ОБЪЕДИНЕННЫЙ
ИНСТИТУТ
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

Дубна

10-3183-1

В.Н. Аносов, Ю.Н. Денисов

ПРОГРАММА ЦИФРОВОГО АНАЛИЗАТОРА ГАРМОНИК МАГНИТНОГО ПОЛЯ

ЛАБОРАТОРИЯ ЯДЕРНЫХ ПРОБЛЕМ

1967 г.

10-3183-1

В.Н. Аносов, Ю.Н. Денисов

ПРОГРАММА ЦИФРОВОГО АНАЛИЗАТОРА ГАРМОНИК МАГНИТНОГО ПОЛЯ

Рассматриваемая специализированная вычислительная машина [1] предназначена для гармонического анализа распределения магнитного поля циклических ускорителей (релятивистских циклотронов [2], фазотронов с жесткой фокусировкой [3] и т.д.), заданного в виде 2ν частных значений поля H_λ, равномерно распределенных по всему периметру окружности с радиусом R_1. Коэффициенты Фурье A_k и B_k вычисляются по формулам Бесселя [4]

$$H_0 = \frac{1}{2\nu} \cdot \sum_{\lambda=1}^{2\nu} H_\lambda \tag{1}$$

$$A_k = \frac{1}{\nu} \cdot \sum_{\lambda=1}^{2\nu} H_\lambda \cdot \cos k \cdot \frac{\pi}{\nu} \cdot \lambda$$

$$B_k = \frac{1}{\nu} \cdot \sum_{\lambda=1}^{2\nu} H_\lambda \cdot \sin k \cdot \frac{\pi}{\nu} \cdot \lambda ,$$

где k – порядок определяемой гармоники.

Амплитуды и начальные фазы гармоник находятся по формулам

$$H_{mk} = \sqrt{A_k^2 + B_k^2} \tag{2}$$

$$\phi_{ok} = \operatorname{arc\,tg} \frac{A_k}{B_k} .$$

При проектировании специализированных вычислительных машин, предназначенных для решения узкого круга задач или, как в нашем случае, одной определенной задачи, стремятся удовлетворить поставленным требованиям по быстродействию и точности результатов с минимальным объемом используемого оборудования. Естественно, что это соображение учитывалось и при разработке анализатора гармоник.

На основании чернового варианта программы вычислений по формулам Бесселя, а также известного требуемого времени для решения задачи и точности, с которой необходимо получить результаты, был составлен перечень основных используемых в машине операций и выбраны оптимальные для наших условий разработки варианты построения основных узлов анализатора.

Арифметическое устройство выбрано параллельного типа, так как по числу элементов оно отличается от последовательного АУ только на m -цепочек сквозного переноса в сумматоре (где m - разрядность сумматора), тогда как для обеспечения равного быстродействия в последовательном АУ пришлось бы существенно увеличить тактовую частоту работы элементов, что сделало бы его по стоимости сравнимым с параллельным АУ, но менее надежным в работе.

В машине принята система представления двоичных чисел с фиксированной запятой. Это можно сделать благодаря тому, что диапазон чисел, с которым оперирует анализатор, невелик и заранее известен, программа стандартна и поэтому нет трудностей с подбором масштабных коэффициентов. Недостатки системы представления чисел с фиксированной запятой в анализаторе гармоник практически не проявляются, тогда как преимущества этой системы по сравнению с представлением чисел с плавающей запятой довольно убедительны. В машине, оперирующей с числами с фиксированной запятой, значительно упрощаются АУ и устройство управления и возрастает скорость счета. В анализаторе запятая фиксирована непосредственно после знакового разряда.

В запоминающем устройстве анализатора числа хранятся в прямом коде со знаком. Этот способ по скорости выполнения таких операций как сложение и вычитание несколько уступает способу хранения чисел в ЗУ в дополнительном коде, но оказывается более удобным при выполнении ряда других операций: присвоение знака числу, сравнение по модулю, извлечение части разрядов числа и при некоторых, описанных ниже, нестандартных операциях, выполняемых в анализаторе.

Принятый способ хранения чисел в ЗУ машины предопределил и выбор метода умножения. Умножение чисел должно вестись в прямых кодах, а знак произведения образовываться отдельно. В результате анализа ряда методов умножения [5] с целью выяснения наиболее пригодного из них для применения в анализаторе гармоник был выбран способ, отличающийся следующими особенностями. Умножение двоичных кодов сомножителей начинается со старших разрядов. Сдвиги в регистрах множителя и сумматора идут в сторону старших разрядов, а в

регистре множимого сдвиг отсутствует. При перемножении m -разрядных сомножителей 2m -разрядное произведение получается за счет того, что частичные произведения по мере их получения сдвигаются в освобождающиеся младшие разряды множителя. Операции сдвига частичного произведения в сумматоре и прибавление множимого к содержимому сумматора разнесены во времени. Блок-схема АУ, необходимого для реализации выбранного метода умножения, приведена на рис. 1.

Деление является относительно часто встречающейся операцией в программе анализатора гармоник, поэтому вопрос о том, вводить ли деление как отдельную операцию или же заменить его умножением на обратную величину, получаемую программным методом, был решен в пользу осуществления отдельной операции деления. Это потребовало минимальных добавочных затрат оборудования, поскольку одним из критериев пригодности выбранного метода умножения была возможность осуществления на том же АУ и операции деления.

Операция деления выполняется в анализаторе методом, не требующим восстановления остатка [6]. Из делимого вычитается делитель, и в случае получения отрицательного остатка прежнее значение остатка не восстанавливается. Полученная разность сдвигается влево на один разряд, и в следующем цикле деления делитель уже не вычитается, а прибавляется к остатку. Положительный знак остатка означает, что очередная цифра частного "1", отрицательный - "0". Ввиду того, что вычитание делителя из остатка производится в обратном коде, при делении нацело в остатке получается обратный код нуля "1, 111, ... , 1". Это приведет к ошибочному определению очередной цифры частного ("0" вместо "1"), и в дальнейших шагах деления в частном будут все "1". Таким образом, отбрасывая в частном младшие разряды дальше m -ого, можно получить результат деления с ошибкой в единицу m -го разряда. Чтобы избежать этого, в машине в операции деления введено округление, заключающееся в прибавлении "1" в младший m -ый разряд частного.

Приведенный набор операций в процессе коррекции чернового варианта программы был дополнен новыми, нестандартными операциями. Описание этих операций приводится ниже вместе с описанием окончательного варианта программы.

Программа цифрового анализатора гармоник включает в себя следующие основные части:

1. Программа подготовки машины к работе.

2. Программа перевода исходных данных H_λ из десятичной системы счисления в двоичную и засылки их в МОЗУ.

3. Программа счета H_0, заканчивающаяся переводом двоичного кода результата в десятичный и печатью десятичного кода H_0.

4. Программа счета коэффициентов Фурье A_k и B_k.

5. Программа определения модуля k-ой гармоники H_{mk}.

6. Программа вычисления начальной фазы ϕ_{ok}, заканчивающаяся переводом двоичного кода H_{mk} и ϕ_{ok} в десятичный и печатью результата в десятичном коде.

Логическое сочленение перечисленных выше кусков программы поясняет блок-схема, показанная на рис. 2.

Рассмотрим каждую часть программы подробнее.

Программа подготовки машины к работе состоит из команд очистки рабочих ячеек МОЗУ и команд восстановления в определенных ячейках МОЗУ исходного кода формируемых команд.

Для программного перевода исходного массива 2ν чисел H_λ из шестиразрядного десятичного кода в двадцатиразрядный двоичный код использован метод, описанный в работе [7]. Поясним суть этого метода на примере. Предположим, что необходимо перевести в двоичную систему счисления число 0,512. Это число можно записать в виде $10^{-3}\cdot[(5\cdot10+1)\cdot10+2]$. Из такой формы записи становится понятным алгоритм перевода. Двоичный эквивалент цифры старшего разряда десятичного числа умножается на 10 (в двоичной системе счисления умножение идет на код "1010", который хранится в младших разрядах 20-разрядной сетки машины) и прибавляется к двоичному эквиваленту цифры второго по старшинству разряда. Сумма этих чисел вновь умножается на 10, складывается с двоичным эквивалентом цифры следующего разряда и т.д. до тех пор, пока не будут использованы все цифры переводимого десятичного числа. Двоичные эквиваленты цифр, из которых состоит переводимое десятичное число, получаются на дешифраторе, преобразующем десятичный код входных чисел H_λ в двоично-десятичный. В заключение полученный в результате вышеизложенных операций двоичный код умножается на 10^{-n}, где n -число разрядов в переводимом десятичном числе (в рассматриваемом примере $n = 3$).

Для того чтобы избежать переполнения разрядной сетки машины при использовании описанного алгоритма перевода на машине с фиксированной запятой, необходимо четырехразрядные двоичные эквиваленты десятичных цифр при операциях в арифметическом устройстве машины записывать в самые младшие разряды соответствующих регистров. Это привело к появлению нестандартной операции умножения, обозначаемой в дальнейшем x''. Дело в том, что при умножении кода "1010", находящегося в младших разрядах регистра множимого, на 4-разрядный двоичный эквивалент очередной переводимой десятичной цифры, который также находится в младших разрядах регистра множителя, получится максимально (восьмиразрядное) произведение в самых младших разрядах сумматора. При обычном же умножении двух m-разрядных кодов m старших разрядов произведения образуются в регистре множителя и оттуда считываются в память машины. Поэтому для сохранения идентичности процесса считывания результата умножения в МОЗУ машины полученный в сумматоре АУ код произведения коэффициента "1010" на двоичный эквивалент очередной переводимой десятичной цифры хранится там до следующей команды, которая представляет собой обычное умножение, но по адресам множимого и множителя в памяти записаны нули. После выполнения такой команды произойдет только сдвиг оставшегося от предыдущей операции содержимого сумматора на "m" разрядов и засылка его в регистр множителя, откуда обычным способом произведение считывается в МОЗУ.

Заключительным этапом программы перевода десятичного кода числа в двоичный является умножение полученного двоичного кода на поправочный коэффициент, учитывающий множитель 10^{-n} и масштабирующий коэффициент 2^m, где m -число разрядов в регистрах АУ машины, возникающий как следствие засылки 4-разрядных двоичных кодов переводимых десятичных цифр в младшие разряды регистров АУ. Следовательно, поправочный множитель равен

$$\theta = 10^{-n}\cdot 2^m. \tag{3}$$

Легко показать на примере, что коэффициент $\theta > 1$. Действительно, пусть $n = 3$, тогда, так как $m \approx 3{,}3 \cdot n \approx 10$, то $\theta = 10^{-3}\cdot 2^{10} = 1{,}024 > 1$. Учитывая, что в рассматриваемой машине запятая фиксирована перед старшим разрядом двоичного кода, то есть числа, с которыми оперирует машина должны быть по модулю меньше единицы, множить результат перевода десятичного кода числа в двоичный не коэффициент θ нельзя. Операция умножения заменяется в этом случае делением на коэффициент.

$$\eta = \frac{1}{\theta} = 10^{n} \cdot 2^{-m} < 1 . \qquad (4)$$

При составлении программы вычисления H_0 можно было идти двумя путями, соответствующими двум видам развернутой записи формулы (1):

$$а) \quad H_0 = \frac{1}{2\nu} \cdot (H_1 + H_2 + \ldots + H_\lambda) \qquad (5)$$

$$б) \quad H_0 = \frac{1}{2\nu} \cdot H_1 + \frac{1}{2\nu} \cdot H_2 + \ldots + \frac{1}{2\nu} \cdot H_\lambda . \qquad (6)$$

Для рассматриваемой машины [1] $H_\lambda = 10000{,}0 - 20000{,}0$ э и поэтому число разрядов регистров АУ и ЗУ равно 20. Максимальное десятичное число, которое можно записать в такой разрядной сетке равно 99999,9 э. Следовательно, воспользовавшись вариантом а), получим переполнение регистров АУ при суммировании H_λ уже на 5–10 шаге, а так как число членов в этой сумме равно $2\nu = 24, 36, 48, 72$ или 144 [1], то для ликвидации переполнения в этом варианте необходимо ввести n добавочных старших разрядов, где "n" определяется максимально возможной $\sum_{\lambda=1}^{2\nu} H_\lambda$. Поскольку $H_0 \leq 20000{,}0$ э, а $2\nu \leq 144$, то:

$$(H_1 + H_2 + \ldots + H_\lambda)_{max} = 2\nu_{max} \cdot H_{\lambda max} = 2880000{,}0 ,$$

что составляет 8 десятичных или 27 двоичных разрядов. В анализаторе $m = 20$, следовательно, $n = 7$ двоичным разрядам.

Если при вычислении H_0 реализовать вариант б) и оставлять только m-разрядные произведения $\frac{1}{2\nu} \cdot H_\lambda$, то максимальная ошибка в каждом таком произведении составит 2^{-m} и, следовательно, максимальная погрешность суммы $\sum_{\lambda=1}^{2\nu} \frac{1}{2\nu} \cdot H_\lambda$ может оказаться в ℓ-ом разряде, поскольку она равна:

$$2\nu \cdot 2^{-m} = 2^{\ell} \cdot 2^{-m} = 2^{-m+\ell} .$$

Таким образом, для того чтобы получить в этом варианте программы H_0 с точностью 2^{-m}, нужно ввести со стороны младших разрядов произведений $\frac{1}{2\nu} \cdot H_\lambda$ "ℓ" добавочных разрядов. Тогда результаты умножения $\frac{1}{2\nu}$ на H_λ располагаются в $(m+\ell)$ -разрядной сетке и свободные старшие разряды гарантируют от переполнения регистров АУ при последующем сложении этих произведений, а ℓ добавочных младших разрядов не дадут накопиться ошибке в сумме $\sum_{\lambda=1}^{2\nu} \frac{1}{2\nu} \cdot H_\lambda$ больше, чем 2^{-m}.

Сравнение вариантов а) и б) показывает, что их реализация требует одинакового добавочного оборудования, но так как вариант а) имеет значительно меньшее время счета из-за наличия только одного умножения вместо 2ν умножений, необходимых в варианте б), то при составлении программы машины был принят этот вариант с некоторыми модификациями, рассмотренными ниже.

Найденное для варианта а) число дополнительных разрядов $n = 7$ должно быть скорректировано, поскольку из [1] известно, что амплитуда некоторых гармоник H_{mk} достигает 6000,0 э и, следовательно, $H_{mk}^2 = 36000000{,}00$, то есть образующееся при вычислении H_{mk} подкоренное выражение $(A_k^2 + B_k^2)$ может иметь 33 двоичных разряда и число дополнительных разрядов увеличивается до 13. Было признано нецелесообразным устранять переполнения регистров машины путем увеличения разрядной сетки. Это затруднение преодолено введением специальных операций $+'$ и $+''$, позволяющих получать $2m$ -разрядные суммы: $\sum_{\lambda=1}^{2\nu} H_\lambda$;

$$\sum_{\lambda=1}^{2\nu} H_\lambda \cdot \cos k \cdot \frac{\pi}{\nu} \cdot \lambda \quad ; \quad \sum_{\lambda=1}^{2\nu} H_\lambda \cdot \sin k \cdot \frac{\pi}{\nu} \cdot \lambda$$

и подкоренное выражение $(A_k^2 + B_k^2)$ на m -разрядных АУ и ЗУ.

Поясним на примере принцип сложения двух $2m$ -разрядных чисел на m-разрядном АУ. Числа могут быть как положительными, так и отрицательными. Например:

1,10111001 – обратный код I-го слагаемого,

1,11001001 – обратный код II-го слагаемого.

Делим код обоих чисел на две части по m разрядов в каждой.

1001– m младших разрядов I-го слагаемого,

1001 – –"– II -го слагаемого,

1011 – m старших разрядов I -го слагаемого,

1100 – –"– II -го слагаемого.

Процесс суммирования разбивается на следующие этапы:

1. Суммирование младших разрядов:

$$\begin{array}{r} + \begin{array}{r} 1001 \\ \underline{1001} \end{array} \\ 1 \text{ перенос} \longrightarrow \quad 0010 \end{array}$$

Единица переноса, возникшая при сложении младших разрядов, запоминается и после сложения старших разрядов добавляется к их сумме.

2. Сложение старших разрядов

$$\begin{array}{r} + \begin{array}{r} 1.1011 \\ 1.1100 \end{array} \\ \hline 1 \ \text{перенос} \quad 1.0111 \end{array}$$

Учтем единицу переноса из младших разрядов

$$\begin{array}{r} + \begin{array}{r} 1.0111 \\ 1 \end{array} \\ \hline 1.1000 \end{array}$$

3. Сложение единицы переноса из старших разрядов с суммой младших разрядов (циклический перенос)

$$\begin{array}{r} + \begin{array}{r} 0010 \\ 1 \end{array} \\ \hline 0011 \end{array},$$

при этом возможен случай, когда при таком сложении возникнет еще одна единица переноса в старшие разряды. Следовательно, необходим еще один этап.

4. Сложение суммы старших разрядов с единицей вторичного переноса из младших разрядов.

Таким образом, полный цикл сложения двух $2m$ разрядных чисел, одно из которых или оба отрицательны, выполняется в 4 команды:

1. Сложение младших разрядов с запоминанием единицы переноса (операция +').

2. Сложение старших разрядов с учетом единицы переноса из младших разрядов и запоминание переноса, образующегося при сложении старших разрядов (операция +").

3. Сложение младших разрядов с единицей переноса из старших разрядов и запоминание нового переноса в старшие разряды (операция +').

4. Сложение старших разрядов со вторичной единицей переноса из младших разрядов (операция +").

Для сложения двух положительных $2m$ -разрядных чисел достаточно всего двух команд – операция +' и операция +", поскольку при суммировании старших разрядов перенос в этом случае возникнуть не может.

При получении $2m$ -разрядных сумм $\sum\limits_{\lambda=1}^{2\nu} H_\lambda \cos\Delta\alpha\cdot\lambda$ и $\sum\limits_{\lambda=1}^{2\nu} H_\lambda \sin\Delta\alpha\cdot\lambda$ каждое очередное слагаемое, перед прибавлением к уже имеющейся частичной

сумме, представляется в виде $2m$ -разрядного двоичного кода, в котором m младших разрядов представляют собственно величину $H_\lambda\cdot\cos\Delta\alpha\cdot\lambda$, либо $H_\lambda\cdot\sin\Delta\alpha\cdot\lambda$, а m старших разрядов всегда нули, но в знаковом разряде – знак очередного произведения, присвоенный операцией +'''. Следует отметить, что при сложении младших разрядов знак их промежуточной суммы всегда устанавливается положительным, так как при последующем шаге суммирования код суммы младших разрядов не должен меняться при передачах из АУ в ЗУ и обратно.

Сумма 2ν слагаемых также хранится в двух ячейках МОЗУ: в одной – старшие разряды со знаком, в другой – младшие всегда с положительным знаком (в знаковом разряде записывается "0"). Для того чтобы такая составная $2m$ – разрядная сумма могла участвовать в дальнейших вычислениях по программе, ее младшим разрядам необходимо присвоить знак всей суммы и трансформировать их код в соответствии с этим знаком. Это производится операцией +''', которая является также нестандартной и нуждается в более подробном пояснении.

Операция +''', помимо указанных выше двух функций – присвоение знака числа нулевой ячейке и присвоение знака старших разрядов младшим разрядам, позволяет получать модуль числа. Во всех трех случаях операция выполняется стереотипно. При образовании модуля числа, число, модуль которого определяется, записывается по 2-му адресу, а по 1-му записан "0". Для присвоения знака ячейке МОЗУ, в которой записан "0", число, знак которого присваивается, записывается по первому адресу, ячейка с содержанием "0" – по второму. Наконец, при присвоении знака старших разрядов младшим, старшие разряды записываются в ячейку МОЗУ по первому адресу, а младшие – по второму.

Программа счета H_0 заканчивается переводом результата вычислений из двоичного в десятичный код и последовательным выводом десятичного кода на печать. Метод перевода /8/ сводится к следующему. Двоичный код числа умножается на 0,625 (двоичный код числа 0,625 – 0,1010) и после каждого такого умножения (нестандартная операция – x) четыре старших разряда образующегося ($m+4$) – разрядного произведения выводятся на специальный регистр блока печати и дешифрируются в соответствующую десятичную цифру. В следующем цикле умножения на 0,1010 множится только m -разрядный остаток.

Эта операция x' проделывается столько раз, сколько десятичных цифр числа H_0 отыскивается. В анализаторе гармоник H_0 получается в виде 6-разрядного десятичного числа.

Выбор варианта а) при составлении программы счета H_0 однозначно определил и построение программы вычисления коэффициентов Фурье A_k и B_k. Сначала получаются значения сумм:

$$H_1 \cdot \cos k \cdot \frac{\pi}{\nu} \cdot 1 + H_2 \cdot \cos k \cdot \frac{\pi}{\nu} \cdot 2 + \dots H_\lambda \cdot \cos k \cdot \frac{\pi}{\nu} \cdot \lambda \qquad (7)$$

$$H_1 \cdot \sin k \cdot \frac{\pi}{\nu} \cdot 1 + H_2 \cdot \sin k \cdot \frac{\pi}{\nu} \cdot 2 + \dots H_\lambda \cdot \sin k \cdot \frac{\pi}{\nu} \cdot \lambda\ , \qquad (8)$$

а затем эти суммы умножаются на $1/\nu$. Новых факторов, могущих вызвать переполнение разрядной сетки машины при получении сумм (7) и (8) по сравнению с $\sum\limits_{\lambda=1}^{2\nu} H_\lambda$, нет. Напротив, умножение каждого значения H_λ на величину, меньшую единицы, а также знакопеременное суммирование могут только уменьшить величины сумм (7) и (8) по сравнению с $\sum\limits_{\lambda=1}^{2\nu} H_\lambda$.

Вычисление как H_0, так и A_k и B_k идет в машине по накопительному алгоритму. В этом случае эффективное сокращение длины программы достигается использованием метода циклического формирования команд в процессе перехода от цикла к циклу счета. Такой метод формирования команд широко использован при построении программы анализатора гармоник. Формирование новой команды из предыдущей производится путем сложения кода предшествующей команды с вспомогательным кодом, в результате чего могут измениться при необходимости либо код операции, либо первый или второй адрес, либо все три параметра команды.

Амплитуда гармоники после отыскания коэффициентов Фурье A_k и B_k находится по формуле (2). Для того чтобы получить заданную точность вычислений, возведение в квадрат m - разрядных кодов A_k и B_k производится с сохранением $2m$ разрядов в произведении при помощи описанной выше операции x''. Сумма $(A_k^2 + B_k^2)$ также получается $2m$ -разрядной (с использованием операций +' и +″).

Извлечение квадратного корня из $2m$ -разрядной суммы $(A_k^2 + B_k^2)$ реализуется в машине "школьным" методом /9/, при котором вычисление корня

рассматривается как одна операция. Это позволяет сократить объем ПЗУ на величину программы, потребовавшейся бы при извлечении корня программным способом. Кроме того при использовании этого метода нет необходимости в процессе извлечения корня оперировать целиком с $2m$ -разрядным подкоренным выражением.

Суть метода полностью совпадает со "школьным" способом извлечения квадратного корня в десятичной системе счисления. Двоичный код числа, из которого извлекается корень, разбивается на пары разрядов, начиная со старшего. Определяется первая цифра корня как наибольший возможный корень из первой пары (старших) разрядов. В двоичной системе счисления пробные цифры корня могут быть только "0" или "1". Эта цифра затем возводится в квадрат и вычитается из пары старших разрядов. К получившемуся остатку приписывается следующая пара цифр подкоренного выражения и определяется очередная цифра корня. Для этого найденная ранее часть корня удваивается и к ней приписывается следующая пробная цифра x_1, такая, чтобы полученное число, умноженное на x_1, было меньше остатка, но максимально приближалось к его значению. Такая последовательность действий повторяется цикл за циклом до тех пор, пока не будут исчерпаны все пары разрядов подкоренного числа. Для получения результата с точностью до единицы m -го разряда подкоренное выражение должно иметь $2m$ двоичных разрядов. Рассматривемый метод удобен тем, что в процессе счета нет необходимости хранить в АУ все $2m$ разрядов подкоренного выражения. Сначала операция производится только с m старшими разрядами подкоренного выражения, а затем, когда они будут исчерпаны, через $m/2$ тактов извлечения корня, в "освободившийся" регистр АУ считываются m младших разрядов подкоренного выражения и операция продолжается. Таким образом, не увеличивая разрядности АУ и ЗУ, можно извлечь корень из $2m$ -разрядного числа с точностью до единицы младшего m -го разряда. Приспособление АУ, разработанного для выполнения ранее перечисленных операций, и для реализации операции извлечения корня не потребовало существенных добавлений к его схеме.

Начальная фаза гармоник ϕ_{ok} вычисляется в машине по формуле (2). Обозначим $\frac{A_k}{B_k} = x$. Поскольку

$$\phi_{ok} = \operatorname{arc\,tg} x = \operatorname{sign} x \cdot \operatorname{arc\,tg} |x|\ , \qquad (10)$$

то ее значение можно получать в два этапа. Сначала вычислить $\mathrm{arc\,tg}\,|x|$, а квадрант, в котором находится отыскиваемое значение ϕ_{ok}, определять с помощью специальной операции ОК (определение квадранта). При помощи этой операции по сочетанию знаков A_k и B_k определяется, в каком квадранте находится отыскиваемое значение ϕ_{ok} и добавлением к полученному значению $|\phi_{ok}|$ либо 0°, либо $\pm 180^\circ$, либо 360° получается фаза ϕ_{ok} в градусах в диапазоне $0-360^\circ$. Анализ знаков A_k и B_k можно было бы выполнить и программным способом (эта программа заняла бы 13 команд), но оказалось рациональнее ввести специальную операцию – ОК.

Необходимая точность определения начальной фазы гармоники ϕ_{ok} составляет десятые доли градуса /1/, поэтому для аппроксимации формулы $\mathrm{arc\,tg}\,|x|$ выбрана формула /10/

$$\mathrm{arc\,tg}\,x \approx \left(C_0 + \frac{C_1}{d_1 + x^2}\right) x, \tag{11}$$

где : $C_0 \approx 0{,}4444$; $C_1 \approx 0{,}9259$; $d_1 \approx 1{,}6666$.

Поскольку коэффициент $d_1 > 1$, а машина не может оперировать с числом больше единицы, необходимо произвести дополнительное преобразование формулы (11) с заменой d_1 на $d_1' = \frac{d_1}{2} < 1$. Тогда это выражение примет вид:

$$\mathrm{arc\,tg}\,x \approx \left(C_0 + \frac{C_1'}{d_1' + \frac{x^2}{2}}\right) x, \tag{12}$$

где: $C_1' = \frac{C_1}{2}$; $d_1' = \frac{d_1}{2}$.

Кроме того, поскольку по формуле (12) вычисляется значение угла в радианах, а необходимо иметь его в градусах, в это выражение следует ввести коэффициент перевода углов из радиан в градусы $k \approx 0{,}5729577954\ldots$ °/рад. Эта величина коэффициента k определена с учетом принятого в машине масштабирования 360° – 0,360000. Окончательно:

$$\mathrm{arc\,tg}\,x = \left(C_0' + \frac{C_1''}{d_1' + \frac{x^2}{2}}\right) x, \tag{13}$$

где
$$C_0' = k \cdot C_0 = 0{,}025467790,$$
$$C_1'' = k \cdot C_1' = 0{,}026525823,$$
$$d_1' = 0{,}833333333.$$

Формула (13) дает возможность получить 4 верных десятичных значащих цифры в величине $\mathrm{arc\,tg}\,x$ только при условии, что $|x| < \sqrt{2} - 1$. В рассматриваемом случае $0^\circ < \phi_{ok} < 360^\circ$, следовательно, $0 < |x| < \infty$ и для использования этой формулы необходимо преобразовать аргумент $|x|$ к значению, не превышающему 0,414. В качестве первого шага необходимо ввести предварительное сравнение модулей $|A_k|$, $|B_k|$ и деление меньшего из них на большее. Тогда, если $|A_k| < |B_k|$, то $|x| < 1$ и вычисление ведется обычным способом. При $|A_k| > |B_k|$ и соответственно $|x| > 1$ вычисляется $\mathrm{arc\,tg}\,|x_1|$, где $|x_1| = \frac{1}{|x|} < 1$, а полученный результат преобразуется по формуле

$$\mathrm{arc\,tg}\,|x| = \frac{\pi}{2} - \mathrm{arc\,tg}\,|x_1|. \tag{14}$$

Для $|A_k| = |B_k|$, т.е. $|x| = 1$, $|\phi_{ok}| = 45^\circ$. В этом случае счет идет по специальной программе и значение ϕ_{ok} определяется номером квадранта, получаемым из анализа знаков коэффициентов A_k и B_k. Программа счета по формуле (13) обходится и тогда, когда один из коэффициентов A_k или B_k равен нулю. В этом случае ϕ_{ok} равен 0°, 90°, 180° или 270°. Действительное ϕ_{ok} выбирается из этих величин операцией ОК.

В тех случаях, когда $0{,}414 < |x| < 1$, производится следующее преобразование аргумента. Известно, что

$$\mathrm{arc\,tg}\,x - \mathrm{arc\,tg}\,y = \mathrm{arc\,tg}\,\frac{x - y}{1 + xy}.$$

Пусть $y = 1$, тогда

$$\mathrm{arc\,tg}\,|x| = \frac{\pi}{4} + \mathrm{arc\,tg}\,x_2, \tag{15}$$

где $x_2 = \frac{|x| - 1}{|x| + 1}$. Так как в данной машине операции могут производиться только с числами по модулю меньшими единицы, то удобнее записать:

$$x_2 = \frac{\frac{|x|}{2} - \frac{1}{2}}{\frac{|x|}{2} + \frac{1}{2}}. \tag{16}$$

Для выбранного соотношения между $|x|$ и x_2 значение аргумента x_2 всегда меньше 0,414 при $0{,}414 < |x| < 1$. Таким образом, вычислив по формуле (13) $\mathrm{arc\,tg}\,x_2$ и подставив его значение в (15), получим (для случая $|A_k| < |B_k|$) искомый $\mathrm{arc\,tg}\,|x|$. Если же в начале программы вычисления ϕ_{ok} было определено, что $|A_k| > |B_k|$, то полученное значение $\mathrm{arc\,tg}\,|x|$ еще раз корректируется по формуле (14).

Логическая схема программы вычисления начальной фазы гармоник приведена на рис. 3.

После ввода в анализатор гармоник исходной информации в виде 2ν значений магнитного поля H_λ вычисление H_0 и ϕ_{0k}, H_{mk} может производится многократно в любой последовательности.

П Р И Л О Ж Е Н И Е

№№ пп	Условное обозначение операции	Название операции	Характеристика операции
1	2	3	4
1	ПЧ	Передача числа	Число из ячейки "а" передается в ячейку "в"
2.	+	Сложение	Числа из ячеек "а" и "в" алгебраически складываются. Результат направляется в ячейку "в"
3.	+'	Сложение младших разрядов 2m -разрядного числа	Числа из ячеек "а" и "в" алгебраически складываются. Перенос из старшего разряда в знаковый разряд запоминается на T'_3 (запоминающий триггер). Цепь циклического переноса блокируется. Знак результата устанавливается в "0". К результату прибавляется единица, если T^2_3 (запоминающий триггер) находится в единичном состоянии. Результат направляется в ячейку "в"
4.	+"	Сложение старших разрядов 2m -разрядного числа	Числа из ячеек "а" и "в" алгебраически складываются. Циклический перенос из знакового разряда запоминается на T^2_3 (запоминающий триггер). К результату прибавляется "1", если T^1_3 находится в единичном состоянии. Результат направляется в ячейку "в"
		Условный переход	Операция выполняется в две команды
5	1 – 1 ,	Первая команда условного перехода	Числа из ячеек "а" и "в" сравниваются по абсолютной величине. Если число в ячейке "а" больше числа в ячейке "в", то знак разности положительный, и наоборот. Результат сравнения остается в сумматоре
6.	1<1	Вторая команда условного перехода	Если знак разности в первой команде условного перехода был отрицателен, управление передается следующей по номеру команде, если положительный, то команде, номер которой указан во втором адресе
7	1 = 1	Вторая команда условного перехода	Анализируется код, находящийся в сумматоре после первой команды условного перехода. Если это код нуля, то управление передается команде, номер которой указан во втором адресе, если код не равен нулю, то выполняется следующая по номеру команда
8	:	Деление	Число из ячейки "а" делится на число из ячейки "в". Частное округляется. Результат записывается в ячейку "в" (по второму адресу)
9.	x	Умножение	Числа из ячеек "а" и "в" перемножаются. Результат округляется и записывается в ячейку "в"
10.	x"	Умножение чисел с запоминанием остатка в сумматоре	Числа из ячеек "а" и "в" перемножаются. Получается 2m -разрядное произведение. Старшие m -разрядов произведения записываются в ячейку "в". m -разрядный остаток сохраняется в сумматоре и используется при выполнении следующей команды
11.	x'	Умножение чисел с последующим подъемом клавиши печатающей машинки	Числа из ячеек "а" и "в" перемножаются. Четыре старших разряда произведения передаются в устройство вывода, где они дешифрируются в десятичную цифру с подъемом соответствующей клавиши печатающей машинки

1	2	3	4
12	$\sqrt{\ }$	Извлечение корня	Извлекается корень из $2m$-разрядного числа, старшие m-разрядов которого хранятся в ячейке "а", младшие – в ячейке "в". Результат получается m-разрядный и отправляется на хранение в ячейку "в"
13	ОК	Определение квадранта	Анализируются знаки чисел в ячейках "а" и "в". В зависимости от комбинации этих знаков в P_{II} устанавливается двоичный код квадранта в градусах: "180°", "-180°" или "360°", который, складываясь с величиной угла, получаемой по формуле $\phi = \text{arc tg}\,\frac{A_k}{B_k}$, дает результат, записываемый в ячейку "в".
14	+"	Присвоение знака числу	Числу, записанному по второму адресу в ячейке "в", присваивается знак в зависимости от знака чисел в ячейках "а" и "в". В случае присвоения отрицательного знака происходит преобразование кода числа на обратный
15	Печ.	Печать	Происходит срабатывание соленоида "Возврат каретки" в печатающей машинке, которая таким образом подготавливается к приему и печати нового 6-разрядного десятичного числа
16	Ост.	Останов	Прекращается поступление тактовых импульсов в цепи машины

ПРИЛОЖЕНИЕ П

Адрес команды	Код операции	Первый адрес	Второй адрес	Примечание
I	2	3	4	5
		Программа подготовки машины к работе		
К+I	ПЧ	a_I	β_2	Очистка ячейки β_2
+2	ПЧ	a_I	γ	Очистка ячейки γ
+3	ПЧ	H_I	яч. К+2I	Восстановление первоначального содержимого ячеек МОЗУ N К+20 и N К+2I.
+4	ПЧ	H_4	яч. К+20	
+5	Ост.			Останов
		Программа перевода чисел из десятичной системы в двоичную		
+6	ПЧ	H_2	яч. К+I2	Восстановление исходного состояния ячейки МОЗУ N К+I2
+7	ПЧ	в_I	β_1	
+8	X"	a_2	β_1	
+9	X	a_I	β_1	
+I2	+	в_2	β_1	
+I3	+	γ_1	К+I2	Формирование команды N К+I2
+I4	+	a_3	β_2	
+I5	/–/	a_4	β_2	Условный переход
+I6	/</		К+8	
+I7	ПЧ	a_I	β_2	Очистка яч. β_2
+20	ПЧ	a_5	C_I	
+2I	:	β_1	C_I	

I	2	3	4	5
+22	+	γ_2	К+20	Формирование команды N К+20
+23	+	γ_2	К+2I	Формирование команды N К+2I
+24	+	a_3	ξ	
+25	ПЧ	ξ	яч.2ν	
+26	Ост.			Останов

Программа счета среднего поля H_0

m +I	ПЧ	a_I	X	Очистка ячейки X
+2	ПЧ	H_3	яч. m+8	Восстановление исходного состояния ячеек МОЗУ $N m$+8, N К+2I, N К+20.
+3	ПЧ	H_I	яч. К+2I	
+4	ПЧ	H_4	яч. К+20	
+5	ПЧ	a_I	β_1	Очистка ячейки β_1
+8	ПЧ	C_I	δ	
+9	+'	δ	X	Накопление младших разрядов $\sum_{\lambda=1}^{2\nu} H_\lambda$ в ячейке X.
+I0	+''	a_I	β_1	Накопление старших разрядов $\sum_{\lambda=1}^{2\nu} H_\lambda$ в ячейке β_1
+II	+	γ_1	яч. m+8	Формирование команды $N m$+8
+I2	+	a_3	β_2	
+I3	/-/,	РЯ"2ν"	β_2	Условный переход
+I4	/</		$N m$+8	
+I5	ПЧ	σ	d_3	
+I6	X	a_7	d_3	
+I7	X''	d_3	β_1	Умножение "I/2ν" на $\sum_{\lambda=1}^{2\nu} H_\lambda$
+I8	X	d_3	X	
+I9	ПЧ	a_6	β_2	

I	2	3	4	5
+20	X'	a_8	X	
+2I	+	a_9	β_2	
+22	/-/,	β_2	a_I	Условный переход
+23	/</		$N m$ +20	
+24	ПЧ	a_I	ξ	
+25	Печ,			
+26	Ост.			Останов

Программа счета коэффициентов Фурье A_k и B_k

S +I	ПЧ	a_I	d_3	Очистка ячейки d_3
+2	ПЧ	a_I	d_4	Очистка ячейки d_4
+3	ПЧ	a_I	X	Очистка ячейки X
+4	ПЧ	a_I	β_1	Очистка ячейки β_1
+5	ПЧ	H_I	К+2I	Восстановление исхрдного содержимого ячеек МОЗУ N К+2I, N К+20, $N S$ +II, $N S$ +22.
+6	ПЧ	H_4	К+20	
+7	ПЧ	H_3	S +II	
+8	ПЧ	H_3	S +22	
+II	ПЧ	C_I	δ	
+I2	ПЧ	a_{I0}	ξ_δ	
+I3	X	d_1		
+I4	+'''	δ'	ξ	Присвоение знака содержимого яч. δ содержимому яч. ξ
+I5	+'	δ	d_3	Накопление младших разрядов $\sum_{\lambda=1}^{2\nu} H_\lambda \cos k\frac{\pi}{\nu}\lambda$ в яч. d_3
+I6	+''	ξ	X	Накопление старших разрядов $\sum_{\lambda=1}^{2\nu} H_\lambda \cdot \sin k\frac{\pi}{\nu}\lambda$ в яч. X

I	2	3	4	5
+I7	+′	a_I	d_3	
+I8	+″	a_I	X	
+I9	+	γ_1	NS +II	Формирование команды NS ~~+II~~.
+22	ПЧ	C_I	δ	
+23	X	d_2	δ	
+24	ПЧ	a_{I0}	ζ	
+25	+‴	δ	ζ	Присвоение знака содержимого яч. δ содержимому яч. ζ
+26	+′	δ	d_4	
+27	+″	ζ	β_1	
+28	+′	a_1	d_4	
+29	+″	a_I	β_1	
+30	+	γ_1	NS +22	Формирование команды NS + 22
+3I	+	a_3	β_2	
+32	/-/,	РЯ"2Y"	β_2	Условный переход
+33	/</		NS +II	
+34	+‴	X	d_3	Присвоение знака содержимого яч.X содержимому яч. d_3
+35	X″	σ	X	
+36	X	σ	d_3	
+37	+‴	β_1	d_4	Присвоение знака содержимого яч. β_1 содержимому яч. d_4
+38	X′	σ	β_1	
+39	X	σ	d_4	

I	2	3	4	5
	Программа счета амплитуды К-ой гармоники			
d+I	ПЧ	d_3	ζ	
+2	X″	ζ	ζ	Получение $2m$-разрядного произведения $A_к \cdot A_к = A_к2$
+3	X	a_I	β_2	
+4	ПЧ	d_4	β_1	
+5	+‴	a_I	δ	Образование в яч. δ модуля от первоначального содержимого этой ячейки.
+6	X″	β_1	β_1	Получение $2m$-разрядного произведения $B_к \cdot B_к = B_к2$
+7	X	a_I	δ	
+8	+′	β_2	δ	Получение $2m$-разрядной суммы ($A_к2 + B_к2$)
+9	+″	β_1	ζ	
+I0	$\sqrt{\ }$	ζ	δ	Извлечение квадратного корня из $2m$-разрядного выражения ($A_к2 + B_к2$) с точностью до m разрядов
	Программа счета начальной фазы К-ой гармоники			
a+I	ПЧ	a_I	X	
+2	/-/,	a_I	δ	Условный переход на проверку равенства $\mathcal{H}_{mk} = 0$
+3	/=/		C+2	
+4	/-/,	d_4	d_3	Условный переход на проверку равенства $\lvert A_к \rvert = \lvert B_к \rvert$
+5	/=/		Na +32	
+6	/-/,	d_3	d_4	Условный переход на проверку неравенства $\left\lvert \frac{A_к}{B_к} \right\rvert < I$
+7	/</		Na +I0	
+8	ПЧ	d_4	ζ	
+9	:	d_3	ζ	

I	2	3	4	5
+10	/−/	α_4	α_3	Условный переход
+11	/</		N $a+14$	
+12	ПЧ	α_3	ζ	
+13	:	α_4	ζ	
+14	+'''	a_I	ζ	Образование в яч. ζ модуля от первоначального содержимого этой ячейки
+15	ПЧ	ζ	β_2	
+16	/−/	a_{II}	β_2	Условный переход на проверку неравенства $/\frac{A_k}{B_k}/ < 0{,}414$
+17	/</		N $a+23$	
+18	X	a_7	β_2	
+19	ПЧ	β_2	X	
+20	+	a_{12}	X	
+21	+	a_7	β_2	
+22	:	X	β_2	
+23	ПЧ	β_2	X	
+24	X	X	X	
+25	X	a_7	X	
+26	+	a_{13}	X	
+27	:	a_{14}	X	
+28	+	a_{15}	X	
+29	X	β_2	X	
+30	/−/	a_{II}	ζ	Условный переход на проверку неравенства $/\frac{A_k}{B_k}/ < 0{,}414$
+31	/</		N $a+33$	
+32	+	a_{16}	X	
+33	/−/	α_4	α_3	Условный переход на проверку неравенства $/A_k/ < /B_k/$
+34	/</		N $a+36$	

I	2	3	4	5
+35	+	a_{17}	X	
+36	ОК	α_3	α_4	
+37	+'''	a_I	X	Образование в яч.X модуля от первоначального содержимого этой ячейки.

Программа перевода амплитуды и начальной фазы K-ой гармоники из двоичного кода в десятичный и вывод на печать

C+I	+	α_4	X	
+2	ПЧ	H_5	N C+6	Восстановление первоначального содержимого ячейки МОЗУ N C+6.
+ 3	ПЧ	a_{18}	β_2	
+4	ПЧ	a_6	ζ	
+6	X'	a_8	δ	
+7	+	a_9	ζ	
+8	/−/	ζ	a_I	Условный переход
+9	/</		N C+6	
+10	Печ.			
+ II	+	γ_2	N C+6	
+I2	+	a_9	β_2	
+I3	/−/	β_2	a_I	Условный переход
+I4	/</		N C+4	
+I5	Ост.			Останов.

Расположение констант в постоянном запоминающем устройстве

a_1	– "0"	a_{10}	– "0"
a_2	– "10"	a_{11}	– "0,414"
a_3	– "1"	a_{12}	– "–1/2"
a_4	– "5"	a_{13}	– "d_1'"
a_5	– "$10^6 \cdot 2^{-20}$"	a_{14}	– "C_1'"
a_6	– "6"	a_{15}	– "C_0'"
a_7	– "1/2"	a_{16}	– "$\pi/4$"
a_8	– "0,625"	a_{17}	– "$-\pi/2$"
a_9	– "–1"	a_{18}	– "2"

σ – „$\frac{1}{r_i}$" d_1 – „$\cos \Delta\alpha_i \cdot \lambda$"

d_2 – „$\sin \Delta\alpha_i \lambda$"

Формирующие команды γ_1 и γ_2 имеют вид:

γ_1

Коп	A_1	A_2
0000	000000001	000000000

γ_2

К	A_1	A_2
0000	000000000	000000001

Первоначальный вид формируемых команд:

H_1	:	β_1	C_1
H_2	+	b_1	β_1
H_3	ПЧ	C_1	δ
H_4	ПЧ	a_5	C_1
H_5	X	a_8	δ

Расположение информации в МОЗУ

$C_1 \div C_{144}$ – $\mathcal{H}_1$; $в_1 \div в_6$ – адреса двоичных тетрад, хранящихся в устройстве ввода,

$\beta_1, \beta_2, d_3, d_4$, „R.2r" ξ δ X – рабочие ячейки

Литература

1. В.Н. Аносов, Ю.Н. Денисов, Н.И. Дьяков, В.И. Прилипко, Ю.И. Сусов, П.Т. Шишлянников. Препринт ОИЯИ № 10-3002-1, Дубна, 1966.
2. М.А. Гашев и др. Релятивистский циклотрон на энергию 700 Мэв. Труды международной конференции по ускорителям. Дубна, 1963г. Атомиздат, 1964.
3. В.Н. Канунников, А.А. Коломенский и др. Исследования, связанные с разработкой ускорителей типа кольцевых фазотронов. Труды международной конференции по ускорителям, 1959.
4. М.Г. Серебренников. "Гармонический анализ", Гостехиздат, 1948 г.
5. М.А. Карцев. Арифметические устройства электронных цифровых машин, Физматгиз, 1958 г.
6. П.П. Головистиков и др. Арифметическое устройство и устройство управления БЭСМ. Физматгиз, 1960.
7. Э. Бут, К. Бут. Арифметические цифровые машины. ИЛ, 1959.
8. Б.М. Каган, Т.М. Тер-Микаэлян. Решение инженерных задач на электронных вычислительных машинах. Госэнергоиздат, 1958.
9. Цифровая техника и вычислительные устройства. Сборник 3 АН СССР, 1962.
10. А.А. Корнейчук, А.С. Марков, Н.Ю. Ширикова. Препринт ОИЯИ № 2250, Дубна, 1965.

Рукопись поступила в издательский отдел
24 февраля 1967 г.

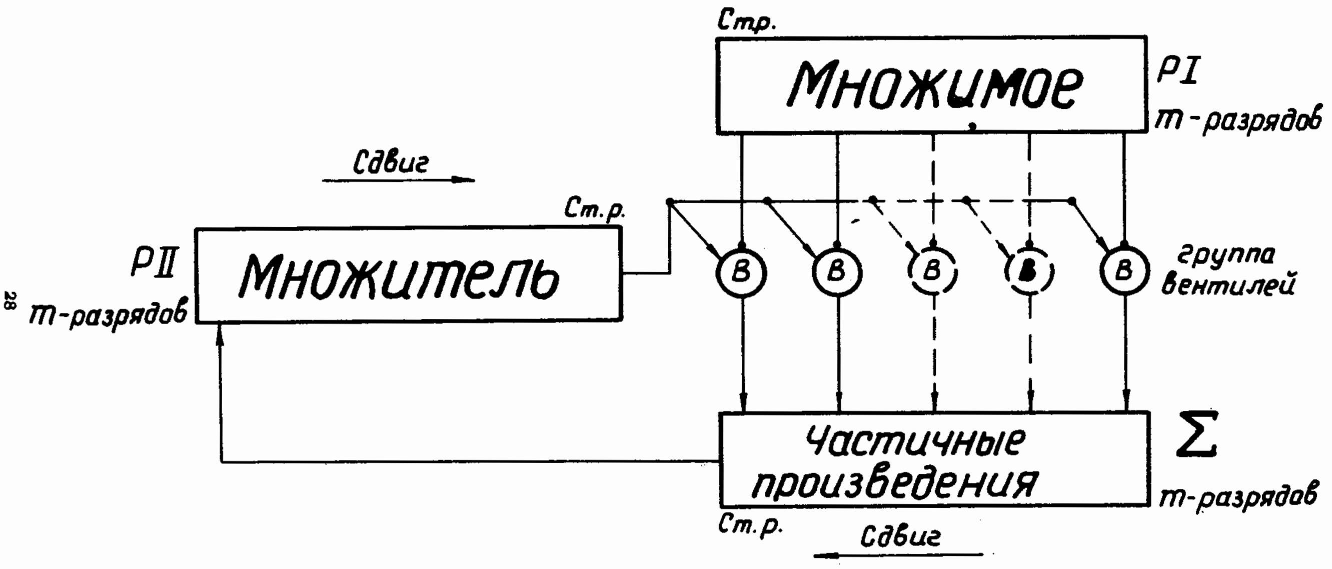

Стр.
Множимое
PI
m-разрядов
Сдвиг
Ст.р.
PII
Множитель
m-разрядов
В
В
В
В
В
группа вентилей
Частичные произведения
Σ
m-разрядов
Ст.р.
Сдвиг

Рис. 1

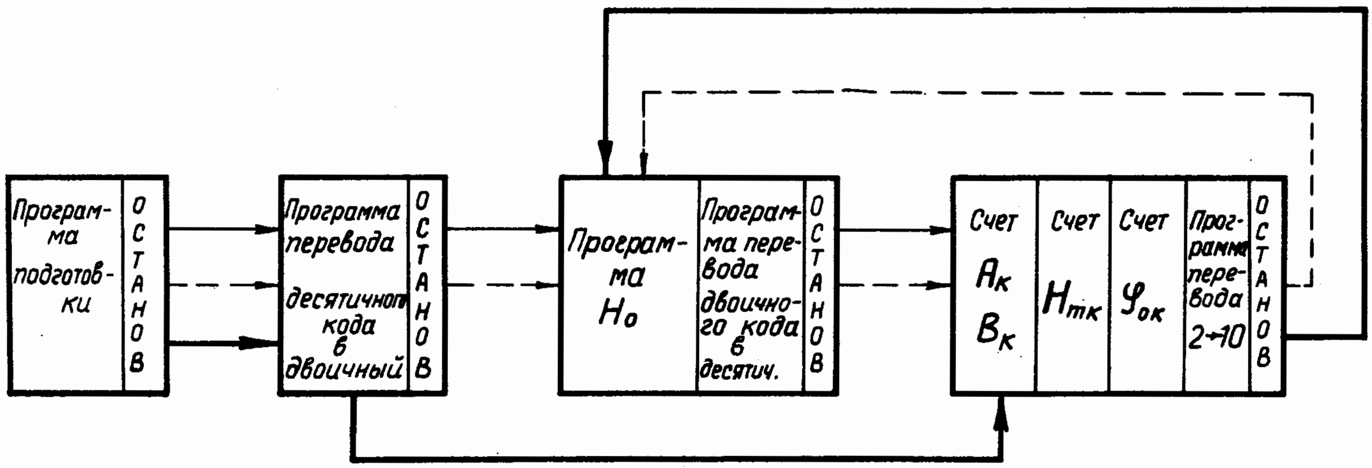
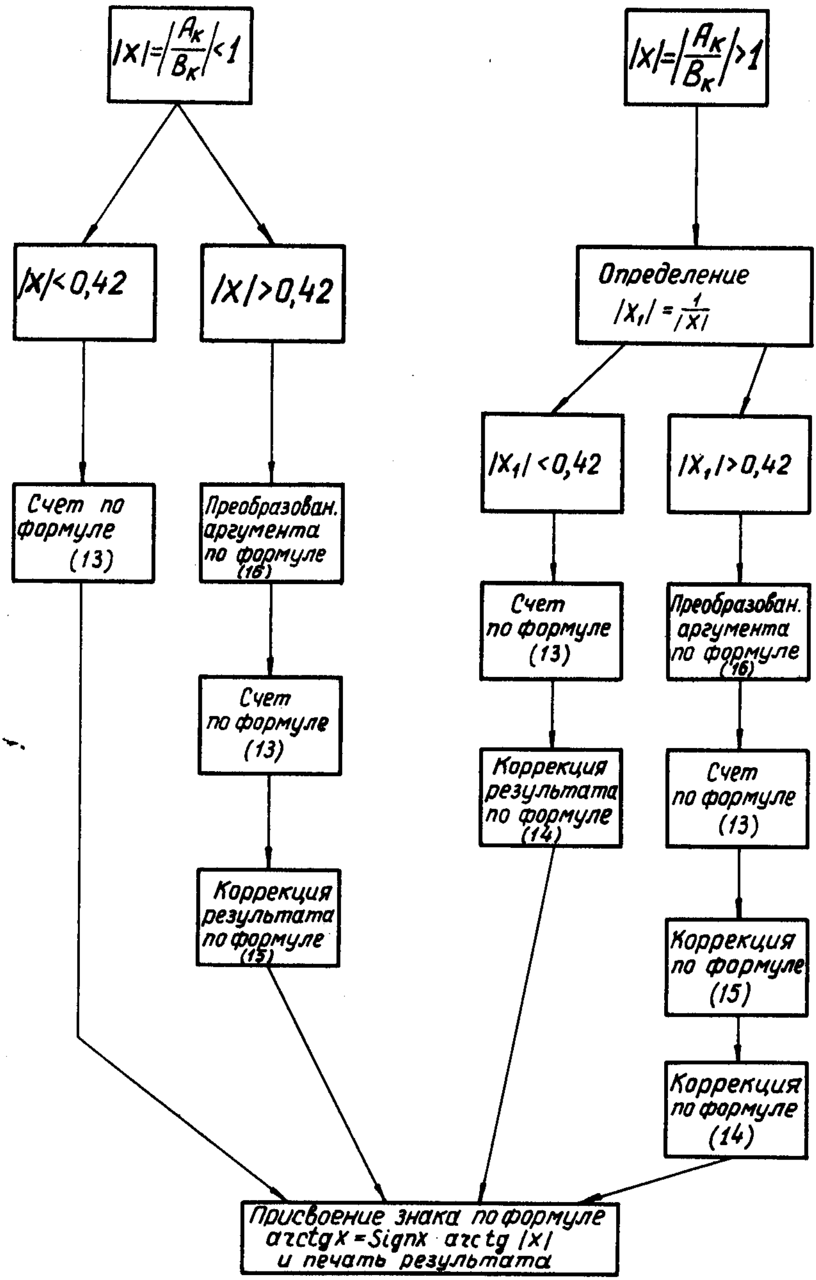

Програм-ма подготов-ки
ОСТАНОВ
Программа перевода десятичного кода в двоичный
ОСТАНОВ
Програм-ма H_o
Програм-ма пере-вода двоично-го кода в десятич.
ОСТАНОВ
Счет $A_к$ $B_к$
Счет $H_{тк}$
Счет $\varphi_{ок}$
Прог-рамма пере-вода 2→10
ОСТАНОВ

Рис. 2


|X|=|Ak/Bk|<1
|X|=|Ak/Bk|>1
|X|<0,42
|X|>0,42
Определение |X1|=1/|X|
|X1|<0,42
|X1|>0,42
Счет по формуле (13)
Преобразован. аргумента по формуле (16)
Счет по формуле (13)
Коррекция результата по формуле (15)
Счет по формуле (13)
Коррекция результата по формуле (14)
Преобразован. аргумента по формуле (16)
Счет по формуле (13)
Коррекция по формуле (15)
Коррекция по формуле (14)
Присвоение знака по формуле arctgX=SignX arctg |X| и печать результата

Р и с. 3