

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО  
ОБРАЗОВАНИЯ БССР

Брестский инженерно-строительный институт  
Кафедра высшей математики

Афонин В.Г.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

по организации вычислений на ЭЦМ "Проминь-М"  
и "Проминь-2" по курсу вычислительной техники

Брест, 1972.

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО  
ОБРАЗОВАНИЯ БССР

Брестский инженерно-строительный институт  
Кафедра высшей математики

Афонин В.Г.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

по организации вычислений на ЭЦВМ "Проминь-М"  
и "Проминь-2" по курсу вычислительной техники

Брест, 1972.

## О Г Л А В Л Е Н И Е

	Стр.
I Запоминающие устройства /ЗУ/ . . . . .	I
II Представление чисел в машине . . . . .	2
III Ввод чисел в машину . . . . .	3
IV Как проверить исправна ли машина ? . . . . .	5
У Вспомогательные, арифметические и микропрограм- мные операции, выполняемые на ЭЦМ "Проминь" . . . . .	5
VI Правила набора программ штеккерами и счета по заданной программе . . . . .	II
VII Операции передачи управления . . . . .	13
VIII Организация циклов на ЭЦМ "Проминь-М" . . . . .	14
IX Команда Чт2А в ее использование . . . . .	23
X Кодировка чисел и команд в ЭЦМ "Проминь" . . . . .	25
XI Об использовании "запрещенных" ячеек . . . . .	26
XII Дополнительные сведения о командах ЭЦМ "Проминь" Таблица № I. Система команд ЭЦМ "Проминь-М" и "Проминь-2" . . . . .	27 29

Методы вычислительной математики позволяют свести решение многих задач и выполнению определенных действий над числами, поэтому ЭЦМ являются универсальными машинами, решающими самые разнообразные задачи. Для решения любой задачи на ЭЦМ составляется программа, представляющая собой определенную последовательность команд. Каждая команда есть приказ выполнить какую-либо операцию: сложение, умножение и т.д. В целом процедура работы на ЭЦМ такова: в машину вводится программа и исходные данные, после чего машина автоматически выполняет команды программы, т.е. производит определенные действия над исходными данными. Если программа составлена верно, то задача будет решена.

Так как "Проминь-2" в принципе отличается от "Проминь-М" лишь большими вычислительными возможностями, мы будем вести изложение в основном применительно к "Проминь-М".

Малогобаритная полупроводниковая ЭЦМ "Проминь-М" предназначена для решения инженерных задач малой и средней сложности. Для хранения программы, исходных данных и промежуточных результатов вычислений в "Проминь-М", как и во всякой ЭЦМ, имеются

#### 1. ЗАПОМИНАЮЩИЕ УСТРОЙСТВА /ЗУ/.

Числовое ЗУ предназначено для хранения исходных данных и промежуточных результатов вычислений. Оно разбито на элементы, называемые ячейками памяти /всего 160 ячеек/. Каждая из ячеек имеет свой номер, называемый адресом этой ячейки. В машине имеется также сумматор, который можно считать специальной ячейкой.

Если машина включена, то можно считать, что в любой момент времени в сумматоре и в каждой ячейке памяти находится по одному-то одному числу. При записи в ячейку /или сумматор/ другого числа предыдущее содержимое ячейки /или сумматора/ автоматически стирается.

Из 160 ячеек только 79, с номерами 01-79 используются свободно.

В ячейке 00 всегда хранится число "0"; в ячейках 80-99 постоянно хранятся одни и те же числа /машинные константы/. Остальные 60 ячеек - так называемые "запрещенные" ячейки; их номера помечают звездочками.

Командное ЗУ состоит из 100 ячеек, каждая из которых имеет свой номер /00+99/ и может хранить одну команду.

ЭЦМ "Проминь-2" содержит уже два блока числового ЗУ /160+160 ячеек/. В соответствии с этим число свободных ячеек увеличивается: 001+079 /0-й блок/ и 100+179 /1-й блок/. На первом этапе знакомства с машиной рекомендуется иметь дело лишь с 0-м блоком. Командное ЗУ может поместить уже 160 команд. 0-й блок числового ЗУ идентичен /тождественен/ блоку "Проминь-М", поэтому все программы, составленные для "Проминь-М" пригодны и для "Проминь-2".

**П. ПРЕДСТАВЛЕНИЕ ЧИСЕЛ В МАШИНЕ.** Машина оперирует с числами, содержащими 5 значащих цифр. /Напомним, что значащими цифрами называют все верные цифры числа, кроме нулей, стоящих в начале. Например, в числе 0,0273 три значащих цифры, в числе 3507 - четыре, в числе  $3,7 \cdot 10^3$  - две/. При этом следует учитывать, что ввиду накопления погрешностей округления вычисления в ЭЦМ обычно ведутся с одним-двумя запасными десятичными знаками.

При выполнении большинства операций машина оперирует с десятичными числами, представленными в нормализованной форме. Любое число  $x$  в нормализованной форме записывается следующим образом:  $x = 10^p \cdot m$ ,

где  $p$  - целое число, называемое порядком числа  $x$ ,  
 $m$  - мантисса числа, причем  $0,1 \leq |m| < 1$ .

Примеры записи чисел в машине:

$$217080 = 10^6 \cdot 0,21708 \text{ записывается } +6+21708.$$

$$-0,000345 = 10^{-3} \cdot -0,345 \text{ записывается } -3-34500,$$

$$0,143346 = 10^0 \cdot 0,143346 \text{ записывается } +0+14335.$$

Правильная и быстрая запись числа в нормализованной форме достигается практикой. Можно лишь заметить, что если число по модулю больше 0,1, то порядок его будет равен числу нулей, стоящих после запятой /знак порядка "-"/; если  $0,1 \leq |x| < 1$ , то, очевидно,  $m = x$ ,  $p = 0$ ; если же число по модулю не меньше 1, то порядок будет равен числу цифр, стоящих перед запятой /знак порядка "+"/.

Максимальное по модулю число, представимое в машине -  $\pm 0,99999 \cdot 10^9$ ; если при вычислениях получается большее по модулю число, то машина останавливается и загорается лампочка "переполнение".

Минимальное /по модулю/ число  $\neq 0$ , которое можно ввести в машину есть число  $\pm 0,1 \cdot 10^{-10}$  /-0+10000/. Минимальное /по модулю/ число  $\neq 0$ , которое может получиться при вычислениях есть число  $\pm 0,1 \cdot 10^{-9}$ ; если при вычислениях получится меньшее число, то оно запишется как +0+00000 / так называемый машинный ноль/.

Нормализованная форма - основная форма представления чисел в машине и, если не оговорено противное, на всегда следует вводить в нормализованной форме.

Однако при использовании некоторых программ, требуется вводить числа в так называемой адресной форме. Число в адресной форме должно содержать не более двух цифр и записываться в 3-м и 4-м разрядах мантиссы:

$$+ \boxed{0} + \overline{00000000}$$

Например, числа 2 и 37 в адресной форме /коротко: 2A и 37A/ записываются так: +0+00200 и +0+03700.

УКАЗАНИЕ: Записывать числа, фигурирующие в упражнениях I' и I" /стр. 4/ в том виде, в котором они будут вводиться в машину.

Ш. ВВОД ЧИСЕЛ В МАШИНУ. Ввод чисел в машину и проверка содержимого ячеек памяти может осуществляться в двух режимах: "Ввод I" и "Ввод 2".

а/ В режиме "Ввод I" /адресный ввод/ можно либо записать число в одну из ячеек с номером 01+79, либо проверить содержимое любой ячейки /00+99/.

При нажатии на клавишу "Запись в ячейку" число, набранное на клавиатуре "Порядок" и "Мантисса" записывается в ячейку\*, адрес которой набран на клавиатуре "Адрес".

При нажатии кнопки "Вызов на См" содержимое ячейки, номер которой набран на клавиатуре "Адрес" вызывается на сумматор и просматривается визуально.

\* Одновременно это же число посылается в сумматор, содержимое которого высвечивается на индикаторном табло.

В обоих случаях на индикаторном табло в части "Адрес" выводится число, набранное на клавиатуре "Адрес".

Пример 1. Пусть нам надо в ячейку 38 ввести число 1972,5. Для этого на клавиатуре "Адрес" набираем число 38, на клавиатуре "Порядок" и "Мантисса" +4+19725 и нажимаем кнопку "Запись в ячейку". Глядя на индикаторное табло, убеждаемся, что запись числа произведена верно.

Пример 2. Проверим содержимое ячейки 81 /там должно храниться число  $\pi$  /. На клавиатуре "Адрес" набираем число 81 и нажимаем кнопку "Вызов на См" - индикаторное табло показывает нам, что в ячейке 81 действительно хранится число  $+1+31416=3,1416 = \pi$  .

Упражнение 1. Числа 0,28; -32,7854; 0,00381; 7A; 28A занести соответственно в ячейки 02,43,75,54,62; сделать проверку.

б/ В режиме "Ввод 2" /групповой ввод/ осуществляется ввод чисел в ячейки с номерами, расположенными подряд, например, в ячейки 11,12,13,14... Именно, при нажатии клавиши "Запись в ячейку" число, набранное на клавиатуре "Порядок" и "Мантисса" записывается в ячейку\* с адресом на 1 большим числа, высвеченного в части "Адрес" индикаторного табло.

Аналогичным образом осуществляется проверка содержимого ряда ячеек /при этом нажимает кнопку "Вызов на См"/.

Пример 3. Чтобы ввести ряд чисел, начиная, например, с 17-й ячейки, необходимо в режиме "Ввод 1" ввести первое число в 17-ю ячейку, а затем, набрав на клавиатуре следующее число и установив режим "Ввод 2", нажать кнопку "Запись в ячейку" - число запишется теперь в 18-ю ячейку; при последующем нажатии кнопки "Запись в ячейку" число с клавиатуры пойдет в 19-ю ячейку и т.д.

ЗАМЕЧАНИЕ. Если ряд чисел записывается начиная с 01-й ячейки, то достаточно нажать "Нач. сброс" и затем работать только в режиме "Ввод 2".

Упражнение 1. а/ В ячейки 20+30 занести число 1, в ячейки 31+35 занести соответственно числа 0,1; 2,3; 0,003, 0,07; 26. Сделать проверку.

б/ Проверить содержимое ячеек 80+99 /см. таблицу констант/

\* см. сброску на стр. 3.

Опишем еще назначение некоторых кнопок на пульте машины.

Клавиша БЗ - сброс клавиатуры.

Нажатие кнопки "Нач. сброс" приводит к установке устройств машины в исходное положение: в сумматор, в регистры "Адрес" и "Ъ команды" засылаются нули, вследствие чего на индикаторном табло появляются нули.

При одновременном нажатии двух кнопок "Стирание памяти" во все свободные ячейки /01+79 в "Проминь-М" и 001+079, 100+179 в "Проминь-2"/ вносятся нули: +0+00000; пользоваться этими кнопками нужно лишь в случае действительной необходимости.

IV. КАК ПРОВЕРИТЬ ИСПРАВНА ЛИ МАШИНА? Для проверки правильности работы машины служит набор тестовых программ /или просто тестов/, пробитых на перфокартах. Основным можно считать тест микропрограммной матрицы /ММ/.

#### Порядок работы с тестом ММ.

1. Вставить перфокарты теста в наборное поле.
2. Включить машину.
3. Нажать кнопку "Нач. сброс".
4. Нажать одновременно обе кнопки "Стирание памяти".
5. Нажать кнопку "Нач. сброс".
6. В ячейки с номерами 18 и 19 ввести число +1-01397.
7. В режиме "Авт" нажать кнопку "Пуск".
8. Машина должна остановиться перед командой 64, при этом на сумматоре должно быть число +1-16710.

V. ВСПОМОГАТЕЛЬНЫЕ, АРИФМЕТИЧЕСКИЕ И МИКРОПРОГРАММНЫЕ ОПЕРАЦИИ, ВЫПОЛНЯЕМЫЕ НА ЭЦВМ "ПРОМИНЬ-М". Как уже было сказано ранее, ЭЦВМ способна лишь выполнять заданные команды /операции/ по строго определенным правилам. Знание этих правил необходимо как для того, чтобы разобратся в уже составленной программе, так и для того, чтобы составить нужную программу самостоятельно.

а/ Вспомогательные операции: Чтение /ЧтА/, Запись /ЗпА/, Остановка /Ост/, Печать /Ост 01/, Формирование анал. /ФрА/ - всюду А означает адрес ячейки, участвующей в выполнении данной операции. В дальнейшем будем обозначать через /А/ содер-

6.-

химое ячейки с адресом А, а через /S/ - содержимое сумматора. Заметим, что, например, /00/ = +0+00000 /равно 0"/, /86/ = +1+10000 /равно 1/, /87/ = +1+20000 /равно 2/.

Опишем теперь работу каждой из вспомогательных операций\*

Чтение /посылка числа в сумматор/. При выполнении команды ЧТА содержимое ячейки с адресом А переносится в сумматор, что условно можно записать так: /S/ := /A / /содержимое сумматора становится равным содержимому ячейки с адресом А/. При этом содержимое ячейки с адресом А сохраняется неизменным, а содержимое сумматора, естественно, меняется.

Запись /числа из сумматора в ячейку/. При выполнении команды ЗПА происходит запись содержимого сумматора в ячейку с адресом А - /A / := /S /, при этом содержимое сумматора не меняется.

Остановка. По команде Ост машина прекращает вычисления.

Печать. По команде Ост ОI машина печатает содержимое сумматора.

Формирование знака. По команде ФРА знак мантиссы /т.е. знак числа/ в сумматоре становится равным знаку числа, хранящегося в ячейке с адресом А. В остальном содержимое сумматора не меняется. В частности, после выполнения команды Фр 00 /или просто Фр / в сумматоре образуется абсолютная величина хранящегося там ранее числа. т.к. /00/ = +0+00000

б/ Арифметические операции: сложение, вычитание /I-го и 2-го рода/ умножение, деление.

Сложение. По команде Сл А содержимое сумматора складывается с содержимым ячейки с адресом А, результат остается в сумматоре: /S/ := /S/ + /A/.

Вычитание /Выч IA/ /S/ := /S/ - /A/

Вычитание /Выч 2A/ /S/ := /A/ - /S/

Умножение /Умн A/ /S/ := /S/ \* /A/

Деление /Дел A/ /S/ := /S/ / /A/

Таким образом, при выполнении каждой из арифметических операций одно из чисел берется из сумматора - другое из ячейки;

результат от выполнения операции остается в сумматоре. При этом, естественно, содержимое ячейки остается неизменным.

Все арифметические операции машина производит в режиме с плавающей запятой, т.е. предполагается, что числа над которыми производятся действия представлены в нормализованной форме, результат также представляется в нормализованной форме.

Для примера рассмотрим, как машина производит сложение чисел. Пусть складывается два числа:

$$\begin{array}{r} 348,75 = 10^3 \cdot 0,34875 \\ 0,27401 = 10^0 \cdot 0,27401 \end{array}$$

При сложении этих чисел сначала происходит выравнивание порядков /число с меньшим порядком записывается в виде  $10^3 \cdot 0,00027/$ , а уже затем сложение:

$$10^3 / 0,34875 + 0,00027 = 10^3 \cdot 0,34902.$$

Поэтому, складывая, например, любое число с  $/80/ \Rightarrow +5 \cdot 00000$  мм в результате получаем целую часть данного числа.

Попутно отметим одну особенность команды СЛА: если порядок  $/S/ >$  порядка  $/A/$ , то при выравнивании порядков происходит округление: если же порядок  $/A/ >$  порядка  $/S/$ , то округления не происходит.

В общем случае, при выполнении арифметических операций действия производятся с округлением, результат всегда нормализуется.

ПРИМЕЧАНИЕ. При выполнении любой арифметической операции может произойти аварийный останов /Авост/ из-за переполнения разрядной сетки /образования числа, большего по модулю, чем  $10^9 \cdot 0,99999/$ . При этом на пульте загорается лампочка "Переполнение". Продолжать работу на машине в этом случае можно нажав кнопку "Нач. сброс".

Приведем примеры программ на использование вышеприведенных операций.

Пример I. Составить программу для нахождения суммы  $a + b$ , если  $a = /41/$ ,  $b = /42/$ . Очевидно, для нахождения суммы двух чисел нужно воспользоваться командой СЛА. Но сначала необходимо одно из слагаемых поместить в сумматор /это можно сделать

с помощью команды ЧтА/. В итоге, считая, что первая команда программы имеет номер 00, оставляем такую программу:

```
00 Чт 41 /S/ = /41/ = a
01 Сл 42 /S/ = /S/ + /42/ = a + l
02 Ост
```

- после выполнения программы машина остановится, результат, оставленный в сумматоре, можно будет прочитать на табло.

**Пример 2.** Составить программу для подсчета выражения  $-(x^2+2x)$  при условии, что  $x=30$ ,  $2=87$ . Записываем наше выражение в виде  $-[x(x+2)]$  и составляем такую программу:

```
00 Чт 30 /S/ = x
01 Сл 87 /S/ = x + 2
02 Умн 30 /S/ = (x+2)x
03 Выч2 00 /S/ = 0 - (x+2)x = -x(x+2)
04 Ост
```

Результат снова остается в сумматоре.

**Указание.** Проверить, что если бы мы не преобразовали  $(x^2+2x)$  в виду  $x(x+2)$ , то для подсчета этого выражения нужен было бы написать не 3, а 6 команд.

**КОНТРОЛЬНЫЕ ВОПРОСЫ:** 1. Какими из вышеприведенных команд можно изменить содержимое ячейки с адресом А7? 2. Какие из команд заведомо не меняют содержимое сумматора?

Ответы: 1. Только командой ЗпА.

2. ЗпА, Ост 01, Ост.

**б/ Микропрограммные операции.** Для выполнения каждой микропрограммной операции оставлены стандартные программы "впянные" в машину.

**Операции вычисления основных элементарных функций:**

$V^{\wedge}$ ,  $\sin$ ,  $\cos$ ,  $\tan$ ,  $\arcsin$ ,  $\arccos$ ,  $\arctan$ ,  $\ln$ ,  $\exp$ ,  $\sinh$ ,  $\cosh$ .  
Напомним, что  $\ln x = \log_e x$ ,  $\exp\{x\} = e^x$ ,  $\sinh x = (e^x - e^{-x})/2$ ,  
 $\cosh x = (e^x + e^{-x})/2$  где  $e = 2,71828\dots$ . Отметим, что по определению логарифма имеет место тождество  $x = e^{\ln x} = \exp\{\ln x\}$ . Поэтому  $a^b = \exp\{\ln a^b\} = \exp\{b \cdot \ln a\}$  - эта формула будет использована в дальнейшем.

**При вычислении элементарных функций во всех случаях аргумент берется из сумматора. Результат получается в сумматоре.**

Соответствующие команды имеют вид  $\sqrt{\quad}$ ,  $\sin$ ,  $\ln$  и т.д. - адресная часть в этих командах отсутствует.

Необходимо отметить, что при вычислении тригонометрических функций аргумент считается заданным в радианах; при вычислении обратных тригонометрических функций находятся их главные значения в радианах.

Пример 3. Составить программу для подсчета выражения:

$$\frac{|a^3| + \sqrt{\sin^2 x + a^2}}{a^2 + 2^x} = \frac{|a^3| + \sqrt{\sin^2 x + a^2}}{a^2 + \exp(x \ln 2)}; \quad (a = 41, x = 20, 2 = 187)$$

Сначала будем вычислять значение знаменателя, затем числителя /читателю предлагается обосновать это общее правило/. Далее, замечаем, что вычисленное один раз значение  $a^2$  можно будет в дальнейшем использовать еще два раза. С учетом этих замечаний составляем программу:

00 Чт	41	11 Зп	32/32/: = $\sin x$
01 Умн	41	12 Умн	32
02 Зп	30/30/: = $a^2$	13 Сл	30
03 Чт	87	14 $\sqrt{\quad}$	
04 $\ln$	/S/: = $\ln/S/ = \ln 2$	15 Зп	33/33/: = $\sqrt{\sin^2 x + a^2}$
05 Умн	20	16 Чт	30
06 $\exp$	/S/: = $2^x$	17 Умн	41
07 Сл	30	18 Фр	00
08 Зп	31/31/: = $a^2 + 2^x$	19 Сл	33
09 Чт	20	20 Дел	31
10 $\mu$		21 Ост	

В этой программе отсутствует столь подробный комментарий, как в первых двух, т.к. содержимое сумматора меняется многократно в процессе вычислений. В основном указано лишь, в каких ячейках хранятся нужные нам промежуточные результаты.

ПРИМЕЧАНИЕ. Мы начинали составлять наши программы

с команды под № 00, хотя могли бы начинать с команды под любым другим номером.

ОБЩИЕ УКАЗАНИЯ: а/ в программах для подсчета алгебраических выражений последовательность команд соответствует последовательности действий, выполняемых согласно известным правилам арифметики /сначала вычисление значений элементарных функций, затем возведение в степень, умножение, деление, сложение, вычитание/.



## II.-

Ответ располагается в ячейках, начиная с  $n(n+1)+1$ , т.е. непосредственно после ячеек, где находились коэффициенты системы.

Отметим еще, что при использовании операций РС и  $\bar{a}$  ячейки 7I+76 заняты.

**VI. ПРАВИЛА НАБОРА ПРОГРАММ ШТЕККЕРАМИ И СЧЕТА ПО ЗАДАННОЙ ПРОГРАММЕ.** Команды следует набирать на наборном поле в соответствии с их написанием. Коды /символы/ операций набираются штеккерами с белыми головками; адреса набираются штеккерами с черными головками. Лишь при наборе адреса "запрещенной" ячейки в разряд единиц необходимо вставить соответствующий штеккер с красной /оранжевой/ головкой.

Команды для выполнения микропрограммных операций, а также команда Ост считаются безадресными, т.е. набираются одним штеккером в виде  $\sqrt{\quad}$ ,  $\text{мл}\sqrt{\quad}$ ,  $\text{ост}\sqrt{\quad}$ ,  $\text{РС}\sqrt{\quad}$ . Знак  $\sqrt{\quad}$  означает, что соответствующая позиция пропускается.

Опишем некоторые особенности набора команд:

I. В машине "Проминь-М". Штеккер с цифрой "0" не используется: там ~~жж~~, где должен стоять этот штеккер, пропускается соответствующая позиция.

II. В машине "Проминь-2". а/ Чтобы набрать штеккерами адрес ячейки из 0-го блока, необходимо в разряде десятков поставить штеккер с цифрой, имеющей штрих, или, если число десятков равно нулю, пропустить соответствующую позицию.

Например, команда Чт 04I должна быть набрана в виде Чт 4'I /либо Чт 4'I'/; команда Зп 007 - в виде Зп  $\sqrt{\quad}$  7/либо Зп  $\sqrt{\quad}$  7'/; команда Чт 000 - в виде Чт  $\sqrt{\quad}$   $\sqrt{\quad}$ .

Для числовых ячеек I-го блока в разряде десятков помещают штеккер "без штриха". Например, команда Сл 150 набирается в виде Сл 5  $\sqrt{\quad}$ ; команда Дел 106 в виде Дел 06 /либо Дел 06'/.

Таким образом, штеккеры с цифрой "0" используются лишь для набора адресов ячеек из диапазона I00+I09 /либо для набора номеров команд из диапазона I00+I09: см. ниже/.

б/ Если в одной из команд передачи управления /см.далее раздел УП/ нам нужно набрать номер команды из диапазона от 00 до 99, то это осуществляется так же, как при наборе адреса числовой ячейки 0-го блока; номера команд от 100 до 159 набираются так же, как адреса ячеек 1-го блока.

Чтобы машина произвела вычисления по программе, которая начинается с команды под номером 00, следует нажать кнопку "Нач.сброс", установить режим "Авт" и нажать кнопку "Пуск"; машина последовательно выполнит команды программы. Если же программа начинается с команды под другим номером, то на клавиатуре "Номер команды" следует набрать нужный номер и нажать правую кнопку "Стирание памяти" - в регистр "№ команды" занесется набранное число и при нажатии кнопки "Пуск" машина начнет вычисления с нужной команды.

При поиске ошибок в программе /т.е. при отладке программы/ часто используется режим "Такт". В этом режиме при однократном нажатии кнопки "Пуск" машина выполняет лишь одну очередную команду.

Обычно машина прекращает вычисления по команде 0ст /или Сст 01, если цифропечатающее устройство отключено/. Прекратить вычисления можно также нажав кнопку "0ст" или "Нач.сброс". Но иногда бывает необходимо, чтобы машина остановила вычисления перед командой с данным номером. Для этого набирают нужный номер на клавиатуре "№ команды" и нажимают кнопку "Вкл". Во всех случаях, когда машина прекращает вычисления, на индикаторном табло высвечивается номер команды, перед выполнением которой машина "остановилась".

В заключение укажем еще, что режимы К1 и К2 используются лишь при поиске неисправностей в машине.

Упражнение 3. Решить на машине /командой РС/ систему 4-го порядка

$$2x_1 + x_2 - 5x_3 + x_4 = 8$$

$$x_1 - 3x_2 \quad \quad -6x_4 = 9$$

$$2x_2 - x_3 + 2x_4 = -5$$

$$x_1 + 4x_2 - 7x_3 + 6x_4 = 0$$

решение которой:  $x_1 = 3$ ,  $x_2 = -4$ ,  $x_3 = -1$ ,  $x_4 = 1$ .  
Проверку решения произвести с помощью операции  $\bar{a} \cdot \bar{b}$ .

УП. ОПЕРАЦИИ ПЕРЕДАЧИ УПРАВЛЕНИЯ. РАЗВЕТВЛЯЮЩИЕСЯ ПРОГ-  
РАММЫ. До сих пор мы встречались с программами, у которых  
очередность выполнения команд соответствует их нумерации, т.е.  
команды выполняются в "естественной" последовательности. Часто,  
однако, приходится сталкиваться с ситуацией, когда необходимо  
нарушать "естественный" порядок выполнения команд в программе.

В связи с этим во всякой ЭЦМ предусмотрены так называемые  
операции передачи управления. Таких основных операций в "Про-  
минь-М" три: Уп1  $\alpha$  /условная передача 1/, Уп2  $\alpha$  , Бп  $\alpha$  /без-  
условная передача/. Здесь  $\alpha$  означает номер нехоторой команды  
в программе.

Операции Уп1  $\alpha$  и Уп2  $\alpha$  обычно используются в случае, ког-  
да в зависимости от выполнения /или невыполнения/ того или ино-  
го условия, машина должна выполнять ту или иную последователь-  
ность действий. Соответствующие программы называются развет-  
вляющимися.

При выполнении команды Уп1  $\alpha$  машина "смотрит" знак числа в  
сумматоре; если знак "-", то следующей будет выполняться коман-  
да под номером  $\alpha$  , в противном случае /когда  $|S| \geq 0$  / будет  
выполняться команда, следующая по номеру за командой Уп1  $\alpha$  .

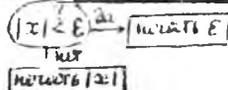
При выполнении команды Уп2  $\alpha$  , если  $|S| = 0$  , то следующей  
будет выполняться команда под номером  $\alpha$  , в противном случае  
(когда  $|S| \neq 0$  ) будет выполняться команда, следующая по номеру  
за Уп2  $\alpha$  .

После выполнения команды Бп  $\alpha$  следующей будет выполнена  
команда под номером  $\alpha$  .

Содержимое сумматора при выполнении команд передач управ-  
ления не меняется.

Пример. Составить программу для нахождения и печати боль-  
шего из 2-х чисел:  $|x|$  и  $\varepsilon$  , предполагая, что  $x = /54/$ ,  $\varepsilon = /55/$ .

Решение. Составляем сначала блок-схему программы:



Наша программа имеет вид

00	Чт	54		→	05	Чт	56	} печать  x	
01	Фр	00			06	Ост	01		
02	Фр	56	/54/ =  x		07	Ост			
03	Вч	I 55	} ( x  - ε) ?		→	08	Чт	55	} печать ε
04	Уп	I 08				09	Ост	01	
						10	Ост		

Заметим, что при составлении программы мы писали 04 Уп1... затем рассматривали случай когда  $|S|=|x|-\varepsilon > 0$  ( $|x| > \varepsilon$ ), после чего писали команды 05,06,07 /печатать  $|x|$  / и только после этого дописывали: 04, Уп1 08. Этот прием является общим при использовании команд Уп1  $\mathcal{L}$  и Уп2  $\mathcal{L}$ .

Помечание. При любых значениях  $x$  и  $\varepsilon$  машина не выполняет три команды /либо с номерами 05-07, либо с номерами 08+10/.

Упражнение 4. Сократить данную программу на одну команду /см. команды 06,07 и команды 09,10/, воспользовавшись операцией Бп  $\mathcal{L}$ .

УШ. ОРГАНИЗАЦИЯ ЦИКЛОВ НА ЭЦМ "ПРОМИнь". Использование ЭЦМ особенно эффективно в случаях, когда одна и та же последовательность действий повторяется машиной многократно /циклически/. При этом в программе определенная последовательность команд должна выполняться многократно; такие программы называют циклическими.

Центральными командами в циклических программах можно считать те, которые обеспечивают нужное число повторений упомянутой последовательности команд /нужное число повторений "цикла"/. Обычно это команды условной передачи управления /Уп1  $\mathcal{L}$  и Уп2  $\mathcal{L}$  / в сочетании с другими командами.

Различают три основных вида циклических программ /или коротко, циклов/: простые, итерационные и циклы с переадресацией. Используются также вложенные циклы /цикл в цикле/.

Перейдем к рассмотрению примеров на составление циклических программ.

#### 1. Простые или арифметические циклы.

Вычислить сумму  $1^3 + 2^3 + 3^3 + \dots + 20^3 = \sum_{k=1}^{20} k^3$ .

При составлении программы для вычисления суммы обычно выбирают ячейку "суммирования", содержимое которой будем обозначать буквой "с". Программа должна сперва поместить в эту ячейку число 0 /т.е. выполнить операцию  $s:=0$ /, а затем нужное число раз менять содержимое этой ячейки по формуле  $s:=s+$  очередное вычисленное слагаемое нашей суммы. В результате таких операций в ячейке суммирования сначала скажется

первое слагаемое, затем сумма первых двух слагаемых, сумма первых трех слагаемых и т.д. - говорят, что в ячейке суммирования накапливается нужная нам сумма.

Составим теперь блок-схему решения задачи



Значения  $C$  и  $K$  меняются в процессе вычислений: будем поэтому  $C$  и  $K$  называть переменными величинами или просто переменными.

Предлагаем читателю убедиться в том, что наша блок-схема действительно реализует задачу вычисления  $\sum_{k=1}^{21} k^3$ . В самом деле, сначала  $C=0$  и  $K=0$ . Затем  $K:=0+1=1$ ; очевидно,  $K \neq 21$  /"нет"/ и выполняется операция  $C:=0+1^3=1^3$  /1-й "цикл"/. После этого  $K:=1+1=2$ ;  $2 \neq 21$  /"нет"/  $\rightarrow C:=1^3+2^3$  /2-й "цикл"/. Затем  $K:=2+1=3$ ;  $3 \neq 21$  /"нет"/  $\rightarrow C:=1^3+2^3+3^3$  /3-й "цикл"/ и т.д. Наконец  $C:=C+20^3=1^3+2^3+\dots+20^3$ , затем  $K:=20+1=21$ ;  $21=21$  /"да"/ печать  $C$ /для  $K=21$  суммирование уже производить не нужно/.

Теперь остается написать программу, команды которой последовательно реализуют блоки нашей блок-схемы.

Произведем сначала распределение машинной памяти.

Пусть  $C=50$ ,  $21=51$ ,  $K=52$ ,  $I=86$ .

Начальный ввод:  $21 \rightarrow 51$  /число 21 вводится в ячейку 51/.

#### ПРОГРАММА I

00	Чт	00	$C = K = 0$ $K = K + 1$ $(K - 21) \neq 0$	08	Чт	52	$C = C + K^3$
01	Зп	52		09	Умн	52	
02	Зп	50		10	Умн	52	
03	Чт	52		11	Сл	50	
04	Сл	86		12	Зп	50	
05	Зп	52		13	Бп	03	
06	Выч I	51		14	Чт	50	
07	Уп2	14		15	Ост		

Итак программа без всяких изменений пригодна для вычисления

$\sum_{k=1}^{n+1} k^3$  - нужно только в ячейку 51 ввести число  $n+1$ .

Далее, нашу блок-схему легко переделать для вычисления

$\sum_{k=1}^{n+1} f(k)$ , где  $f(k)$  - произвольная функция. Для этого в блоке суммирования вместо  $k^3$  надо написать  $f(k)$  и число 21 заменить на  $n+1$ .

Ниже приводится программа для вычисления  $\sum_{k=1}^n f(k)$ , где  $f(k)$  - произвольная функция. Такие программы называют стандартными, т.к. они решают целый класс задач.

Пусть  $c=177$ ,  $n+1=178$ ,  $n=179$

ПРОГРАММА IC

00	Чт	00	} $c := k := 0;$	→ 08	Бп	14	переход к
01	Зп	77					
02	Зп	79	} $k = k + 1;$	09	Сл	77	} $c = c + f(k)$
03	Чт	79			10	Зп	
04	Сл	86	} $(k - (n + 1) = 0)$	11	Бп	08	} $\left. \begin{array}{l} \\ \end{array} \right\} \text{начало } c$
05	Зп	79			12	Чт	
06	ВычI	78		13	Ост		
07	УпI	12					

Инструкция к программе IC. Начальный ввод:  $n+1 \rightarrow 78$ .

Подпрограмма для вычисления  $f(k)$  должна начинаться с 14-й команды. При составлении этой подпрограммы следует знать, что "к" находится на сумматоре и в 79-й ячейке. Вычисленное значение  $f(k)$  должно остаться на сумматоре; в конце подпрограммы должна стоять команда Бп 09. Программа использует ячейки 77+79.

Упражнение 6. Составить программу для подсчета  $\sum_{k=1}^n (a-k)^3$ , где "а" задано.

Указание. Здесь можно воспользоваться программой IC, положив  $f(k) = (a-k)^3$ .

2. Итерационные циклы встречаются в задачах, где число повторений цикла заранее не известно и определяется в процессе вычислений. Типичным примером итерационного цикла является реализация метода Ньютона для приближенного решения нелинейных уравнений вида,  $f(x) = 0$ . (\*)

Для отыскания определенного корня уравнения (\*) сначала выбирают некоторое значение  $x_0$ , близкое к этому корню.  $x_0$  называется начальным приближением: в ряде случаев его можно отыскать графически/. Желательно еще, чтобы  $x_0$  было таким, что  $f(x_0) f'(x_0) > 0$

Затем образуют последовательность  $x_0, x_1, \dots, x_n, \dots$  по формулам  $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$ ,  $x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$ , ...,  $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ , ... такую последовательность называют итерационным процессом. При условии, что  $x_0$  выбрано достаточно близким к корню уравнения (\*), указанная последовательность сходится к этому корню.

Вычисления обычно прекращают, когда  $|x_{n+1} - x_n|$  не превосходит заданного  $\varepsilon > 0$  и  $x_{n+1}$  принимают за приближенное значение корня. Для примера рассмотрим задачу нахождения наибольшего корня уравнения  $\sin x - 0,1e^x = 0$  / здесь  $f(x) = \sin x - 0,1e^x$  /. Для нахождения начального приближения  $x_0$  перепишем уравнение в виде  $\sin x = 0,1e^x$  и построим приближенно графики функций  $y = \sin x$  и  $y = 0,1e^x$ . Очевидно, абсциссы точек пересечения графиков будут давать приближенные значения значений корней нашего уравнения.

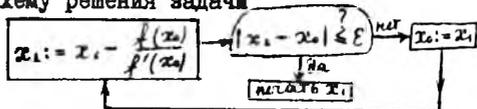


Из построенных графиков легко видеть, что искомым корнем лежит между  $\frac{\pi}{2}$  и  $\pi$  /  $\sin \frac{\pi}{2} = 1, e^{\frac{\pi}{2}} < 3^{\frac{\pi}{2}} < 3^2 = 9$ , т.е.  $\sin \frac{\pi}{2} > 0,1e^{\frac{\pi}{2}}$ , но  $\sin \pi = 0 < 0,1e^{\pi}$  /.

Возьмем в качестве  $x_0$  число 2,3 и проверим выполнение неравенства  $f(x_0) \cdot f''(x_0) > 0$ . Пользуясь математическими таблицами, находим  $f(2,3) = \sin 2,3 - 0,1e^{2,3} = 0,75 - 0,1 \cdot 10 < 0$ .

$f''(2,3) = -\sin 2,3 - 0,1e^{2,3} < 0$ , т.е.  $f(2,3) \cdot f''(2,3) > 0$ .

Составляем блок-схему решения задачи



Замечание. В итерационном процессе Ньютона для вычисления следующего приближения  $(x_{n+1})$  необходимо знать лишь одно предыдущее  $(x_n)$ . Поэтому в блок-схеме мы вводим всего две переменные, обозначая их  $x_0$  и  $x_1$ .

Распределение памяти:  $x_0 = /70/$ ,  $x_1 = /71/$ ,  $\varepsilon = /72/$ ,  $0,1 = /914^*/$ . Начальный ввод  $x_0 \rightarrow 70$ ,  $\varepsilon \rightarrow 72$ .

Имея в виду, что  $f'(x) = \cos x - 0,1e^x$ , составляем

ПРОГРАММУ 2.

00	Чт	70	06	Выч I	01	
01	вхр		07	Зп	02	/02/ := f'(x0)
02	Умн	914*	08	Чт	70	
03	Зп	01 /01/ := 0,1e^x	09	Умн		
04	Чт	70	10	Выч I	01	
05	сч		11	Дел	02	

I2	Выч2	70		Чт	71	
I3	Зп	71	$f(x) = x_0$	Ост		"печать" $x_1$
I4	Выч1	70	$f'(x) = x_0$	Чт	71	
I5	Фр	00	$f''(x) = x_0$	Зп	70	
I6	Выч2	72	$f'''(x) = x_0$	Бп	00	
I7	Уп I	20	$\epsilon -  x_i - x_{i-1}  > 0$			

нет

Упражнение 7. Вычислить методом Ньютона наибольший корень уравнения  $x^2 - \sin 5x = 0$ , положив  $\epsilon = 10^{-3}$ .

3. Циклы с переадресацией встречаются, например, в задачах, где над каждым из ряда чисел, расположенных в памяти машины в естественной последовательности, нужно произвести одинаковые действия.

Прежде чем переходить к рассмотрению циклов с переадресацией опишем еще несколько команд машины "Проминь-М": ЧтПА, СчПА, СтФА, ВычФА.

Команда ЧтПА /чтение второго ранга/. Пусть в ячейке с адресом А находится число  $+0+03658$  /на месте символа может стоять любая цифра/. Тогда, при выполнении команды ЧтПА в сумматор занесется число, записанное в ячейке с адресом А. Пусть, например, в ячейке 74 хранится число  $+0-03658$  /или  $+0+03673$ , или  $+0+03600$ /. Тогда, при выполнении команды ЧтП74 в сумматор занесется число, находящееся в 36-й ячейке. На самом деле возможности команды ЧтПА шире, чем здесь описано /см. IX/.

Примечание. Если в машине "Проминь-2" в 5-м разряде ячейки А стоит 0, то при выполнении команды ЧтПА происходит обращение к одной из ячеек 0-го блока; если же в 5-м разряде стоит 1, происходит обращение к ячейке 1-го блока.

Команда СчПА /счет циклов и переадресация/.

При выполнении этой команды машина увеличивает на единицу каждое из двух чисел, одно из которых помещается в 3-м и 4-м, другое - 1-м и 2-м разрядах мантиссы ячейки А. Изменение таким образом содержимое ячейки А будет находиться в ячейке А и в сумматоре /т.е. при выполнении команды СчПА меняется содержимое ячейки А и сумматора/. Так происходит во всех случаях, кроме одного: когда при выполнении команды СчПА в 1-м и 2-м разрядах ячейки А находится число 99. В этом случае при выполнении команды СчПА, после упомянутого уве-

личение на единицу двух чисел /в 1-м и 2-м разрядах теперь будут нули/ в сумматор посылается число со знаком "+".

Примеры. Пусть  $/51/ = +0-03564$ ,  $/52/ = +1+06833$ ,  $/53/ = +0-10799$   
 При выполнении команды СчП51 /S/ := /51/ := +0-03665

- " - " - СчП52 /S/ := /52/ := +1+04934

- " - " - СчП53 /S/ := +0+10800 ( /53/ := +0-10800)

Команда СчПА может быть успешно использована как для счета "циклов" в программе, так и для переадресации, т.е. для увеличения на единицу числа, хранящегося в 3-м и 4-м разрядах ячейки А.

Команды СлФА и ВычФА сложение и вычитание с фиксированной запятой. При выполнении этих команд мантисы чисел рассматриваются независимо от порядков и действия над мантиссами производятся как над целыми числами. По команде СлФА содержимое сумматора складывается /в режиме с фиксированной запятой/ с содержимым ячейки А, результат остается в сумматоре:  $/S/ := /S/ + /A/$   
 По команде ВычФА  $/S/ := /S/ - /A/$ .

Относительно порядка суммы /разности/ заметим, что если у одного из чисел:  $/A/$  или  $/S/$  порядок  $+ 0$ , то при выполнении команды Сложение ФА /ВычФА/ порядок суммы /разности/ будет равен порядку другого числа. Пример.

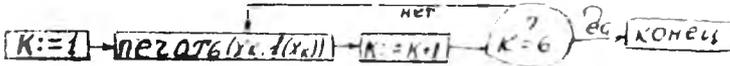
Пусть  $/10/ = +1-08454$ ,  $/S/ = +0+02203$ .

После выполнения команды СлФ10  $/S/ := +1-06451$ . Если же вместо СлФ10 выполнить ВычФ10, то  $/S/ := +1+10657$ . Разумеется, при выполнении операции с фиксированной запятой никакого округления чисел и нормализации результата не происходит.

Пусть теперь нам нужно составить программу для вычисления и печати  $f(x_k) = \sqrt{(x_k - a)^2 + b^2}$  при заданных значениях  $a$ ,  $b$ ,  $x_1$ ,  $x_2$ , ...,  $x_5$ . Геометрически  $f(x_k)$  равно расстоянию от точки с координатами  $(a, b)$  до точки с координатами  $(x_k, 0)$ . Перед значениями функции будем печатать соответствующие значения аргумента.

С появлением команды СчПА в ранних моделях ЭВМ "Проминь" эта команда отсутствовала, команды СлФА и ВычФА стали использоваться значительно реже. В дальнейшем мы не будем пользоваться операциями с фиксированной запятой.

## Блок-схема решения задачи:



Начальный ввод:  $Q \rightarrow 75$ ,  $\theta \rightarrow 76$ ,  $x_1 \rightarrow 01$ ,  $x_2 \rightarrow 02, \dots$ ,  $x_5 \rightarrow 05$ .  
 Для хранения переменной  $K$  отведем ячейку 78, а в ячейку 79 поместим число  $+0-00195$ : в 3-м и 4-м разрядах ячейки 79 помещается адрес  $X_I$  /начальный адрес/, а в 4-м и 5-м разрядах число  $100-5$  /у нас 5 чисел  $x_1, x_2, \dots, x_5$ /.

## ПРОГРАММА 3.

00	ЧТ	79	$k:=1$	08	Чт	76	нет
01	Зп	78		09	Умн	76	
02	ЧТП	78	$/S/ := X_k$	10	Сд	72	
03	Ост	01	печать( $x_k$ )	11	V		
04	Выч1	75		12	Ост	01	печать( $f(x_k)$ )
05	Зп	71/71/	$:= X_k - a$	13	СчП	78	
06	Умн	71		14	Уп1	02	$k:=k+1$ → $k:=6$
07	Зп	72/72/	$:= (x_k - a)^2$	15	Ост		да

Рассмотрим более подробно работу ПРОГРАММЫ 3. Для этого проследим, какие действия производят команды 02+14, каждая из которых выполняется при работе программы многократно /5 раз/.

1-й "цикл": при выполнении команды ЧТП78 машина выполняет команду Чт01, т.е.  $/S/ := X_k$  коротко: ЧТП78  $\Rightarrow$   $/S/ := X_k$ ; по команде 03 Ост 01 печатается  $x_k$ , а по команде 12 Ост 01 печатается  $f(x_k)$ ;  $/S/ := /78/ := +0-00296$ ; 13 Уп1 02  $\Rightarrow$  переход к команде 02, т.к.  $/S/ < 0$ .

2-й, 3-й и 4-й "циклы" выполняются аналогично 1-му "циклу"; их подробное рассмотрение предоставляется читателю. Отметим лишь, что после выполнения 4-го "цикла" будет написано  $f(x_4)$  и  $/78/ := +0-00599$ .

5-й "цикл". ЧТП78  $\Rightarrow$   $/S/ := x_5$ ; СчП78  $\Rightarrow$   $/S/ := +0+00600$  и после команды 13 Уп1 02 выполняется команда под номером 14, т.к.  $/S/ > 0$ .

Примечание 1. Легко видеть, что если бы числа  $x_1, x_2, \dots, x_5$  помещались бы не с 01-й, а, например, с 23-й ячейки, то в ячейку 79 перед началом работы программы следовало бы занести число  $+0-02395$ ; сама программа несколько не изменилась бы.

Далее, если бы у нас было не 5, а  $n$  чисел  $x_1, x_2, \dots, x_n$ , то в 1-й и 2-й разряды мантиссы вместо 95 следовало бы поместить число 100-и.

Примечание 2. Вообще говоря, в последней программе можно было не писать команды 00 и 01; тогда команда I3 имела бы вид I3 СчП79 и содержимое ячейки 79 "портилось" бы в процессе работы программы /командой СчП79/. При этом наша программа была бы несамовосстанавливающейся, т.е., если программа проработает один раз, то, чтобы пропустить ее второй раз /изменив, например, значение  $x_1$ /, нам потребовалось бы заново вводить в ячейку 79 число +0-00I95.

Упражнение 7. Составить программу для подсчета  $\sum_{k=1}^n \sqrt{x_k^2 + b^2}$ , считая  $x_1, x_2, \dots, x_n, b$  известными.

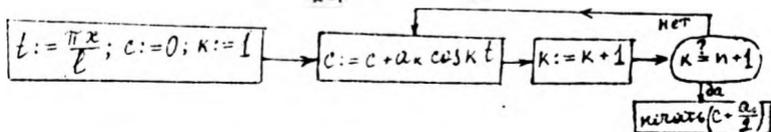
#### 4. ВЛОЖЕННЫЕ ЦИКЛЫ / с использованием символических адресов/.

Нередко приходится составлять программу, которая является частью другой, более сложной программы. При этом сразу произвести распределение памяти и нумерацию команд бывает затруднительно - в таких случаях прибегают к программированию в символических адресах. Будем пользоваться обозначением: адрес всякого числа  $x$  будем записывать в виде  $[x]$ , например,  $[I]=86, [\pi]=8I$  и т.д. (Используя прежние обозначения, имеем  $[x] \equiv x$ ).

Поставим теперь задачу: вычислить и напечатать таблицу значений функции  $f(x)$  на отрезке  $[a, b]$  с шагом  $h$ , /  $h = \frac{b-a}{n}$ , где  $n$  - целое/, т.е. напечатать  $f(a), f(a+h), f(a+2h), \dots, f(a+nh) = f(b)$ . В качестве  $f(x)$  взять  $\frac{a_0}{2} + \sum_{k=1}^n a_k \cos k \frac{\pi x}{l}$  /отрезок ряда Фурье "по косинусам"/; числа  $a_0, a_1, \dots, a_n$ , а также  $l, a, b, h$  считать заданными.

Будем считать, что числа  $a_1, a_2, \dots, a_n$  расположены в естественной последовательности, начиная с ячейки 01. Составим сначала блок-схему и программу вычисления и печати

$$\frac{a_0}{2} + \sum_{k=1}^n a_k \cos k \frac{\pi x}{l} \quad \text{при заданном } x:$$



Пусть  $A_0 = +0-001 \frac{100-n}{2}$ .

(  $0 = /00/$ ,  $I = /86/$  ).

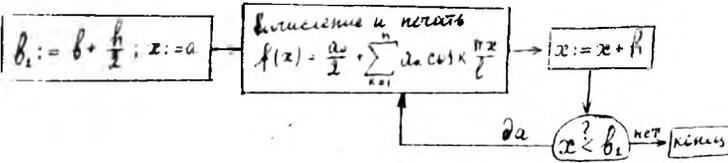
ПРОГРАММА 4<sup>0</sup>.

$\alpha$ +1 Чт  $[\pi]$   
 $\alpha$ +2 УМН  $[x]$   $t := \frac{\pi x}{2}$   
 $\alpha$ +3 Дел  $[t]$   
 $\alpha$ +4 Зп  $[t]$   
 $\alpha$ +5 Чт  $00$   $c := 0$   
 $\alpha$ +6 Зп  $[c]$   
 $\alpha$ +7 Чт  $86$   
 $\alpha$ +8 Зп  $[k]$   $k := 1$   
 $\alpha$ +9 Чт  $A_0$   
 $\alpha$ +10 Зп  $B_0$   
 $\alpha$ +11 ЧтП  $\beta$   $\beta_1 / \beta_0 := a$   
 $\alpha$ +12 Зп  $B_1$   
 $\alpha$ +13 Чт  $[t]$   
 $\alpha$ +14 УМН  $[k]$

$\alpha$ +14  $c := 1$   
 $\alpha$ +15 УМН  $\beta_1$   
 $\alpha$ +16 Сл  $[c]$   
 $\alpha$ +17 Зп  $[c]$   $c := c + \sin \cos kt$   
 $\alpha$ +18 Чт  $[k]$   
 $\alpha$ +19 Сл  $86$   $k := k + 1$   
 $\alpha$ +20 Зп  $[k]$   
 $\alpha$ +21 СчП  $B_0$   
 $\alpha$ +22 УП1  $\alpha$ +10  $(c - 1) \cdot \frac{1}{2} \cdot n \cdot c$   
 $\alpha$ +23 Чт  $[a]$   
 $\alpha$ +24 УМН  $[t]$  печать  $(c + \frac{2c}{3})$   
 $\alpha$ +25 Сл  $[c]$   
 $\alpha$ +26 Ост  $01$

Очевидно, программа самовосстанавливающаяся / команды восстановления  $\alpha$ +8,  $\alpha$ +9/.

Выпишем теперь блок-схему всей задачи



Пояснение. Мы производим сравнение  $x$  не с  $\beta$  а с  $\beta_1$ , т.к. часто  $\beta$  и  $x + \beta$  вычисляются с округлением и переменная  $x$  вообще говоря, может никогда не стать в точности равной  $\beta$ ; она может быть "чуть большей" или "чуть меньшей"  $\beta$ .

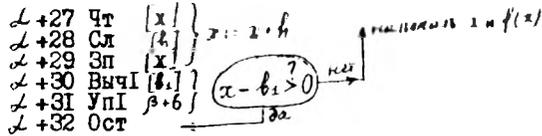
Пример.  $[a, \beta]$  есть  $[0, 1]$ ,  $n = 3$ ,  $\beta_1 = \frac{1}{3} = 0,33333$ ; очевидно,  $n \cdot \beta = 3 \cdot \beta = 3 \cdot 0,33333 < 1$ .

Составим программу по нашей блок-схеме.

$\beta$  Чт  $[h]$   
 $\beta$ +1 УМН  $[1/2]$   $\beta_1 = \beta + \frac{\beta}{2}$   
 $\beta$ +2 Сл  $[h]$   
 $\beta$ +3 Зп  $[h_1]$   
 $\beta$ +4 Чт  $[a]$   $x := a$   
 $\beta$ +5 Зп  $[x]$   
 $\beta$ +6 Чт  $[x]$  печать  $x$   
 $\beta$ +7 Ост  $01$

Далее должна следовать программа 4<sup>0</sup>, откуда заключает, что  $\beta$ +7 =  $\alpha$ -1, т.е.  $\beta = \alpha$ -8 и, значит, можно взять  $\alpha = 8$ , тогда вся программа будет начинаться с команды под номером 00.

После программы / точнее, уже подпрограммы / 4<sup>0</sup> будут идти команды:



Ясно, что здесь команды восстановления L+8 и L+9 будут играть существенную роль, т.к. обращение к подпрограмме 4<sup>0</sup> происходит неоднократно.

Наша программа будет оформлена окончательно, если символическим адресам придать числовые значения, например [π] = 8I, [x] = 79, [l] = 78 и т.д. - числа 8I, 79, 78... называют действительными адресами. Отметим еще, что ячейки β<sub>0</sub>, β<sub>1</sub>, [c] ..., где хранятся промежуточные результаты, называют рабочими ячейками.

**IX. КОМАНДА ЧтПА И ЕЕ ИСПОЛЬЗОВАНИЕ.** Ранее было уже дано предварительное представление об операции ЧтПА. Опишем теперь более полно возможности этой команды.

По команде ЧтПА машина выполняет команду, записанную в ячейке A.

Команды в числовых ячейках памяти располагаются следующим образом\*. Разряд порядка определяет код операции, 3-й и 4-й разряды мантиссы - адрес.

В машине "Кроминь-2" адрес определяется еще 5-м разрядом мантиссы: если там стоит 0, то мы будем иметь дело с ячейкой 0-го блока; если 1 - с ячейкой 1-го блока.

Аналогичные замечания можно сделать относительно команд ЗпПА и БпПА /см. далее/.

Ниже приводится таблица где указано, какие операции могут быть выполнены по команде ЧтПА и каким образом можно записать коды этих операций в ячейки памяти.

\* Как правило, в других ЭЦВМ нет специальной памяти для хранения команд - они хранятся в тех же ячейках, что и числа, т.е. в любой ячейке памяти хранится либо число, либо команда.

Таблица записи кодов операций в числовых ячейках.

№ пп	Символ операции	Код операции в двоичной системе	Порядок соответствующий данному коду	Состояние правой кнопки "Стирание памяти"
1	2	3	4	5
1	Сл	00001	+1	
2	Выч1	00010	+2	нажать кнопку
3	Выч2	00011	+2	
4	Умн	00100	+3	нажать кнопку
5	Дел	00101	+3	
6	Чт	00110	+4	нажать кнопку
7	Зп	00111	+4	
8	ЧтП	01011	+7	
9	ЗпП	01100	+8	нажать кнопку
10	Слф	01110	+9	нажать кнопку
11	Вычф	01111	+9	
12	СчП	10011	-2	

Примечание. При записи кодов команд в случаях 2, 4, 6, 9, 10 кнопки "Запись в ячейку" и "Стирание памяти" /правая/ должны быть нажаты одновременно, причем раньше следует нажать кнопку "Стирание памяти". Эта процедура обеспечивает гашение младшего разряда регистра порядка.

Пример. Чтобы записать в какую-либо ячейку команду Умн35 следует набрать на клавиатуре нужный адрес, установить режим "Ввод", набрать на клавиатуре "Порядок" и "Мантисса" число +3+035.У.У и при нажатой правой кнопке "Стирание памяти" нажать кнопку "Запись в ячейку".

#### Использование подпрограмм, помещенных в числовых ячейках памяти.

Пусть подпрограмма из  $n$  команд размещена в ячейках с адресами  $C, C+1, \dots, C+(n-1)$ . Выберем две ячейки с адресами  $A$  и  $B$ ; в ячейку  $A$  занесем число  $+7-01C$  / код +7 соответствует команде ЧтП, поэтому иногда просто пишут  $A / A = \text{ЧтП} - 0.C$ ;  $B$  - любая свободная ячейка.

Теперь, чтобы обеспечить работу нашей подпрограммы, составляем следующую программу:

```
00 Чт В
01 Чт А
02 Зп В
03 СчП А
04 УпИ 00
05 Ост.
```

Пояснение. Ячейка В нужна для хранения содержимого сумматора при переходе к выполнению очередной команды подпрограммы.

Х. КОДИРОВКА ЧИСЕЛ И КОМАНД В ЭЦВМ "ПРОМИНЬ". Большинство ЭЦВМ оперирует с числами, записанными в виде последовательности нулей и единиц, при этом широко используется двоичная система счисления. Числа в двоичной системе записываются с помощью двух цифр: 0 и 1. Запись чисел производится аналогично записи чисел в десятичной системе счисления, которой мы повседневно пользуемся. Например, число 11011 в двоичной системе равно  $1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 16 + 8 + 0 + 2 + 1 = 27$  в десятичной системе /коротко  $11011_2 = 27_{10}$ /. Число 2 называют основанием двоичной системы счисления. Десятичная система счисления имеет основанием число 10. Например,  $1972 = 1 \cdot 10^3 + 9 \cdot 10^2 + 7 \cdot 10^1 + 2 \cdot 10^0 = 1000 + 900 + 70 + 2 = 1972$ .

В машинах "Проминь" при внутреннем кодировании десятичных цифр в числовых ячейках памяти каждая цифра записывается в виде последовательности из 4-х нулей и единиц с "весами" 5211. Именно, запись 0000 представляет собой сумму  $0 \cdot 5 + 0 \cdot 2 + 0 \cdot 1 + 0 \cdot 1$  вместо знака 0 должны стоять либо 0 либо 1/. Например, цифра 6 записывается в виде 1001 / $1 \cdot 5 + 0 \cdot 2 + 0 \cdot 1 + 1 \cdot 1 = 6$ /, цифра 7 в виде 1011 и т.д. При этом в записи любой цифры /кроме 0 и 5/ в последнем разряде должна стоять 1 /0001/; это делает запись цифр однозначной.

Знаковые разряды кодируются так: знаку "+" соответствует 0, знаку "-" соответствует 1.

При кодировании команд коду /символу/ операции соответствует пять двоичных разрядов; при этом используется двоичная система счисления. Например, код операции деления /05/ записывается в виде 05101 / $05 = 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$ /, код операции V /25/-

записывается в виде  $11001 / 25 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 /$  - см. таблицу на стр. 29-31. 26.-

Далее, адреса разрешенных ячеек кодируются двумя десятичными цифрами с "весами" 5211; адреса "запрещенных" ячеек используют также кодировку с "весами" 8421 / двоичная система/. Например, при записи адреса 8  $12^{\#}$  8 записывается в виде 1101 /кодировка 5211/, а  $12^{\#}$  в виде 1100 /кодировка 8421/. Аналогично кодируются адреса других "запрещенных" ячеек.

Зная кодировку команд, можно легко читать программы, пробитые на перфокартах. Нетрудно также понять, каким образом построена таблица на стр. 24.

#### XI. ОБ ИСПОЛЬЗОВАНИИ "ЗАПРЕЩЕННЫХ" ЯЧЕЕК. В ЭЦМ "Проминь-М"

имеется 60 "запрещенных" ячеек с номерами

02 <sup>#</sup>	12 <sup>#</sup> ...	42 <sup>#</sup>	52 <sup>#</sup> ...	92 <sup>#</sup>
04 <sup>#</sup>	14 <sup>#</sup> ...	44 <sup>#</sup>	54 <sup>#</sup> ...	94 <sup>#</sup>
06 <sup>#</sup>	16 <sup>#</sup> ...	46 <sup>#</sup>	56 <sup>#</sup> ...	96 <sup>#</sup>
010 <sup>#</sup>	110 <sup>#</sup> ...	410 <sup>#</sup>	510 <sup>#</sup> ...	910 <sup>#</sup>
012 <sup>#</sup>	112 <sup>#</sup> ...	412 <sup>#</sup>	512 <sup>#</sup> ...	912 <sup>#</sup>
014 <sup>#</sup>	114 <sup>#</sup> ...	414 <sup>#</sup>	514 <sup>#</sup> ...	914 <sup>#</sup>

Из них 12 ячеек с номерами 82<sup>#</sup>+914<sup>#</sup> отведены для констант, 28 ячеек, с номерами 02<sup>#</sup>+410<sup>#</sup> используются для вычисления элементарных функций. Остальными ячейками, кроме 52<sup>#</sup> можно свободно пользоваться. Занесение чисел с пульта в "запрещенные" ячейки производится одновременным нажатием левой кнопки "Стирание памяти" и "Запись в ячейку" в режиме "Ввод1".

Далее, если тумблер на субблоке А5 / второй сверху блок за правой дверцей/ поставить в положение 16, то, во-первых, при одновременном нажатии обеих кнопок "Стирание памяти" нули запишутся как в "разрешенные", так и в "запрещенные" свободные ячейки и, во-вторых, в режиме "Ввод-2" числа будут записываться подряд и в "разрешенные" и в "запрещенные" ячейки: 01, 02<sup>#</sup>, 02, 04<sup>#</sup>, 03, 06<sup>#</sup>, 04, 05, 06, 010<sup>#</sup>... Если же указанный тумблер находится в положении "10", то при нажатии кнопок "Стирание памяти" нули запишутся лишь в "разрешенные" ячейки и в режиме "Ввод2" числа будут заноситься лишь в "разре-

шенные" ячейки.

Проверка содержимого "запрещенных" ячеек, как обычно, осуществляется аналогично записи с помощью кнопки "Вызов на СМ". Как уже было сказано ранее, 0-й блок памяти у "Проминь-2" устроен в точности так же, как и у "Проминь-М". В 1-м блоке памяти машины "Проминь-2" также имеется 60 "запрещенных" ячеек; из них 12 ячеек с номерами 182<sup>ж</sup>+1914<sup>ж</sup> заняты константами, остальными ячейками с номерами 102<sup>ж</sup>+1714<sup>ж</sup> можно свободно пользоваться с учетом сделанных выше замечаний.

Отметим еще, что числа 2<sup>ж</sup>, 4<sup>ж</sup>, 6<sup>ж</sup>, 10<sup>ж</sup>, 12<sup>ж</sup>, 14<sup>ж</sup> изображаются на индикаторном табло путем наложения двух цифр согласно следующей таблице

Десятичные числа	Двоичные коды	Индикация
2 <sup>ж</sup>	0010	0,2
4 <sup>ж</sup>	0100	0,3
6 <sup>ж</sup>	0110	0,4
10 <sup>ж</sup>	1010	5,7
12 <sup>ж</sup>	1100	5,8
14 <sup>ж</sup>	1110	5,9

Свободные "запрещенные" ячейки обычно используются для хранения промежуточных результатов вычислений, т.е. являются рабочими ячейками.

ХП. ДОПОЛНИТЕЛЬНЫЕ СВЕДЕНИЯ О КОМАНДАХ ЭЦМ "ПРОМИНЬ"  
/команды ЗпПА, БпПА, СлФА, ВычФА и микропрограммные операции/.  
Команда ЗпПА /запись по адресу второго ранга/.

По команде ЗпПА число из сумматора записывается в ячейку, адрес которой находится в 3-м и 4-м разрядах ячейки А.

Команда БпПА /безусловных переход по адресу второго ранга/.

По команде БпПА управление передается команде, номер которой записан в 3-м и 4-м разрядах ячейки А. Другими словами, после команды БпПА выполняется команда, номер которой помещен в 3-м и 4-м разрядах ячейки А.

Некоторые особенности команд Сдфа и Вычфа.

Во-первых, при выполнении этих команд порядок результата получается логического сложения порядков чисел, участвующих в выполнении данной операции.

При выполнении логического сложения следует знать, что в качестве слагаемых могут выступать лишь 0 и 1, причем  $0+0=0, 0+1=1, 1+0=1, 1+1=1$ .

Пример. Порядок +6/01001/ сложим логически с порядком -2/10011/ получим 11011, т.е. -7.

Далее, при действиях над числами с равными мантиссами следует учитывать следующее.

При сложении с фиксированной запятой равных мантисс противоположных знаков получается мантисса равная 0 со знаком 1-го слагаемого / т.е. /S/ /.

При Вычф равных положительных /отрицательных/ мантисс получается +0/-0/. Эти сведения могут оказаться полезными, если результат используется для разветвления в программе.

Некоторые особенности микропрограммных операций.

При вычислении  $\ln x, \sqrt{x}$ , если  $x < 0$  / а для  $\ln x$  и при  $x 0$  / проходит аварийный останов /Авост/. При этом продолжать вычисления можно лишь нажав кнопку "Нач. сброс". Сигнал Авост вырабатывается также при вычислении  $e^x$  и  $\ln x, \ln x$ , если  $x$  не принадлежит интервалу  $-10^6 < x < 20,723/$ .

Далее, при вычислениях тригонометрических функций для аргумента  $|x| < 10$  имеем точность 4 знака. Существенно уменьшается точность при  $|x| > 10$ .

В заключение опишем команду Чт 52\*. По команде Чт 52\* число, набранное на клавиатуре переносится на сумматор. Этот факт удобно использовать, например, при вычислении значений  $f(x)$  для различных значений  $x$ .

Пример. Сооставим программу для вычисления и печати  $f(x) = \sqrt{x^2 + 1}$ , предполагая, что  $x$  набрано на клавиатуре.

00	Чт 52*	/S/ = x	03	Сл 86
01	Зп 01		04	$\sqrt{\quad}$
02	Ост 01	печать x	05	Ост 01 печать $f(x)$
03	Умно 01		06	Ост 00

Очевидно, как только программа начнет работать, на клавиатуре можно ~~вы~~бирать следующее значение аргумента.

## СИСТЕМА КОМАНД ЭЦВМ "ПРОМИЙ-М" и "ПРОМИЙ-Э"

Код операции в десятичной и двоичной системах	Наименование операции	Условное обозначение	Содержание
01/0001/	Сложение	СлА	$ S  =  A  +  A $
02/0010/	Вычитание 1	Вч1А	$ S  =  A  -  A $
03/0011/	Вычитание 2	Вч2А	$ S  =  A  -  S $
04/00100/	Умножение	УмнА	$ S  =  S  \cdot  A $
05/00101/	Деление	ДелА	$ S  =  S  /  A $
06/00110/	Посылка числа в сумматор	ЧтА	$ S  =  A $
07/00111/	Запись числа из сумматора в ячейку	ЗпА	$ A  =  S $
08/01000/	Безусловная передача управления 1	Бп 1	После команды Бп 1 выполняется команда с номером 1
09/01001/	Условная передача управления 1	Уп1 1	Если число в сумматоре отрицательное, то после Уп1 1 выполняется команда с номером 1, если же знак числа в сумматоре +, выполняется команда, очередная по номеру за Уп1 1.
10/01010/	Условная передача управления 2	Уп2 1	Если $ S  = 0$ , то выполняется команда 1; если же $ S  \neq 0$ , выполняется команда, очередная по номеру за Уп2 1.
11/01011/	Чтение второго ранга	ЧтПА	Выполняется команда, записанная в ячейке А, В частности, если $ A  = -1 \pm 0$ <u>Нис.</u> , то команда ЧтПА эквивалентна команде Чт N / см. стр. 18, 23/.
12/01100/	Запись по адресу второго ранга	ЗпПА	Число из сумматора записывается в ячейку, адрес которой находится в 3-м и 4-м разрядах ячейки А.
13/01101/	Безусловная передача по адресу второго ранга	БпПА	После БпПА выполняется команда, адрес которой находится в 3-м и 4-м разрядах ячейки А.

А	Г	С	И	Э	И
14/0110/	Сложение с фиксированной запятой	СлФА			$/S/ = /S/ + /A/$ Происходит алгебраическое сложение
15/0111/	Вычитание с фиксированной запятой	ВычФА			$/S/ = /S/ - /A/$ вычитание/ мантисс числа в сумматоре и числа в ячейке А /стр 19 /.
16/10000/	Синус	111			$/S/ = \sin/S/$ При вычислении
17/10001/	Косинус	111			$/S/ = \cos/S/$ значений триго-
18/10010/	Тангенс	111			$/S/ = \tan/S/$ нометрических функций аргумент считается выраженным в радианах.
19/10011/	Синус гиперболический				$/S/ = \sinh/S/$
19/10011/	Счет циклов и СчПА переадресация				Увеличиваются на единицу два числа, одно из которых находится в 1-м и 2-м, другое в 3-м и 4-м разрядах ячейки А.
<p>Нумерация разрядов мантиссом:</p> <p><math>\pm \square \pm \square \square \square \square</math>  <math>5^4 4^3 3^2 2^1 1^0</math></p>					
21/10101/	Тангенс гиперболический	th			$/S/ = \tanh/S/$
22/10110/	Арсинус	arsin			$/S/ = \arcsin/S/$ Вычисленные
23/10111/	Аркосинус	arcsin			$/S/ = \arccos/S/$ значения об-
24/11000/	Арктангенс	arctg			$/S/ = \arctan/S/$ обратных тригонометрических функций выражены в радианах.
25/11001/	Корень квадратный	$\sqrt{\quad}$			$/S/ = \sqrt{S/}$
26/11010/	Экспоненциал	exp			$/S/ = e^{S/}$
27/11011/	Логарифм натуральный	ln			$/S/ = \ln/S/$
28/11100/	Умножение векторов	$\vec{a} \cdot \vec{b}$			Вычисляется скалярное произведение двух n-мерных векторов /стр. 10/
29/11101/	Формирование знака	ФрА			Знак числа в сумматоре становится равным знаку числа в ячейке А. В частности, по команде Фр00 /или просто Фр/ образуется модуль числа в сумматоре.

1	2	3	4
30/IIIЮ/	Решение системы	РС	Решается система линейных алгебраических уравнений до 7-го порядка включительно /стр.Ю/
31/IIII/	Остановка	Ост	Машина прекращает вычисления
31/IIII/	Печать	Ост01	Печатается содержимое сумматора.

**ПОЯСНЕНИЕ** Обозначения типа  $/S/ = |S| - /A/$ ,  $/S/ = \sin S/$  следует читать так: содержимое сумматора становится равным прежнему содержимому сумматора минус содержимое ячейки А; содержимое сумматора становится равным синусу прежнего содержимого сумматора.

В заключение укажем примерное время выполнения некоторых операций в ЭЦМ "Проминь".

В течение 1 сек. машина может выполнять /в среднем/:

1. Операций сложения /вычитания/ 1000;
2. Операций умножения 100;
3. Операций деления и вычисления значений элементарных функций 1;
4. Операций передачи управления 7000.

Сделаем еще замечание общего характера.

Данное пособие посвящено описанию работы конкретной ЭЦМ. Однако основные принципы работы: программное управление, оперирование с содержимым ячеек, где могут храниться произвольные числа /или команды/ - являются общими для всех ЭЦМ. Далее, блок-схемы и основные приемы составления программ, приведенные в настоящем пособии, могут быть успешно использованы при решении задач на любой ЭЦМ.

**БРЕСТСКИЙ ИНЖЕНЕРНО-СТРОИТЕЛЬНЫЙ ИНСТИТУТ**

Отвественный за выпуск ~~инж. Корольков~~ В.И.

АБ 25189, подписано к печати 17/УП-1972 года, формат 64x84 1/16 п.л., объем 2 п.л., заказ № 361, тираж 600 экз.

Отпечатано на ротаприте МСС обьстатуправления г.Брест

Бесплатно.