H_y операция производится над числами, находящимися в ячейках, номера которых записаны в A_1 и A_2 или (a2), если справедливо требуемое условие.

Модификация P_y . Во внутреннем коде в разрядах, отведенных для модификации, $\alpha_{36} = 1$; $\alpha_{35} = 1$; $\alpha_{34} = 0$.

При выполнении команды с этой модификацией число, записанное по первому адресу, выделяется, сдвигается вправо на 11 разрядов и операция выполняется над полученным числом и содержимым второго адреса или (a2), если удовлетворяется требуемое условие.

Модификация L_y . Во внутреннем коде в разрядах, отведенных для модификации, $\alpha_{36} = 1$; $\alpha_{35} = 1$; $\alpha_{34} = 1$.

При выполнении команды с модификацией L_y число, записанное по первому адресу, выделяется, сдвигается влево на 7 разрядов и операция выполняется над полученным числом и содержимым второго адреса или (a2), если удовлетворяется требуемое условие.

Система команд. Для выполнения действий над числами в машине предусмотрена система из 44 команд (см. приложение 4), которые являются простыми машинными операциями. Но помимо машинных операций в системе команд существуют еще так называемые псевдооперации, которые осуществляются с помощью группы машинных операций. Псевдооперации имеют также свой внешний и внутренний коды. Внешний код псевдооперации подобен коду машинных операций. При составлении программы программист может пользоваться псевдооперациями наравне с машинными операциями.

При описании операций приняты следующие обозначения:

(A_1) - код, находящийся в ячейке A_1 ;

$[A_1]$ - содержимое A_1 ;

$[A_1]_0$ - содержимое A_1 в зависимости от значения модификации;

$$\theta = \{n, p, k, l, p_y, n_y, l_y\};$$

$\rightarrow A$ - засылка в ячейку A ;

$C\bar{C}K$ - счетчик команд;

(\bar{A}_2) - арифметический сдвиг вправо (A_2);

(\bar{A}_2) - логический сдвиг влево (A_2).

При выполнении команд следует учесть:

1. Холостая команда не выполняет никакого действия. Обычно она используется для формирования переменных команд или для переадресации.

Пример:

$100) \times 2\ n$

$101) + 100\ n\ 110$

$110) \downarrow 20\ n\ 21$.

После выполнения 101-й команды содержимое 110-й ячейки увеличится на 2 по первому адресу и будет выполняться команда:

$110) \downarrow 22\ n\ 21$.

2. Команды $c, c_1, c_2, c_5, b, b_1, b_2, b_3, g, g_1, g_2, g_3, y, y_1, y_2, y_3, a, a_1, \delta, \delta_1, n, n_1$ используются только при работе с числами с фиксированной запятой или с целыми числами. Причем операции $c, c_1, b, b_1, y, y_1, g, g_1, a, a_1, \delta, \delta_1, n, n_1$ не анализируют результат действия. Машина будет давать правильные результаты, если истинный ответ удовлетворяет условию $-2 < x < 2$. Команды $c_2, c_5, b_2, b_3, y_2, y_3, g_2, g_3$ в отличие от предыдущих анализируют результат действия. Если $|x| \geq 1$, то номер следующей команды запоминается в ячейке I022 и управление передается в ячейку I023.

3. a2 - стандартная рабочая ячейка, в которой хранится результат выполнения операции.

4. Команда e , - относительный безусловный переход. По первому адресу этой команды задается число, на которое необходимо изменить содержимое счетчика команд, чтобы получить адрес начала новой последовательности команд.

Пример: I00) e, I14.

После выполнения I00-й команды будет выполняться команда с адресом I14, так как $I00 + I14 = I14$.

5. В операциях с комплексными и длинными числами нужно пользоваться значениями $\theta = n, \theta = k$.

6. После команд печати необходимо ставить команду 02270H, если печать идет в строку и 02274H, если печать идет в столбец.

7. Поскольку в машине "Наира-С" предусмотрены операции с плавающей запятой, то команда нормализации входит в микропрограммы этих операций. Отдельно команда нормализации может употребляться только при работе с числами с фиксированной запятой, когда нужно перейти от фиксированной запятой к плавающей.

Примеры составления программ в системе команд даны в приложении 5.

Условные команды во внешнем коде получаются из безусловных путем добавления к ним некоторого условия и адреса, содержимое которого проверяется. В общем виде внешний код условной команды имеет вид

$kA_1\theta A_2 zm,$

где k - код операции; A_1 - первый адрес; θ - модификация ($\theta = \{n_y, p_y, l_y\}$); A_2 - второй адрес (номера ячеек памяти с I-й по I5-ю, не больше!); z - проверяемое условие; m - адрес, для которого проверяется данное условие.

Условная команда выполняется в том случае, когда удовлетворяется условие z , в противном случае пропускается как холостая. В случае, когда $m = 0$, условие проверяется для (a2); z может принимать 8 значений, которые условно кодируются следующим образом:

- $z = 0$ проверяется условие $[m] = 0$;
- $z = 1$ проверяется условие $[m] \neq 0$;
- $z = 2$ проверяется условие $[m] > 0$;
- $z = 3$ проверяется условие $[m] > 0$;
- $z = 4$ проверяется условие $[m] < 0$;
- $z = 5$ проверяется условие $[m] < 0$;
- $z = 6$ проверяется условие "I вариант";
- $z = 7$ проверяется условие (∞) (условное переполнение).

При $z = 6$ условная команда выполняется, если на пульте управления нажата клавиша "Вариант".

Приводим примеры получения условных команд из безусловных.

Безусловные команды

1. $g 256nI4$

$\theta = n$.

Разделить содержимое I4-й ячейки на содержимое 256-й ячейки.

2. $u, 4785nI2$

$\theta = n$.

Обратиться к подпрограмме, начинающейся с ячейки 4785 с выходом в ячейку I2.

Условные команды

1. $g 256nI4 > 5$

$\theta = n_y$.

Если содержимое ячейки $5 > 0$, то разделить содержимое ячейки I4 на содержимое 256-й ячейки.

2. $u, 4785nI2 = 3$

$\theta = n_y$.

Если содержимое ячейки $3 = 0$, то обратиться к подпрограмме, начинающейся с ячейки 4785 с выходом в I2-ю ячейку.

3. $сп7И13$
 $\theta = H .$

Сложить числа с плавающей запятой, записанные в ячейках 7I и 13.

4. $в10К12$
 $\theta = K .$

Из второго адреса содержащего ячейки 12 вычесть 10.

5. $z10и0$
 $\theta = n .$

Ввести число в ячейку с номером 10.

3. $сп7И13\zeta$
 $\theta = H_y .$

Если результат предыдущего действия 0, то сложить числа с плавающей запятой, записанные в ячейках 13 и 7I.

4. $в10К12\infty$
 $\theta = K_y .$

Если значения в разрядах α_{36} и α_{35} различные, т. е. $[\alpha_{35}] \neq [\alpha_{36}]$, то вычесть из содержимого ячейки 12 число 10.

5. $10и12$
 $\theta = n_y .$

Если на пульте управления нажата клавиша "Вариант", то ввести число в ячейку 10.

Ввод чисел и команд в машину. Для ввода команды следует набрать на печатающей машинке:

адрес команды
букву К
команду
возврат каретки
букву Ш.

Если вводится целый массив команд, то следует набирать адрес первой команды
букву К
массив команд (после каждой команды возврат каретки)
букву Ш.

При вводе чисел правила остаются те же самые, только необходимо набирать одну из букв после ввода адреса:

m - для целого числа,
 z - для дробного числа,
 n - для чисел с плавающей запятой,
 g - для чисел с двойной длиной,
 c - для чисел в двоичном коде.

Пример: Чтобы записать число с плавающей запятой 25,3 в ячейку 256, надо на печатающей машинке набрать:

256п
25,3 (вк)
ш

Составление программ. Прежде чем составить программу решения той или иной задачи, нужно для нее выбрать метод решения и, если задача достаточно трудная, составить блок-схему (или же логическую схему) и распределить память машины. Выбор метода решения зависит от конкретной задачи и от возможностей машины. Если метод решения достаточно объемный, то его можно условно разбить на отдельные части - блоки. Последовательность этих блоков и будет блок-схемой решения задачи. Чтобы по блок-схеме составить программу, нужно каждый блок записать на машинном языке, т.е. командами. Составленная программа размещается в каком-либо участке памяти машины. При этом следует помнить, что некоторые ячейки использовать нельзя, поскольку они являются рабочими ячейками стандартных программ и микропрограмм.

Программы простейших арифметических и алгебраических выражений. При составлении таких программ нужно придерживаться некоторых правил.

1. В дробных выражениях сначала программируется знаменатель, а затем числитель.

2. В разностях сначала программируется вычитаемое, затем уменьшаемое.

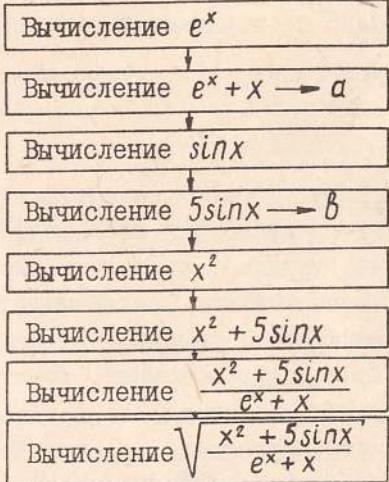
3. Избегать лишних машинных пересылок, используя различные модификации команд.

Арифметические и алгебраические выражения программируются в той же последовательности, как и при ручном счете, с той только разницей, что обычные операции (сложение, вычитание и т.д.) заменяются машинными командами.

Пример: Составить программу вычисления выражения

$$y = \sqrt{\frac{x^2 + 5\sin x}{e^x + x}} .$$

Сначала составляется подробная блок-схема.



Программа в машине должна быть помещена в каких-то определенных ячейках. У каждой ячейки есть свой номер, поэтому все команды пронумерованы по номерам ячеек, в которых они будут помещены.

I00) x	рабочие ячейки
I01) 5	
I20) e^x I00Н10	
I21) sp I00Н10	
I22) sn I00Н11	
I23) up I01Н11	
I24) n I00Н12	
I25) ul I00Н12	
I26) cp I11Н12	
I27) gp I12Н10	
I28) kp I10Н13	
I29) KOH	
I30) $e^x + x$	
I31) $5\sin x$	
I32) x	
I33) значение корня	

Разветвляющиеся программы. Часто встречаются задачи, в которых в зависимости от какого-либо условия необходимо вести вычисления по разным формулам.

Программы таких задач называются разветвляющимися. Программы разветвляются с помощью условных команд.

Пример: Предположим, что каким-либо путем получается значение аргумента x , величину которого заранее предсказать нельзя. Для этого значения x , в зависимости от его величины, вычислить

$$y = x^2 + 3, \text{ если } x > 1;$$

$$y = e^x + 1, \text{ если } x \leq 1.$$

Программа решения этого примера следующая:

исходные данные в ячейки вводятся как цифры с плавающей запятой

I00) x

I01) 1

I02) 3

рабочие ячейки

I03) для x

I04) для e^x

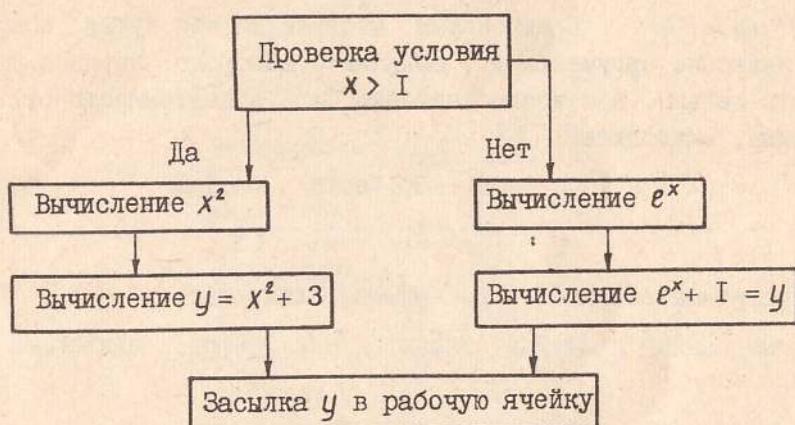
Сама программа занимает 12 ячеек, начиная со I10-й

I10) n I00Н103 } блок проверки условия
I11) bp I01Н103 }

I12) u I17п >
I13) e^x I00Н104 }
I14) cp I01Н104 }
I15) n I04Н105 }
I16) u I21п } вычисления $y = e^x + 1$

I17) n I00Н103 }
I18) up I00Н103 } вычисления $y = x^2 + 3$
I19) cp I02Н103 }
I20) n I03Н105 }
I21) KOH

Блок-схема этой программы такова:



В разветвляющихся программах нужно следить, чтобы конечный результат в каждой ветви получался в одной и той же ячейке.

Программа в трех направлениях разветвляется аналогично.

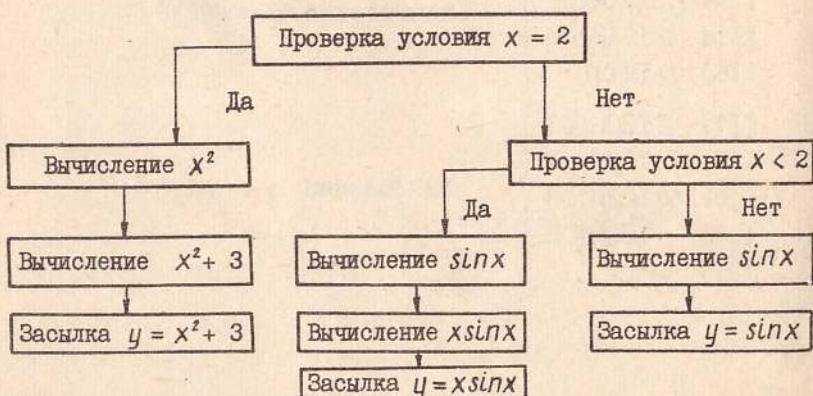
Пример: Вычислить

$$y = \sin x, \text{ если } x > 2;$$

$$y = x \sin x, \text{ если } x < 2;$$

$$y = x^2 + 3, \text{ если } x = 2.$$

Блок-схема разветвления этой программы:



Массив рабочих ячеек:

I00) x			
I01) 3			
I02) 2			
I03) для x			
I04) sin x			
I05) для y			
I10) n I00ИI03 } проверка условия			
I11) bп I02ИI03 }			
I12) uI18п =			
I13) uI23п <			
I14) sпI00ИI04 } вычисления y = sin x			
I15) yпI00ИI04 }			
I16) пI04ИI05 }			
I17) uI24п }			
I18) пI00ИI03 }			
I19) yпI00ИI03 }			
I20) сп I01ИI03 }			
I21) П I03ИI05 }			
I22) uI24п }			
I23) sпI00ИI05 } вычисления y = sin x			
I24) к 0н }			

Переадресация. При решении задач, в которых необходимо вести вычисления по одной и той же формуле, но с различными значениями аргументов, представляет интерес случай, когда значения аргументов совершенно произвольны.

Пример: Найти значения функции $y = x^2 + \sin x$ для следующих значений x :

$$x_1 = 1; \quad x_2 = 5; \quad x_3 = 7; \quad x_4 = 8; \quad x_5 = 12.$$

Для решения этой задачи удобно все значения аргументов поместить в один массив ячеек, следующих друг за другом, т.е.

I00) 1			
I01) 5			
I02) 7			
I03) 8			
I04) 12.			

Ячейки I05 и I06 будут рабочими

- I10) для x
- I16) для $\sin x$.

Программа вычисления значения функции для первого значения аргумента x_1

- I10) П100Н105
- I11) УП100Н105
- I12) SP100Н106
- I13) СП105Н106.

Чтобы вычислить значение функции y для второго значения аргумента x_2 , нужно изменить I10-ю команду так: I10) П101Н105 и далее вычисления вести по той же программе.

Команды, которые меняются в процессе работы программы, называются переменными. В машине "Наира-С" изменение команд осуществляется с помощью индексного регистра (I-я ячейка памяти машины). Делается это следующим образом. В I-ю ячейку записывается число, на которое должна быть изменена переменная команда, а после переменной команды ставится знак "+". Если в программе встречается команда, в конце которой стоит знак "+", то при исполнении к этой команде прибавляется содержимое индексного регистра и каждый раз выполняется вновь полученная команда.

В приведенном примере I10-я команда меняется каждый раз на I по первому адресу. Это значит, что в I-ю ячейку нужно записать *const* 2048 (I по первому адресу), т.е. программа вычисления будет следующей:

- I10) П2048К1
- I11) П99Н105+
- I12) УП105Н105
- I13) SP99Н107+
- I14) СП105Н107
- I15) С2048К1
- I16) УИ11П.

Можно иначе записать I по первому адресу в I-ю ячейку. Для этого в какую-либо ячейку записать холостую команду Х1Н0 и далее в первую ячейку сделать пересылку этой команды.

Например: I20) Х1Н0
I20) П120Н1.

Прервать вычисления по этой программе можно одним из способов выхода из циклических программ.

Циклические программы. Если какой-либо участок программы повторяется несколько раз, то его принято называть циклом. Существует несколько способов организации циклов.

1. Если количество циклов заранее известно, то программа цикла повторяется при помощи счетчиков, которые бывают двух типов: счетчик по убыванию и счетчик по возрастанию.

В первом случае в ячейку, отведенную для счетчика, заносится число повторений цикла и после каждого просчета из этой ячейки вычитается 1. Программа цикла будет повторяться, пока в счетчике не появится 0.

Во втором случае в ячейку счетчика заносится в начале работы программы 0, а после каждого повторения к этой ячейке добавляется 1. Повторения будут продолжаться до тех пор, пока величина в счетчике не достигнет величины заранее заданного числа повторений.

Для иллюстрации приводим программу предыдущего примера с организованным циклом со счетчиком по убыванию

- I20) П5К108
- I21) П2048К1
- I22) П99Н105 +
- I23) УП105Н105
- I24) SP99Н107 +
- I25) СП105Н107
- I26) С2048К1
- I27) В1К108
- I28) УИ22П #
- I29) КОН.

2. Если количество циклов заранее не известно (например, идет итерационный процесс вычислений), то выход из цикла осуществляется, например, по заданной точности, т.е. когда вычисляемая величина достигнет заданной точности, вычисления по этому циклу прекращаются.

Подпрограммы. В памяти машины (ДЗУ) "Наири-С" жестко записано несколько стандартных программ типовых задач, что значительно облегчает работу программиста, так как существенно сокращает время подготовки задачи к решению (составление программ, отладка и т.д.).

Для обращения к подпрограмме используется команда i, A_1, A_2 , где A_1 – адрес начала подпрограммы, A_2 – ячейка выхода из подпрограммы.

Решение системы линейных алгебраических уравнений n -го порядка ($n \leq 28$). Для решения системы линейных алгебраических уравнений в подпрограмме применен метод "главных элементов". Программа решения может быть использована как самостоятельная и как подпрограмма при решении других задач.

При решении конкретной системы линейных алгебраических уравнений (программа самостоятельная) с печатающего устройства вводятся исходные данные следующим образом. Нажимается клавиша "Пуск I" на сигнальной панели. Затем на печатающей машинке набирается $64n$ и вводятся построчно все коэффициенты системы (после каждого "Возврат каретки" или "Пробел"). После ввода всех коэффициентов набирается буква $ш$ и либо сочетание букв $су$, либо просто начальный адрес программы 7688Ц. Программа, начиная работать, печатает букву $n =$, после этого нужно задать на печатающей машинке порядок системы n и "Возврат каретки". В дальнейшем программа выполняется автоматически, результаты печатаются в виде десятичных чисел

$x_1 =$
 $x_2 =$
 \vdots
 $x_n =$

и машина останавливается.

Если программа используется как подпрограмма, то нужно все коэффициенты системы алгебраических уравнений заслать построчно в ячейки памяти начиная с 64-й; порядок системы целым числом записать в ячейку 993; в ячейку 980 записать команду выхода из подпрограммы; начальный адрес подпрограммы 7723.

В ходе решения системы возможны следующие случаи:

I. При $n > 28$ печатается "не могу".

2. Если система несовместна, машина печатает "реш.нет" и останавливается, если совместна, но решение не единственное, машина печатает ∞ и останавливается. При нажатии клавиши "Пуск 2" машина печатает "реш.нет" и останавливается.

3. При несовпадении контрольных сумм печатается "не совп" и машина останавливается. В этом случае необходима новая загрузка коэффициентов с 64 ячейки и система решается заново.

Результаты получаются в ячейках $64 + 64 + n$ и печатаются, если клавиша "Вариант" не нажата. В противном случае происходит выход из подпрограммы без печати результатов.

При несовпадении контрольных сумм можно на печатающей машинке набрать адрес 7889 и машина выдаст менее точные результаты.

Интегрирование системы n обыкновенных дифференциальных уравнений I-го порядка методом Рунге-Кутта. В системе дифференциальных уравнений $y'_i = f_i(x, y_1, y_2, \dots, y_n)$ ($i=1, \dots, n$) необходимо y_1, y_2, \dots, y_n заменить соответствующими символами, расположенными в следующем порядке:

$c, b, y, g, l, m, z, h, x, a, \beta, \gamma, \chi, u, o, e, l, p, t, w, \rho$
 а аргумент x заменить символом t , тогда система запишется в таком виде

$$\begin{aligned} c' &= f_1(t, c, b, y, \dots); \\ b' &= f_2(t, c, b, y, \dots); \\ y' &= f_3(t, c, b, y, \dots). \end{aligned}$$

Исходные данные вводятся как числа с плавающей запятой в следующие ячейки:

66 – ε – точность;

78 – h – шаг интегрирования; если интегрирование делается с автоматическим выбором шага, то h есть начальный шаг;

79 – M – конец интервала интегрирования;

80 – t_0 – начальное значение аргумента;

81 - 80 + n - начальные значения;
197 - H - шаг печати результатов.

На клавиатуре печатающего устройства при интегрировании с постоянным шагом набрать rk или 7352 и, а при автоматическом выборе шага - dy или 7354 и.

Когда машина печатает $n =$, нужно набрать количество уравнений системы и "Возврат каретки".

Когда машина печатает $c' = b' =$, то надо задать функции

$$f_1(t, c, b, y \dots); \\ f_2(t, c, b, y \dots) \quad \text{и т.д.}$$

и "Возврат каретки" после каждой функции.

Нажатием клавиши "Вариант" обеспечивается вывод результата решения задачи в точках $t_0 + H, t_0 + 2H + \dots + M$.

При решении задачи могут встретиться следующие случаи:

1. Если машина печатает $n > 21$ (при решении систем с постоянным шагом) или $n > 17^*$ (при автоматическом выборе шага), то это означает, что машина не может решить систему, так как количество уравнений больше, чем предусмотрено в программе.

2. Если машина печатает "не могу", то это значит, что программа вычислений правых частей не помещается в ячейках 370 - 979.

3. По этой программе можно решать системы, в правых частях которых имеется не более 125 различных постоянных (без учета показателей степени), которые помещаются в ячейках 201 - 325.

4. Если при решении задачи произошел сбой, то для повторного пуска задачи нужно восстановить начальные данные и передать управление в ячейку 7428.

5. Чтобы остановить машину после окончания интегрирования системы обыкновенных дифференциальных уравнений, нужно в ячейку 980 заслать команду "Останов".

Если необходимо продолжить интегрирование, то следует сохранить содержимое ячеек $76 + 80 + n ; 80 + 3n + 1 + 80 + 4n$; содержимое массива ячеек начиная с 201-й, где хранятся постоянные, входящие в правые части данной системы; массива ячеек начиная с 370-й, где хранятся команды программ вычисления правых частей данной системы, составленных с помощью ал. Если программы вычисления правых частей составляет программист, то он

должен адрес начала этих программ как целые числа заслать в ячейки $80 + 3n + 1 + 80 + 4n$, количество уравнений системы - в ячейку 77, а в ячейку 76 заслать нуль (при варианте интегрирования с постоянным шагом) и единицу (при варианте с автоматическим выбором шага). При составлении программы вычисления правых частей надо учитывать, что значения y_1, \dots, y_n , которые входят в правые части соответственно, помещаются в ячейки 33 - 53, а значение аргумента - в ячейку 54. Значение функции правой части следует поместить в 20-ю ячейку. Для программы вычисления правых частей надо использовать ячейки 201 - 979. В конце программы вычисления каждой правой части надо добавить команду и 7470п. Начальный адрес программы 7428. Результаты интегрирования получаются в ячейках $80 + 80 + n$ и $80 + n + 1 + 80 + 2n$.

Вычисление определенного интеграла. Определенный интеграл вычисляется по формуле Симпсона

$$\int_a^b y dx \approx \frac{2}{3} h \left[\frac{y_0 + y_{2m}}{2} + 2(y_1 + y_3 + y_5 + \dots + 2y_{2m-1}) \right],$$

где $h = \frac{b-a}{2m}$ (интервал интегрирования разделен на $2m$ равных частей).

Если программа вычисления интегралов используется как самостоятельная программа, то нужно ввести исходные данные как числа с плавающей запятой в следующие ячейки:

70 - a - нижний предел интегрирования;

71 - b - верхний предел интегрирования;

72 - ϵ - точность вычисления;

ш;

на клавиатуре печатающего устройства набрать ил или 7996 и; когда машина остановится и загорится лампочка "Чтение", задать подинтегральную функцию $f(x)$ и "Возврат каретки".

Если программа используется как подпрограмма, нужно заполнить ячейки 70, 71, 72, как и в предыдущем случае; в ячейку 74 послать команду и0п 74; в 980-ю ячейку заслать команду выхода i к 980 (i - номер команды, к которой нужно вернуться после вычисления интеграла) и передать управление в ячейку 7997 командой и 7997п; когда машина остановится на

"Чтении", нужно задать подынтегральную функцию $f(x)$ и "Возврат каратки"; ответ получается в ячейке 67.

Примечание. 1. Если машина печатает "не могу" сразу после задания $f(x)$, то это значит, что программа вычисления $f(x)$ не помещается в ячейках 76 - 200. В этом случае $f(x)$ нужно разбить на $f_1(x)$ и $f_2(x)$ так, чтобы программы их вычисления поместились в ячейках 76 - 200, и результаты просуммировать.

2. Если машина печатает "не могу" после вычислений, то это значит, что $|h_k| < 2^{-24}$.

3. Если программа вычисления интегралов используется как подпрограмма внутри какого-либо цикла и подынтегральная функция при этом не меняется, а пределы интегрирования меняются, то $\text{п}0\text{л}74$ пишется вне цикла, в противном случае - внутри цикла.

4. По данной программе можно вычислять интеграл, если подынтегральное выражение содержит не более чем 125 различных постоянных.

Программа ЛА предназначена для вычисления величин: определителя (во); обратной матрицы (ом); собственных чисел матрицы (сч); собственных векторов матрицы (св).

В программе реализован метод Леверни с видоизменением Д.К.Фадеева¹⁾.

Одним обращением к программе можно вычислить либо все указанные величины одновременно, либо одну из них, либо некоторую комбинацию их.

Программа ЛА имеет несколько входов для вычисления каждой из перечисленных величин в отдельности или их комбинаций.

II3I2 - во
II3I4 - ом
II3I6 - сч
II3I8 - сч, св
II320 - во, ом, сч, св
II322 - во, ом
II324 - во, сч
II326 - ом, сч
II328 - во, сч, св
II330 - ом, сч, св.

¹⁾ Д. К. Фадеев, В. Н. Фадеева. Вычислительные методы линейной алгебры. М., Физматгиз, 1963.

Программа может использоваться как самостоятельная и как подпрограмма при решении других задач.

В случае самостоятельной программы исходная матрица вводится как числа с плавающей запятой по строкам в массив ячеек начиная с 61-й. Обращение к программе осуществляется с печатающего устройства по команде Nu , где N - входной адрес соответствующей части программы. Кроме того, для вычисления определителя и обратной матрицы можно обратиться к программе, напечатав сочетание букв: $b0$ - для вычисления определителя; om - для обращения матрицы. После выполнения команды om машина печатает $n =$, после чего следует напечатать порядок матрицы и нажать "Возврат каратки".

Машина переходит к решению.

Если программа используется как подпрограмма, исходная матрица размещается по строкам как числа с плавающей запятой в ячейках начиная с 61-й. В 40-ю ячейку записывается порядок матрицы как целое число, в 32-ю - целое число в соответствии со следующей таблицей.

Вычисляемая величина	Целое число	Команда, осуществляющая засылку целого числа в 32-ю ячейку
во	1	$pIn32$
ом	2	$p2n32$
сч	4	$p4n32$
сч, св	12	$p12n32$
во, ом, сч, св	15	$p15n32$
во, ом	3	$p3n32$
во, сч	5	$p5n32$
ом, сч	6	$p6n32$
во, сч, св	13	$p13n32$
ом, сч, св	14	$p14n32$

Вход в программу осуществляется по адресу II368, выход - по адресу 980. Например, для обращения к программе вычисления определителя и обращения матрицы 4-го порядка следует написать команды:

$p4n40$
 $p3n32$
 $pII368k980.$

При работе программы ЛА в обоих случаях результаты выводятся в виде десятичных чисел. Перед каждой величиной печатается ее сокращенное название:

опрд. =

обр.матр. и др.

Собственные числа обозначены буквой x и при выдаче результатов печатается

$x_1 =$

$x_2 =$

$x_n = .$

При печати собственных векторов перед каждым вектором ставится буква b_i , где i - числовой индекс. Результаты находятся в следующих ячейках: определитель - в ячейке 52; обратная матрица по строкам - в $6I + n^2 \div 60 + 2n^2$; собственные числа - в $6I + 5n^2 \div 60 + 5n^2 + 2n$ и $838 \div 837 + 2n$; собственные векторы не сохраняются; коэффициенты характеристического полинома матрицы начиная со второго - в $6I + 5n^2 + 2n + 60 + 5n^2 + 3n$. Каждое собственное число занимает две соседние ячейки, в первой из которых находится действительная часть, во второй - мнимая.

При вычислении собственных векторов значение определителя и обратная матрица стираются.

Примечание. 1. Соответствие между индексами собственных чисел и собственных векторов (т.е. между x_i и b_i) не всегда соблюдается. Чтобы выяснить это соответствие, следует выдать на печать содержимое $2n$ ячеек начиная с 838-й как числа с плавающей запятой. В этом массиве находятся собственные числа (каждое собственное число занимает две соседние ячейки). Собственные векторы печатаются в том же порядке, в каком расположены в данном массиве собственные числа.

2. Если при обращении матрицы исходная матрица окажется особенной, то печатается "матр.особ."

3. Если контрольная матрица программы нулевая и нажата клавиша "Вариант", то печатается "контр.матр." и те элементы матрицы, которые не равны нулю. Если клавишу "Вариант" нажать перед вычислением собственных векторов (после того, как печатается значение последнего собственного числа), то для каждого собственного числа будет напечатано n собственных векторов. Это необходимо в случае, когда печатается нулевой вектор, т.е. первый столбец матрицы Q_K нулевой.

4. Если порядок матрицы $n > 12$, машина печатает " $n > 12$ не могу" и переходит к приему нового порядка. Максимальный порядок матрицы $n = 12$. Исходная матрица сохраняется.

Умножение матриц. Умножение матрицы A на матрицу B происходит по правилу "строка на столбец". Программа предназначена для умножения как квадратных, так и прямоугольных матриц.

Для прямоугольных матриц приняты обозначения: n - число строк 1-й матрицы; m - число столбцов 1-й матрицы; p - число столбцов 2-й матрицы.

Для квадратных матриц: n - порядок матриц.

Исходные матрицы записываются по строкам начиная с 51-й, т.е. $5I + 50 + pm$ - 1-я матрица и $5I + pm + 50 + pm + pm$ - 2-я матрица (для квадратных матриц соответственно $5I + 50 + p^2$ и $5I + p^2 + 50 + 2p^2$). Произведение матриц получается в массиве, следующем за массивом исходных данных, т.е. в ячейках $5I + pm + pm + 50 + pm + pm + pm$ (или $5I + 2p^2 + 50 + 3p^2$ для квадратных матриц).

Если программа используется как самостоятельная, следует обратиться к ней по команде 2904 и. После выполнения этой команды машина печатает $n =$, после чего для квадратных матриц следует напечатать порядок матрицы и пробел, а для прямоугольных матриц сделать пробел, напечатать число строк первой матрицы и еще раз сделать пробел. Наличие пробела после $n =$ является признаком того, что матрицы прямоугольные, поэтому машина печатает $m =$. Далее, не делая пробела, напечатать число столбцов первой матрицы и дать пробел. Затем машина печатает $p =$. После этого напечатать число столбцов второй матрицы и сделать пробел.

Машина переходит к вычислению.

Например, 2904 и $n=$ пробел 5 пробел, $m=3$ пробел, $p=1$ пробел.

Примечание. Если сделать пробел после $m =$ или $p =$, то машина неправильно воспринимает информацию. В этом случае следует снова обратиться по команде 2904 и.

Если программа используется как подпрограмма для квадратных матриц следует в ячейку 32 записать порядок матрицы как целое число; исходные матрицы записать по строкам в ячейки:

$5I + 50 + n^2$ - 1-я матрица; $5I + n^2 + 50 + 2n^2$ - 2-я матрица и передать управление по адресу 2970 командой $u, 2970k980$.

Результат получается в ячейках $5I + 2n^2 + 50 + 3n^2$.

В случае прямоугольных матриц в ячейки 30, 31, 32 записать размеры матриц n, m, p как целые числа и в ячейки 45, 46, 47 записать порядки n, m, p по 1-му адресу. Это можно осуществить следующими командами:

$n "p "n30$

$\delta, II n45$

$n "m "n31$

$\delta, II n46$

$n "p "n32$

$\delta, II n47.$

Исходные матрицы записать по строкам в ячейки $5I + 50 + pm$ - 1-я матрица; $5I + pm + 50 + pm + mp$ - 2-я матрица и передать управление по адресу 2952 командой $u, 2952k980$.

Произведение матриц получается в ячейках $5I + pm + mp + 50 + pm + mp + pr$ по строкам. Если клавиша "Вариант" нажата, результат не печатается.

Максимальные размеры матриц должны удовлетворять условиям: для квадратных $p \leq 17$; для прямоугольных $pm + mp + rp \leq 900$. Исходные матрицы сохраняются.

Решение алгебраических уравнений. Для нахождения корней полинома применена комбинация методов Берстру и Бернулли.

Программа предназначена для нахождения корней полинома $a_0x^n + a_1x^{n-1} + \dots + a_n = 0$, где a_i ($i = 0, \dots, n$) - действительные коэффициенты; $a_0 \neq 0$ и порядок $n \leq 38$.

Корни печатаются по мере нахождения. Уравнение в машину можно ввести различными способами.

Уравнение набирается вручную с печатающим устройством. Для этого необходимо в режиме "Универсальный" нажать клавишу "Пуск I" и набрать на печатающем устройстве сочетание ay .

Затем набрать уравнение в общепринятой математической записи в порядке убывания степени неизвестного до знака " $=$ ".

Члены с нулевыми коэффициентами пропускаются. После знака " $=$ " автоматически печатается "0" и начинается решение уравнения.

Корни печатаются в следующем виде:

$x_{10} = -0,313028 + i 0,38256; x_9 = -0,313028 - i 0,38256$ и т.д.

Индекс при неизвестном убывает от n до 1. После нахождения всех корней машина останавливается, ждет ввода нового уравнения. Если уравнение не решается с принятой в программе точностью (10^{-6}), то машина печатает значение точности вычислений и запрашивает новое:

$$\begin{aligned}\delta &= 0,000\ 000\ 23 \\ \delta &= .\end{aligned}$$

Если программиста устраивает значение корня с меньшей точностью, то он задает после знака равенства новое значение. Например: $\delta = 0,00001$.

После нажима клавиши "Возврат каретки" уравнение продолжает решаться с новой точностью.

Программа позволяет вычислить корни с точностью $\delta = 10^{-4}$, если корни простые. При кратных корнях возможна более низкая точность.

Если после ввода уравнения с печатающего устройства и его решения нужно автоматически перейти к какому-либо адресу, то следует предварительно заслать в ячейку 980 команду вида $u Np$, где N - номер команды, к которой следует переход; набрать II874 и только после этого ввести уравнение в виде, описанном ранее.

Чтобы заслать коэффициенты уравнения программным путем или с перфоленты, нужно ввести $n + 1$ коэффициент уравнения (включая нулевые) в ячейки 836 + 836 + n , показатель степени n ввести в ячейку 993, код буквы, которой обозначена переменная, ввести в ячейку 1017 целым числом (причем вводится внутренний код, например, код $x = 521$). После этого перейти к команде II994, если после решения следует останов.

Если после решения необходим переход к какому-либо адресу, то осуществляется переход к команде по адресу II995. При этом заранее должна быть послана команда $u Np$ в ячейку 980.

Примечание. 1. При решении уравнения можно производить печать промежуточных результатов для контроля хода решения. Для этого нажать клавишу "Вариант".

2. Результаты решения хранятся в ячейках памяти $830 + 838 + 2n - I$. Каждое значение корня занимает 2 ячейки: в первой записана его действительная часть, во второй - мнимая.

3. В случае грубых ошибок со стороны оператора при наборе уравнения с печатающим устройством машина печатает знак "*". В этом случае уравнение следует ввести заново. Если допущена ошибка в коэффициенте при наборе уравнения, то ее можно исправить, добавив член с правильным коэффициентом. Все исправления должны вводиться до набора свободного члена.

4. При больших степенях уравнения программа работает медленно из-за небольшой скорости машины и сложности метода.

5. После окончания решения уравнения значения корней расположены в ячейках $838 + 838 + 2n - I$.

Отладка программ. Для отладки написанной программы в машине предусмотрены:

Режим "Выдача памяти". Для перехода в этот режим необходимо нажать клавиши "Вывод на печать" и "Выдача памяти".

Для вывода на печать содержимого какого-либо адреса нужно напечатать:

адрес

букву *к* для команды

букву *т* для целого числа

букву *г* для дробного числа

букву *п* для числа с плавающей запятой

букву *д* для числа с двойной длиной

букву *с* для числа в двоичном коде

возврат каретки.

Если нужно вывести подряд несколько адресов, то напечатать адрес

одну из требуемых букв *к, т, г, п, д, с*

количество адресов, которое необходимо вывести
возврат каретки.

После этого машина автоматически выводит адрес и рядом его содержимое.

Пример: Вывести 5 команд начиная с адреса I25.

Печатаем *I25к5*, после чего машина выводит на печать

I25п30н15

I26од3н15

I27уп3н15

I28и, 268п12

I29е, 2н < 15.

Если вывод на печать производится на стандартные бланки, то после печати каждого 50 адресов машина останавливается, если нажата клавиша "Вариант" на панели сигнализации. После смены бланка вывод продолжается нажатием клавиши "Пуск 2".

При необходимости вывести только содержимое адреса (без печати самого адреса) нужно напечатать:

первый адрес

одну из требуемых букв *к, т, г, п, д, с*

количество адресов

букву *п*

возврат каретки.

Машина сначала выводит на печать первый адрес с соответствующей буквой и далее выводимые данные, по окончании печатается буква *ш* и 35 раз код пробела. Это связано с тем, что печать без адресов в основном применяется для вывода на перфорацию.

Пример: *I25к2п.*

Машина выводит

I25к

п30н15

од3н15

ш.

Иногда удобно выводить дополнительно адреса и команды в восьмеричном коде. Для этого нужно напечатать:

первый адрес

букву *к*

количество адресов

букву *б*

возврат каретки.

Пример: *3471к2б*

Машин выводит

347I 06617 2310 0777 20000 e,I6360p
347I 06620 2074 0020 0627 m515n23.

Примечание. Если выводится не команда, то вместо буквы к нужно напечатать одну из требуемых букв.

"Останов по адресу". Чтобы проверить какие-либо промежуточные результаты, удобно использовать "Останов по адресу". Для этого нужно на сигнальной панели нажать клавишу "Останов по адресу", а на пульте управления набрать адрес в восьмеричной системе, по которому нужно остановить вычисления. Для продолжения вычислений по программе надо нажать клавишу "Пуск 2".

"Шаговый режим". При каждом нажатии на клавишу "Пуск 2" будет выполняться очередная машинная операция. Для перевода машины в шаговый режим надо нажать клавишу "Шаговый" на пульте сигнализации. Если в программе применены псевдооперации, то удобнее пользоваться не шаговым режимом, а полуавтоматическим. В отличие от шагового при полуавтоматическом режиме работы при каждом нажатии на клавишу "Пуск 2" будет выполняться одна команда (безразлично, машинная это операция или псевдокоманда).

"Повторение операции". Чтобы повторить какую-либо операцию, следует набрать адрес требуемой операции в СЧК; нажать клавиши "Повторение операций" и "Пуск 2".

В этом случае непрерывно будет повторяться одна и та же команда при каждом нажатии клавиши "Пуск 2" один раз. Для повторения микрокоманды нужно нажать клавишу "Повторение микрокоманды", тогда при нажатии на клавишу "Пуск 2" будет один раз выполняться одна и та же микрокоманда.

ПРИЛОЖЕНИЯ

Приложение I

Внешние коды символов

Б п/п	НР		Код клавиш																	
	0	I	В ПрEV					Пробивка на перфоленте												
			Символ	Индекс	Символ	Степень	6	5	4	3	2	I	6	5	4	Син- хро.	3	2	I	
I	0						0	0	0	0	0	0				*				
2	I						I	0	0	0	0	0	I					*		*
3	2						2	0	0	0	0	0	I	0						
4	3						3	0	0	0	0	0	I	I						
5	4						4	0	0	0	0	I	0	0						
6	5						5	0	0	0	I	0	I							
7	6						6	0	0	0	I	I	0							
8	7						7	0	0	0	I	I	I							
9	8						8	0	0	I	0	0	0							
10	9						9	0	0	I	0	0	I							
II	a						,	0	0	I	0	I	0							
I2	б						/	0	0	I	0	I	I							
I3	ѣ		*				0	I	I	0	0	0								
I4	ѣ	l					I	I	I	I	0	0								
I5	ѣ	i					0	0	I	I	0	0								
I6	ѣ	K					I	0	0	I	I	0								
I7	ж	0					I	I	I	I	I	0								
I8	з	I					I	I	0	I	I	I								
I9	л	2					I	I	0	0	0	0								
20	к	3					I	0	I	I	0	0								
21	р	4					I	0	I	0	0	0								
22	с	5					I	I	0	0	I	0								
23	ч	6					I	I	I	0	I	I								
24	ѣ	7					I	0	0	0	0	I								
25	ѣ	8					I	I	I	I	0	I								
26	ѣ	9					0	I	I	0	0	I								
27	н		Ж				I	0	0	I	0	0								
28	о		ѣ				I	0	I	0	I	I								
29	п		я				I	0	I	I	I	0								
30	ф		ю				I	0	I	I	I	I								
31	ү		з				I	I	0	0	0	0								
32	ҳ		/				I	0	I	0	0	I								
33	н		щ				I	0	I	I	I	0								
34	ц		‡				I	I	0	0	I	I								
35	т		>				I	I	I	0	I	0								
36	ч		>				I	I	0	I	0	0								
37	ш		<				I	0	0	0	I	0								
38	=		€				I	0	0	0	0	0								
39)		∞				I	0	0	I	0	I								
40	(/				I	0	I	0	I	0								
41	,		l				I	0	I	I	0	I								
42	+		‡				I	0	0	0	I	I								
43	-		s				I	I	I	0	0	0								
44	*		t				I	I	I	0	0	I								
45	/		‡				I	0	0	I	I	I								
46	√		z				I	I	0	I	0	I								
47	8Р						0	I	0	I	0	0								
48	НР						0	I	0	I	I	0								
49	Перевод стр.						0	I	0	0	0	I								
50	вк						0	I	0	0	0	0								
51	Пробел						0	I	0	I	I	I								
52							0	I	0	0	I	I								
53							0	I	0	0	I	0								

Приложение 2

Распределение рабочих массивов памяти машины при работе ал

Для составления рабочей программы задачи в ал используются следующие массивы ячеек оперативной памяти машины.

Массив А - для переменных величин без индексов, представленных в форме с плавающей запятой, используются ячейки оперативной памяти (ОЗУ) с 32-й по 55-ю.

Массив Б - для переменных величин без индексов, представленных в форме с фиксированной запятой, которые принимают целочисленные значения, используются ячейки ОЗУ с 56-й по 62-ю. Причем, если переменные i, j, k, p не используются, то эти ячейки могут быть употреблены для хранения переменных величин без индексов, представленных в форме с плавающей запятой.

Массив В - для номеров операторов, которые являются константами с фиксированной запятой, используются ячейки ОЗУ с 69-й по 200-ю. Это значит, что в операторной программе может быть использовано не больше чем 132 оператора.

Массив Г - для числовых констант в форме с плавающей запятой, участвующих в решаемой задаче, используются ячейки с 201-й по 325-ю.

Массив Д - для хранения переменных с индексами отводится массив с конца памяти (с 980-й ячейки в сторону убывания). Размеры массива зависят как от максимальных значений индексов, так и от количества переменных с индексами.

Предположим, что индекс i (случай одного индекса), имеющий максимальное значение $N-1$, присвоен к переменным a, b , тогда массив для переменных a_i будет состоять из ячеек $(981 - N) + 980$, а для переменных b из $(981 - 2N) + (980 - N)$.

Если переменным a, b присвоены индексы i, j (случай двух индексов), имеющие соответственно максимальные значения $N_1 - 1$ и $N_2 - 1$, то массив для переменных a_{ij} занимает ячейки $(981 - N_1 N_2) + 980$, а для переменных b_{ij} $(981 - 2N_1 N_2) + (980 - N_1 N_2)$.

Массив Е - для рабочих ячеек программирующей и рабочей программы: $13 + 18; 20 + 31; 63 + 67; 236 + 369; 999 + 1003; 1006; 1007$.

Массив Ж - для хранения значений функций при построении графиков. Для него отводится место непосредственно после второго массива. Его длина определяется количеством значений функции, по которым строится график.

Рабочая программа, составленная ал по операторной программе, начинается с ячейки 370.

Если происходит перекрытие массивов, то задачу следует разделить на отдельные части, так как целиком она не входит в память машины.

Приложение 3

Исправление программы оператора "Печатаем"

Обычно при выводе результатов печатающая машинка делает 20-и пробелов после каждой выводимой величины, т.е. если выводится число 23,075, то делается 15 пробелов и только после этого печатается следующее число. Такой вывод очень удобен, когда отпечатанный результат используется в отчетах в виде таблиц. Но при таком выводе очень много времени теряется на печать. Для сокращения времени вывода на печать необходимо исправить программу, реализующую оператор "Печатаем". Программа, составленная в режиме *ап*, занимает массив ячеек начиная с 370-й, номера операторов располагаются в ячейках начиная с 69-й. Чтобы вывести на печать программу, реализующую оператор "Печатаем", необходимо к 68 прибавить *N* оператора "Печатаем", который он занимает в программе. Затем в режиме "Выдача памяти" вывести на печать содержимое $68 + N$ и $68 + N + 1$ ячеек в виде целых чисел. Тем самым определяется массив ячеек, в которых записана программа, реализующая оператор "Печатаем". Затем в виде команд в режиме "Выдача памяти" вывести этот массив на печать. После этого в универсальном режиме можно уже исправить программу печати.

Пример: Необходимо исправить оператор "Печатаем" в задаче (см. пример 3, стр. 73, который записан под № 9).

В режиме "Выдача памяти" нужно набрать в печатающей машинке: *77m2* и дать возврат каретки.

Машинка выведет на печать:

77 419

78 427.

Следовательно, программа печати записана в ячейках 419 - 426, т.е. занимает массив из 8 ячеек начиная с 419-й. С машинки в режиме "Выдача памяти" вводим информацию 419k8 и машина выведет на печать массив:

0 2220 x

0 2250 =

пп 41n518 печать содержимого 41-й ячейки

02214 н y

02250 н =

пп 35n518 печать содержимого 35-й ячейки

02274 н вк

и78 н переход к программе следующего оператора.

После нажатия клавиши "Пуск I" вносятся следующие изменения:

419к

x холостая операция

x холостая операция

пп 41n6 печать содержимого 41-й ячейки

02270н пробел

02270н пробел

пп 35n6 печать содержимого 35-й ячейки.

Затем можно переходить к решению задачи, т.е. перейти в режим *ап*: после нажатия клавиши "Пуск I" набирается *па* и исполним I.

Печать до исправления:

a = 1,0

x = 0,099999 y = 0,051244

x = 0,199999 y = 0,104951 и т.д.

Печать после исправления:

a = 1,0

0,099999 0,51244

0,199999 0,104951.

ПРИЛОЖЕНИЕ 4

Система команд "Нацири-С"

Номер команды	Название операции	Выполнение действия		Примечание	Выполнение операции	Изменение матрицы	Выполнение действия	Примечание
		3	4					
Числовые операции								
<i>x</i>	Хотостиль опечатка	$x \text{ } 0\theta$	(сик) \rightarrow сик	$a\theta$ сохраняется	δ	Логическая единица на A_i ; $\overline{A_i}$ на A_i $\rightarrow A_i; a\theta$		
<i>и</i>	Переход из состояния	$iA_i\theta$	$[A_i]_e^n \rightarrow$ сик	$a\theta$ сохраняется	ζ	Логическое значение на A_i $\rightarrow A_i; a\theta$		
<i>и</i>	Переход возвратом	$iA_i\theta A_i$	$[A_i]_e^n \rightarrow$ сик и возвратом	$a\theta$ сохраняется	η	Нормализация	$\overline{A_i} \theta A_i$, если возврат	
<i>ц</i>	Безусловный переход	$iA_i\theta \theta$	$[A_i]_e^n \rightarrow$ сик	Применяется только при выходе из цикла	ν	Логическое значение	$[A_i]_e^n \rightarrow$ сик $\rightarrow A_i; a\theta$	
<i>и</i>	Переход по ячейку	$iA_i\theta \theta$	$[A_i]_e^n \rightarrow$ сик	Выполняется если на матрице имеется ячейка	ρ	Логическое сложение	$[A_i]_e \vee [A_i]_e \rightarrow (a\theta)$	
<i>е</i>	Изменение с запоминанием сик	$eA_i\theta A_i$	(сик) $* [A_i]_e \rightarrow A_i; a\theta$		τ	Логическое сложение	$[A_i]_e \vee [A_i]_e \rightarrow A_i; a\theta$	
<i>е</i>	Относительный безусловный переход	$eA_i\theta$	(сик) $* [A_i]_e \rightarrow$ сик	$[A_i]_e$ рассматривается как целое число в сложении проксессора по под	λ	Логическое умножение	$[A_i]_e \wedge [A_i]_e \rightarrow A_i; a\theta$	
<i>п</i>	Передача числа	$nA_i\theta A_i$	$[A_i]_e \rightarrow A_i; a\theta$		μ	Рекомендация Округлить	$(a\theta) \cdot [A_i]_e \rightarrow (a\theta)$	
<i>и</i>	Останов	$iA_i\theta$	$[A_i]_e \rightarrow$ сик и останов		ν_1	Сложение по под	$[A_i]_e \rightarrow A_i; a\theta$	
<i>и</i>	Арифметический спектр: видимо	$iA_i\theta \theta$	$(a\theta) \rightarrow$ сик и останов		ν_2	Сложение (вывод)	$[A_i]_e^n \rightarrow$ печать	
<i>а</i>	Арифметический спектр: неправильно	$aA_i\theta A_i$	$[A_i]_e^n \rightarrow [A_i]_e \rightarrow A_i; a\theta$		ϑ	Сложение (вывод)	$((a\theta) + 2^n + 2^l) \rightarrow$ печать	
<i>и</i>	Сложение	$cA_i\theta A_i$	$(a\theta) * [A_i]_e \rightarrow A_i; a\theta$		χ	Число (целое)	$< p_2 BH; A_i; (a\theta)$	Рекомендуется $\theta = n$
<i>и</i>	Деление	$dA_i\theta A_i$	$(a\theta) * [A_i]_e \rightarrow A_i; a\theta$	Операции выполнения без анализа	ζ_1	Сложение	$\zeta_1 A_i \theta A_i$, $(a\theta) * [A_i]_e \rightarrow A_i; (a\theta)$	Операции выполнения с анализом результата. Если
<i>и</i>	Печать чисел с фиксированной запятой	$pA_i\theta \eta$	$(a\theta) * [A_i]_e \rightarrow A_i; (a\theta)$	запятой после запятой	ζ_2	Сложение	$\zeta_2 A_i \theta A_i$, $(a\theta) * [A_i]_e \rightarrow A_i; (a\theta)$	
Последовательности								
<i>б</i>	Вчитывание	$bA_i\theta A_i$	$(A_i) - [A_i]_e \rightarrow A_i; (a\theta)$	Матричная операция	δ_1	Вчитывание	$bA_i\theta A_i$, $(A_i) - [A_i]_e \rightarrow A_i; (a\theta)$	Для чисел с плавающей запятой
<i>б</i>	Вчитывание	$bA_i\theta A_i$	$(a\theta) - [A_i]_e \rightarrow A_i; (a\theta)$	Матричная операция	δ_2	Вчитывание	$bA_i\theta A_i$, $[A_i]_e - (A_i) \rightarrow A_i; (a\theta)$	Для чисел с плавающей запятой
<i>у</i>	Умножение	$yA_i\theta A_i$	$(A_i) * [A_i]_e \rightarrow A_i; (a\theta)$	Умножение на единицу результата	ψ	Умножение	$yA_i\theta A_i$, $(A_i) * [A_i]_e \rightarrow A_i; (a\theta)$	Для чисел с плавающей запятой
<i>у</i>	Умножение	$yA_i\theta A_i$	$(a\theta) * [A_i]_e \rightarrow A_i; (a\theta)$	Умножение на единицу результата	ψ_1	Умножение	$yA_i\theta A_i$, $(a\theta) * [A_i]_e \rightarrow A_i; (a\theta)$	Для чисел с плавающей запятой
<i>д</i>	Деление	$dA_i\theta A_i$	$(A_i) : [A_i]_e \rightarrow A_i; (a\theta)$	Деление	ϑ_1	Деление	$dA_i\theta A_i$, $(A_i) : [A_i]_e \rightarrow A_i; (a\theta)$	Для чисел с плавающей запятой
<i>д</i>	Деление	$dA_i\theta A_i$	$(a\theta) : [A_i]_e \rightarrow A_i; (a\theta)$	Деление	ϑ_2	Деление	$dA_i\theta A_i$, $(a\theta) : [A_i]_e \rightarrow A_i; (a\theta)$	Для чисел с плавающей запятой
<i>гн</i>	Последовательности							
Для целых чисел								
<i>с</i>	Сложение	$cA_i\theta A_i$	$(A_i) + [A_i]_e \rightarrow A_i; (a\theta)$	Матричная операция	σ	Сложение	$cA_i\theta A_i$, $(A_i) + [A_i]_e \rightarrow A_i; (a\theta)$	Для целых чисел
<i>с</i>	Сложение	$cA_i\theta A_i$	$(a\theta) + [A_i]_e \rightarrow A_i; (a\theta)$	Матричная операция	σ_1	Сложение	$cA_i\theta A_i$, $[A_i]_e - (A_i) \rightarrow A_i; (a\theta)$	Для целых чисел
<i>с</i>	Сложение	$cA_i\theta A_i$	$(A_i) - [A_i]_e \rightarrow A_i; (a\theta)$	Матричная операция	σ_2	Сложение	$cA_i\theta A_i$, $(A_i) - [A_i]_e \rightarrow A_i; (a\theta)$	Для целых чисел
<i>б</i>	Вчитывание	$bA_i\theta A_i$	$(A_i) - [A_i]_e \rightarrow A_i; (a\theta)$	Матричная операция	ϑ	Обратное вычитание	$bA_i\theta A_i$, $(A_i) * [A_i]_e \rightarrow A_i; (a\theta)$	Для целых чисел
<i>б</i>	Вчитывание	$bA_i\theta A_i$	$(a\theta) - [A_i]_e \rightarrow A_i; (a\theta)$	Матричная операция	ϑ_1	Обратное вычитание	$bA_i\theta A_i$, $(a\theta) * [A_i]_e \rightarrow A_i; (a\theta)$	Для целых чисел
<i>ум</i>	Умножение	$umA_i\theta A_i$	$(A_i) * [A_i]_e \rightarrow A_i; (a\theta)$	Умножение на единицу результата	ψ	Умножение	$umA_i\theta A_i$, $(A_i) * [A_i]_e \rightarrow A_i; (a\theta)$	Для целых чисел
<i>дм</i>	Деление	$dmA_i\theta A_i$	$(A_i) : [A_i]_e \rightarrow A_i; (a\theta)$	Деление	ϑ	Деление	$dmA_i\theta A_i$, $(A_i) : [A_i]_e \rightarrow A_i; (a\theta)$	Для целых чисел
<i>дм</i>	Деление	$dmA_i\theta A_i$	$(a\theta) : [A_i]_e \rightarrow A_i; (a\theta)$	Деление	ϑ_1	Деление	$dmA_i\theta A_i$, $(a\theta) : [A_i]_e \rightarrow A_i; (a\theta)$	Для целых чисел
<i>пн</i>	Печать	$pnA_i\theta$	$[A_i]_e \rightarrow$ целое число, переведенное в двоичную систему		ϑ_2	Печать	$pnA_i\theta$, $[A_i]_e \rightarrow$ целое число, переведенное в двоичную систему	Для целевых чисел
<i>пн</i>	Печать	$pnA_i\theta$	$[A_i]_e \rightarrow$ целое число, переведенное в двоичную систему		pl	Печать	$pnA_i\theta$, $[A_i]_e \rightarrow$ целое число, переведенное в двоичную систему	Для целевых чисел
<i>пн</i>	Для вычисления функций	$lnA_i\theta A_i$	$sin([A_i]_e) \rightarrow A_i$		cd	Сложение	$cdA_i\theta A_i$, $[A_i]_e^d + A_i^d \rightarrow A_i^d$	Для целевых чисел

Продолжение

<i>б</i>	Вчитывание	$bA_i\theta A_i$	$(A_i) - [A_i]_e \rightarrow A_i; (a\theta)$	Матричная операция	δ_1	Вчитывание	$bA_i\theta A_i$, $(A_i) - [A_i]_e \rightarrow A_i; (a\theta)$	результат по модулю 2, по нормализации
<i>б</i>	Вчитывание	$bA_i\theta A_i$	$(a\theta) - [A_i]_e \rightarrow A_i; (a\theta)$	Матричная операция	δ_2	Вчитывание	$bA_i\theta A_i$, $[A_i]_e - (A_i) \rightarrow A_i; (a\theta)$	результат по модулю 2, по нормализации
<i>у</i>	Умножение	$yA_i\theta A_i$	$(A_i) * [A_i]_e \rightarrow A_i; (a\theta)$	Умножение на единицу результата	ψ	Умножение	$yA_i\theta A_i$, $(A_i) * [A_i]_e \rightarrow A_i; (a\theta)$	результат по модулю 2, по нормализации
<i>у</i>	Умножение	$yA_i\theta A_i$	$(a\theta) * [A_i]_e \rightarrow A_i; (a\theta)$	Умножение на единицу результата	ψ_1	Умножение	$yA_i\theta A_i$, $(a\theta) * [A_i]_e \rightarrow A_i; (a\theta)$	результат по модулю 2, по нормализации
<i>д</i>	Деление	$dA_i\theta A_i$	$(A_i) : [A_i]_e \rightarrow A_i; (a\theta)$	Деление	ϑ	Деление	$dA_i\theta A_i$, $(A_i) : [A_i]_e \rightarrow A_i; (a\theta)$	результат по модулю 2, по нормализации
<i>д</i>	Деление	$dA_i\theta A_i$	$(a\theta) : [A_i]_e \rightarrow A_i; (a\theta)$	Деление	ϑ_1	Деление	$dA_i\theta A_i$, $(a\theta) : [A_i]_e \rightarrow A_i; (a\theta)$	результат по модулю 2, по нормализации
<i>гн</i>	Последовательности							
<i>с</i>	Сложение	$cA_i\theta A_i$	$(A_i) + [A_i]_e \rightarrow A_i; (a\theta)$	Матричная операция	σ	Сложение	$cA_i\theta A_i$, $(A_i) + [A_i]_e \rightarrow A_i; (a\theta)$	Для целевых чисел
<i>с</i>	Сложение	$cA_i\theta A_i$	$(a\theta) + [A_i]_e \rightarrow A_i; (a\theta)$	Матричная операция	σ_1	Сложение	$cA_i\theta A_i$, $[A_i]_e - (A_i) \rightarrow A_i; (a\theta)$	Для целевых чисел
<i>с</i>	Сложение	$cA_i\theta A_i$	$(A_i) - [A_i]_e \rightarrow A_i; (a\theta)$	Матричная операция	σ_2	Сложение	$cA_i\theta A_i$, $(A_i) - [A_i]_e \rightarrow A_i; (a\theta)$	Для целевых чисел
<i>б</i>	Вчитывание	$bA_i\theta A_i$	$(A_i) - [A_i]_e \rightarrow A_i; (a\theta)$	Матричная операция	ϑ	Вычитание	$bA_i\theta A_i$, $(A_i) * [A_i]_e \rightarrow A_i; (a\theta)$	Для целевых чисел
<i>б</i>	Вчитывание	$bA_i\theta A_i$	$(a\theta) - [A_i]_e \rightarrow A_i; (a\theta)$	Матричная операция	ϑ_1	Вычитание	$bA_i\theta A_i$, $(a\theta) * [A_i]_e \rightarrow A_i; (a\theta)$	Для целевых чисел
<i>ум</i>	Умножение	$umA_i\theta A_i$	$(A_i) * [A_i]_e \rightarrow A_i; (a\theta)$	Умножение на единицу результата	ψ	Умножение	$umA_i\theta A_i$, $(A_i) * [A_i]_e \rightarrow A_i; (a\theta)$	Для целевых чисел
<i>дм</i>	Деление	$dmA_i\theta A_i$	$(A_i) : [A_i]_e \rightarrow A_i; (a\theta)$	Деление	ϑ	Деление	$dmA_i\theta A_i$, $(A_i) : [A_i]_e \rightarrow A_i; (a\theta)$	Для целевых чисел
<i>дм</i>	Деление	$dmA_i\theta A_i$	$(a\theta) : [A_i]_e \rightarrow A_i; (a\theta)$	Деление	ϑ_1	Деление	$dmA_i\theta A_i$, $(a\theta) : [A_i]_e \rightarrow A_i; (a\theta)$	Для целевых чисел
<i>пн</i>	Печать	$pnA_i\theta$	$[A_i]_e \rightarrow$ целое число, переведенное в двоичную систему		ϑ_2	Печать	$pnA_i\theta$, $[A_i]_e \rightarrow$ целое число, переведенное в двоичную систему	Для целевых чисел
<i>пн</i>	Печать	$pnA_i\theta$	$[A_i]_e \rightarrow$ целое число, переведенное в двоичную систему		pl	Печать	$pnA_i\theta$, $[A_i]_e \rightarrow$ целое число, переведенное в двоичную систему	Для целевых чисел
<i>пн</i>	Для вычисления функций	$lnA_i\theta A_i$	$sin([A_i]_e) \rightarrow A_i$		cd	Сложение	$cdA_i\theta A_i$, $[A_i]_e^d + A_i^d \rightarrow A_i^d$	Для целевых чисел

<i>cs</i>	Вычисление $\cos x$	$\text{cs} A_1 \theta A_2$	$\cos [A_1]_e \rightarrow A_2$	Если порядок $[A_1]_e$ больше 27, то печатается $n=$ "Порядок" и останавливается	<i>бд</i>	Чтение	$b \partial A_1 \beta A_2$	$(A_1)^d - [A_1]_e^d \rightarrow A_2^d$
<i>tg</i>	Вычисление $\tan x$	$\text{tg} A_1 \theta A_2$	$\text{tg} [A_1]_e \rightarrow A_2$	При $[A_1]_e = 1$ плавающая точка и останавливается	<i>чд</i>	Умножение	$y \partial_1 \beta A_2$	$(A_1)^d \times [A_1]_e^d \rightarrow A_2^d$
<i>as</i>	Вычисление $\operatorname{atan} x$	$\operatorname{atan} [A_1]_e \rightarrow A_2$	$\operatorname{atan} [A_1]_e \rightarrow A_2$	Если $[A_1]_e > 1$, то печатается $x > j$ и останавливается	<i>дд</i>	Деление	$d \partial A_1 \beta A_2$	$(A_1)^d : [A_1]_e^d \rightarrow A_2^d$
<i>ac</i>	Вычисление $\operatorname{atctg} x$	$\operatorname{atctg} [A_1]_e \rightarrow A_2$	$\operatorname{atctg} [A_1]_e \rightarrow A_2$	Если $[A_1]_e < 1$, то печатается $x < j$ и останавливается	<i>нд</i>	Нормализация	$[A_1]_e \text{ норм}$	$[A_1]_e + 1 \text{ норм} \rightarrow A_2$
<i>at</i>	Вычисление $\operatorname{atctg} x$	$\operatorname{atctg} [A_1]_e \rightarrow A_2$	$\operatorname{atctg} [A_1]_e \rightarrow A_2$	Если $[A_1]_e > 1$, то печатается $x > j$ и останавливается	<i>дн</i>	Денормализация	$d \partial A_1 \beta A_2$	Число с плавающей запятой в виде дробного числа $\frac{[A_1]_e}{A_2}$
<i>ex</i>	Вычисление e^x	$e^x A_1 \theta A_2$	$e^x [A_1]_e \rightarrow A_2$	Если $[A_1]_e > j_2$, то печатается $x > j$ и останавливается	<i>от</i>	Отыскание	$a \partial A_1 \beta A_2$	$[A_1]^d \rightarrow A_2^d$
<i>ln</i>	Вычисление $\ln x$	$\ln A_1 \theta A_2$	$\ln [A_1]_e \rightarrow A_2$	При $[A_1]_e < 0$ печатается $x < 0$ и останавливается	<i>пд</i>	Печать	$n \partial A_1 \theta \theta$	Печать $[A_1]_e^d$ в виде десктического числа
<i>tg</i>	Вычисление $\operatorname{tg} x$	$\operatorname{tg} A_1 \theta A_2$	$\operatorname{tg} [A_1]_e \rightarrow A_2$	При $[A_1]_e < 0$ печатается $x < 0$ и останавливается				
<i>N</i>	Вычисление \sqrt{x}	$\sqrt{A_1} \theta A_2$	$\sqrt{[A_1]_e} \rightarrow A_2$	При $[A_1]_e < 0$ печатается $x < 0$ и останавливается				
Для комплексных чисел								
<i>ck</i>	Сложение	$\text{ck} A_1 \theta A_2$	$\frac{[A_1]_e^{10} + (A_1)}{(A_1)_{10}^{10} + 1) (A_2 + i)} \rightarrow A_2^d$	<i>чк</i>	Умножение	$y \times A_1 \beta A_2$	$\frac{[A_1]_e^{10} \times (A_2)}{(A_2 + i)} \rightarrow A_2$	
<i>dk</i>	Вычитание	$\delta_k A_1 \theta A_2$	$\frac{[A_1]_e^{10} - (A_1)}{(A_1)_{10}^{10} + 1) (A_2 + i)} \rightarrow A_2^d$		Деление	$d \partial A_1 \beta A_2$	$\frac{[(A_1)_{10}^{10} + 1) (A_2 + i)]}{(A_2 + i)} \rightarrow A_2$	

 $[A_1]_e = 1$ $A_1 = 1$ $A_2 = 1$ $A_2^d = 1$ $[A_1]_e^d = 1$ $A_2^d = 1$

Примеры составления программ в системе команд

I. Вычислить значения функции

$$y = \frac{ch(x-1,5)}{x}$$

для $1 \leq x \leq 3,5$, причем шаг меняется в зависимости от интервала x ;если $1 \leq x \leq 1,4$, то $h = 0,1$;если $1,4 < x \leq 2$, то $h = 0,2$;если $2 < x \leq 3,5$, то $h = 0,5$.Точность вычисления $\varepsilon = 0,0001$.Для вычисления y можно использовать рекуррентную формулу

$$chx \approx \sum_{k=0}^{\infty} U_k ; \quad U_0 = 1 ; \quad U_{k+1} = \frac{x^2 \cdot U_k}{(2k+1)(2k+2)}.$$

Блок-схема решения этой задачи представлена на рис. I.

- I00) I
 I01) 2
 I02) I,5
 I03) I,4
 I04) 3,5
 I05) 0,0001
 I06) 0,1
 I07) 0,2
 I08) 0,5

- II0) n I00 n 200 $x_0 = 1$
 I) n I00 n 201 $U_0 = 1$
 2) n 201 n 210
 3) n I00 n 202 $I \rightarrow \text{р.я.}$ (для суммы)

const
 (вводятся как числа с плавающей запятой)

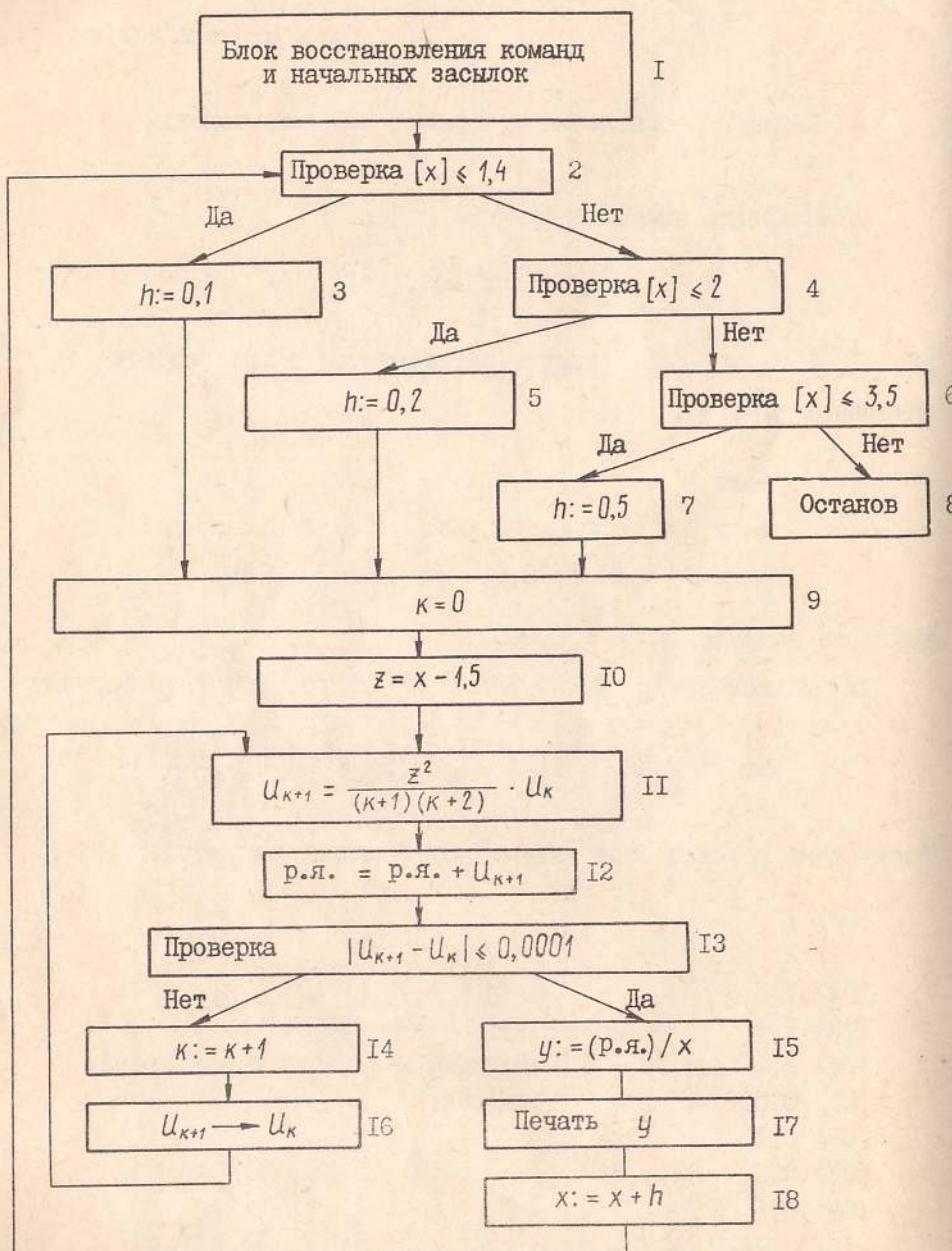


Рис. I

- 4) п 200 н 204 x_0
 5) 0пI03 н 204 I, 4 - x
 6) иII9 п <
 7) пI06 н 205 $h = 0,1$
 8) иI29 п
 9) п200 н 204
 I20) 0пI01 н 204 2 - x
 I) иI24 п <
 2) пI07 н 205 $h = 0,2$
 3) иI29 п
 4) п200 н 204
 5) 0пI04 н 204 3,5 - x
 6) иI28 п >
 7) кон
 8) пI08 н 205 $h = 0,5$
 9) п0к 203 счетчик по k
 X0
 I30) п200 н 206 $x_0 - 1,5 = z$
 I) овI02 н 206
 2) пI01 н 207
 3) уп203 н 207 2k
 4) п207 н 208
 5) спI00 н 208 $(2k + 1)$
 6) спI01 н 207 $(2k + 2)$
 7) уп208 н 207 знаменатель
 8) уп206 н 206 z^2
 9) ов207 н 206 U_{k+1} , где $k = 0,1,2 \dots$
 I40) п210 н 2II U_0
 I) уп206 н 2II
 2) сп2II н 202 $U_k \cdot U_{k+1}$
 сумма
 3) п2II н 2I2
 4) ов210 н 2I2
 5) 0мI05 н 2I2
 6) иI50 п <
 7) спI00 н 203 $U_{k+1} - U_k$
 8) п2II н 2I0 счетчик $k = k + 1$
 9) иI32 п $U_{k+1} \rightarrow U_k$

I50) $0g$ 200н 202 н y
 1) pn 202 н 3
 2) 0 2274 н
 3) sp 205 н 200 $x + h$
 4) $u II2 p$

Ячейки 200 - 212 служат рабочими для этой программы.

2. Дан массив из 50 чисел начиная с 100-й ячейки. Разбить исходный массив на два массива, так чтобы в первом были числа положительные или равные нулю, во втором - числа отрицательные. Для положительных чисел первого массива вычислить функцию $y = \sin(1+x^2)$.

Блок-схема задачи представлена на рис.2.

600) $p 0 k 14$	
1) $p 0 k 12$	счетчик 1-го массива
2) $p 0 k 11$	счетчик 2-го массива
3) $p 50 k 10$	счетчик общего массива
4) $p I4 n I$	
5) $p 100 n 400 +$	выбор числа по знаку
6) $u 616 p <$	
7) $p 400 n 401$	вычисление функции $y = \sin(1+x^2)$
8) $u p 400 n 401$	
9) $sp 633 n 401$	и печать y
610) $sn 401 n 402$	
1) $pn 402 n 6$	засылка числа во 2-й массив
2) $p II n I$	
3) $p 400 n 300 +$	
4) $c I k II$	
5) $u 619 p$	засылка числа в 1-й массив
6) $p I2 n I$	
7) $p 400 n 200 +$	
8) $c I k I2$	
9) $c 2048 k I4$	
620) $b I k 10$	
1) $u 604 p >$	
2) $p 0 k I$	

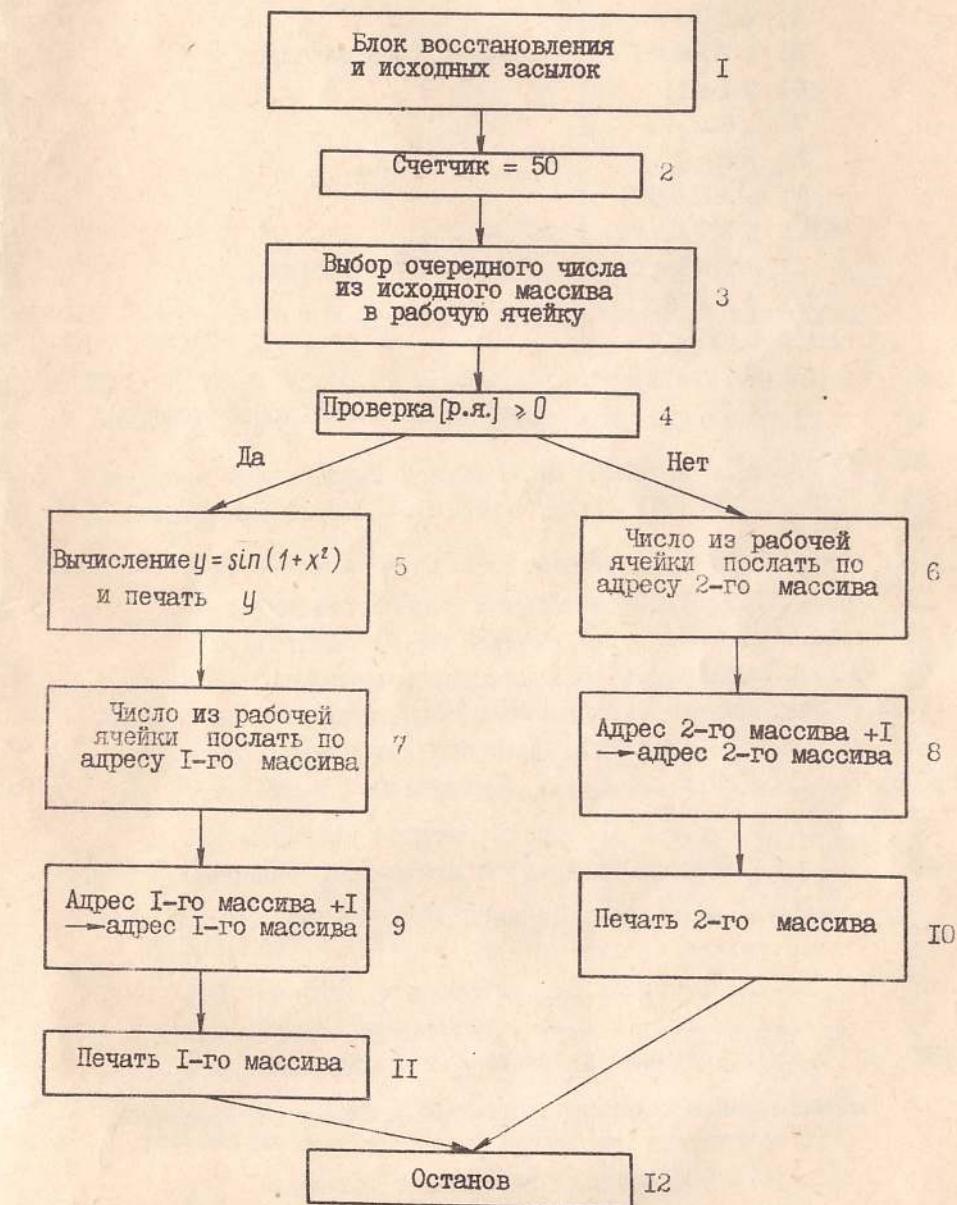


Рис.2

3) <i>пп 300 н 6 +</i>	}	печатать I-го массива
4) <i>о 2274 н</i>		
5) <i>с 2048 к I</i>		
6) <i>в I к II</i>		
7) <i>и 623 п <</i>		
8) <i>п 0 к I</i>		
9) <i>пп 200 н 6 +</i>	}	печатать 2-го массива
630) <i>о 2274 н</i>		
I) <i>с 2048 к I</i>		
632) <i>в I к I2</i>		
I) <i>и 629 п <</i>		
2) <i>к 0 н</i>		
3) <i>и</i>	(как число с плавающей запятой)	

О Г Л А В Л Е Н И Е

ПРЕДИСЛОВИЕ	5
Часть первая. УСТРОЙСТВО ЭЦВМ "НАИРИ-С"....	7
Основные технические данные ЭЦВМ "Наири-С".....	9
Лабораторная работа № I. Блок-схема ЭЦВМ "Наири-С"	II
I. Описание основных узлов машины	II
§ 1. Арифметическое устройство	II
§ 2. Устройство управления	13
§ 3. Оперативное запоминающее устройство	19
§ 4. Долговременное запоминающее устройство..	21
§ 5. Внешние устройства	22
§ 6. Пульт управления	22
II. Рабочий цикл машины	23
Описание блок-схемы операции "Сложение"	23
Лабораторная работа № 2. Внешние устройства и пульт управления	28
§ 1. Внешние устройства	28
§ 2. Пульт управления	34
§ 3. Инструкция по работе на машине	38
Лабораторная работа № 3. Долговременное запоминающее устройство	45
§ 1. Основные технические данные ДЗУ	45
§ 2. Блок-схема ДЗУ	46
§ 3. Описание функциональной схемы	46
§ 4. Цикл обращения к ДЗУ	54

Часть вторая. ПРОГРАММИРОВАНИЕ	57
I. Счетный режим	59
II. Универсальный режим	62
§ I. Автоматическое программирование	62
§ 2. Командное программирование	79
ПРИЛОЖЕНИЯ	I09
Приложение I. Внешние коды символов	III
Приложение 2. Распределение рабочих массивов памяти машины при работе <i>ан</i>	III2
Приложение 3. Исправление программы оператора "Печатаем"	III4
Приложение 4. Система команд "Наири-С"	III6
Приложение 5. Примеры составления программ в системе команд	III9

Татьяна Витальевна Марченко
 Владимир Иванович Харитонов
 Римма Садыковна Харитонова

ЭЦВМ "НАИРИ"
 УСТРОЙСТВО МАШИНЫ И ПРОГРАММИРОВАНИЕ

Ответственный за выпуск Г.С.Зотеева

ПФ 03253. Подписано к печати 21/У1-73г.
 Объем 8 п.л. Тираж 1000. Заказ 150/Ф51. Цена 28 коп.
 Ротапринт КАИ

Казань, К.Маркса, 10, Казанский авиационный институт

0268

Цена 28коп.