

НЕКОТОРЫЕ ОСОБЕННОСТИ ЛОГИЧЕСКОЙ СТРУКТУРЫ МАШИНЫ ГИФТИ И ПРОГРАММИРОВАНИЯ В ЕЕ КОДЕ. I *

А. М. Гильман

Приведены основные сведения о логической структуре и коде машины ГИФТИ. Одной из основных особенностей кода является специальный способ проведения переадресации. Рассматриваются построение и простейшие преобразования схем-программ в коде машины ГИФТИ.

1. ОСНОВНЫЕ СВЕДЕНИЯ О ЛОГИЧЕСКОЙ СТРУКТУРЕ МАШИНЫ ГИФТИ

Машине ГИФТИ спроектирована для решения инженерно-технических задач средней сложности и некоторых логических задач, возникающих в машиностроительной промышленности. При проектировании предусматривалось, что машина должна быть относительно малоламповой и малогабаритной. Эти требования предопределили тип машины, выбор технических средств, примененных при проектировании, и особенности логической структуры машины [1].

Машине ГИФТИ является машиной последовательного типа и имеет два вида внутреннего запоминающего устройства (ЗУ). Основное ЗУ (ОЗУ) состоит из 1984 ячеек, размещенных на 31 дорожке магнитного барабана, а специальное ЗУ (СЗУ) реализовано на регистрах с рециркуляцией на том же барабане. В каждом регистре имеется две ячейки. Всего в СЗУ 32 ячейки. В каждой ячейке ОЗУ или СЗУ может быть помещено 32-разрядное «слово». Время ожидания выборки в среднем равно времени $1/2$ оборота барабана при поиске в ОЗУ и $1/128$ оборота при поиске в СЗУ. Код машины является одноадресным относительно ОЗУ и трехадресным относительно СЗУ. Операции могут производиться с 32- и 64-разрядными числами. Запятая в коде числа фиксирована. Предусмотрено несколько мест фиксации запятой, в том числе и в середине числа. В основу нумерации ячеек ЗУ положены 32-разрядные ячейки. При операциях с 64-разрядными числами ячейки с нечетными номерами присоединяются к ячейкам с четными номерами и им присваивается общий (обязательно четный) номер. Код команды состоит из 32 двоичных разрядов. Команды можно располагать в любых (четных или нечетных) ячейках ЗУ. При работе с 64-разрядными числами под четным номером может быть указано и число и команда, поэтому в коде команды предусмотрено указание, позволяющее производить выбор.

Одной из важных особенностей кода машины ГИФТИ является принятый в ней способ проведения операций переадресации и восстановления. В переадресуемом приказе, хранящемся в одной из ячеек ЗУ, указывается начальный адрес k_0 . Содержание этой ячейки при проведении переадресации не изменяется. Для накопления произведенений t_i (t — шаг переадресации, а i — параметр, управляющий переадресацией) выделены специальные ячейки, которые мы будем называть ячейками переадресации. Содержание этих ячеек не зависит от расположения переадресуемого приказа в ЗУ. При направлении переадресуемого приказа в регистр текущей команды (РТК) к нему прибавляется содержание ячейки пере-

* Во второй части будут приведены примеры составления программ и некоторые соображения по оценке кода машины ГИФТИ.

адресации и, как и при обычном способе переадресации, переадресуемая часть приказа после помещения ее в РТК будет равна k_0+ti . Константы переадресации помещаются в другую группу специальных ячеек. В коде команды предусмотрены две группы разрядов для управления специальными ячейками.

Приведем разбиение разрядной сетки команды на группы.

| | | |
|-------------------------------|----------------------------------|----------------|
| 32 31 30 29 28 | 27 26 25 24 23 22 21 20 19 18 17 | 16 15 14 13 12 |
| 1 гр. | 2 гр. | 3 гр. |
| 11 10 9 8 7 6 5 4 3 2 1 | | |
| 4 гр. | | 5 гр. |
| 6 гр. | | |

В первой группе указывается код операции *, во второй, третьей и четвертой — соответственно адреса I, II и III, в пятой — код ячеек переадресации, в шестой — управляющий код. Разряд № 1 оставлен пустым по техническим соображениям. В адресе 1 указываются ячейки обоих типов внутреннего ЗУ от 0 до 2047 включительно. В адресах II и III указываются только ячейки СЗУ от 0 до 31 включительно. В качестве ячеек переадресации используются ячейки 24, 25, 26 и 27 СЗУ, которым присвоены соответственно обозначения п/а 1, п/а 2, п/а 3 и п/а 4 и коды 00, 01, 10 и 11, указываемые в группе 5. Таким образом, ячейки переадресации должны иметь двойное управление.

В управляющем коде даются указания по переадресации и выборке чисел (64-разрядных «слов») или команд (32-разрядных «слов»). Если машина настроена на работу с 32-разрядными числами, то производится выборка 32-разрядных «слов» независимо от указаний «операция с числом» или «операция с командой». В таблице дана расшифровка управляющих кодов.

| Код группы № 6 | Указание по переадресации | Указание по выборке чисел или команд |
|----------------|---|---|
| 001 | Произвести переадресацию, т. е. текущую команду до ее выполнения сложить с содержимыми ячейки переадресации, номер которой указан в группе № 5. После выполнения команды к содержимому ячейки переадресации ** прибавить 2_i , если операции ведутся с 64-разрядными числами, или 1_i , если операции ведутся с 32-разрядными числами | В адресах I, II, III указываются ячейки, где хранятся 64-разрядные «слова» (далее будем писать кратко—операция с числом). |
| 101 | Переадресацию произвести, прибавить в ячейку переадресации $2_{I,II}$ | Операция с числом. |
| 011 | Переадресацию произвести, прибавить в ячейку переадресации содержание ячейки № 28 СЗУ ***. | Операция с числом. |
| 110 | Переадресацию произвести, прибавить в ячейку переадресации содержание ячейки № 29 СЗУ ***. | Операция с числом. |
| 010 | Переадресацию произвести. Содержимое ячеек переадресации оставить без изменений. | Операция с числом. |
| 000 | Переадресацию не производить. | Операция с числом. |
| 100 | Переадресацию не производить. | Операция с командой. |
| 111 | Переадресацию не производить. До исполнения текущей команды содержимое ячейки переадресации умножить на нуль. | Операция с командой. |

Перечисленные указания по управлению могут быть даны в коде любой операции.

* В наборе операций имеются основные арифметические и логические операции, операции условной и безусловной передачи управления, специальные операции по формированию конечных команд подпрограмм, по выводу и вводу информации.

** Здесь и в дальнейших аналогичных указаниях речь идет о ячейке переадресации, номер которой указан в группе 5 исполняемой команды.

*** При выполнении указания 011 в группе 5 указывается п/а 1 или п/а 3, а при указании 110 п/а 2 или п/а 4.

2. ПРОГРАММИРОВАНИЕ В КОДЕ МАШИНЫ ГИФТИ

За время, прошедшее с момента появления первых вычислительных машин с программным управлением, была выработана гибкая система программирования, позволяющая при помощи небольшого числа приказов выполнять длинную серию операций по переработке информации [2, 3]. Несмотря на это, составление и проверка программ оказались крайне трудоемким процессом.

В 1953 г. А. А. Ляпуновым были сделаны предложения по выработке математического аппарата для частичной формализации программирования. Предложенный им операторный метод программирования [4, 5, 6] может быть использован для любой машины и позволяет упростить составление и проверку программ. На основе этого метода удалось систематизировать способы управления, применяемые в машинах, и поставить задачу нахождения способов наиболее выгодного построения программ.

Мы будем применять операторный метод, используя как принятые формы символьической записи [4, 5, 6], так и некоторые новые символы, учитывающие специфику кода машины ГИФТИ.

Переадресация и восстановление в машине ГИФТИ могут быть выполнены как за счет указаний, даваемых в группах 5 и 6 некоторых команд, так и отдельной командой. Поэтому операторы $F_A(i)$ и $R_A(i)$ будем понимать как общие указания о необходимости переадресации и восстановления некоторых приказов, входящих в оператор A , при изменении параметра i .

Переадресацию оператора счета A по параметру i с помощью кодов групп 5 и 6 одного из приказов, входящих в A , будем обозначать \bar{A}^i . Если \bar{A}^i , \bar{B}^j , \bar{C}^k переадресующиеся операторы счета, то ячейки переадресации, приданые им, будем обозначать a , b , c , а их содержание $[a]$, $[b]$, $[c]$.

Введем операторы f и r , изменяющие содержание управляющих ячеек отдельным приказом (или группой приказов). Переадресацию оператора счета A путем прибавления в ячейку a содержания ячейки k или ячейки b будем обозначать соответственно $f_A(k)$ и $f_A(b)$. Восстановление содержания ячейки a путем вычитания константы, помещенной в ячейку k , будем обозначать $r_A(k)$, а умножение на нуль (гашение) $[a]$ просто r_A . При написании операторов f и r иногда в скобках будем указывать не номер ячейки, а ее содержание, тогда оно будет ставиться в кавычки. Если гашение ячейки a выполняется при помощи кода 111 в группе 6 или путем вытеснения при выполнении приказа, входящего в некоторый оператор W (он может быть и оператором счета, и управления), то будем пользоваться такой записью: W_a . Отметим, что в машине ГИФТИ можно одним приказом погасить сразу две ячейки переадресации. Гашение ячеек a и b , например, может быть выполнено оператором r_{A_e} .

Будем пользоваться также следующими условными обозначениями. Ячейку, в которой хранятся константа для проверки логического условия p_k , обозначим p_k . Вспомогательные и оперативные ячейки обозначим соответственно буквами v и o . Если содержание некоторой ячейки, например v , изменяется в ходе выполнения программы, то для указания начального содержания будем употреблять запись $[v]_0$. Символическую запись будем употреблять и при описании действия операторов. Например, вместо того чтобы говорить словами: действие оператора f_{p_k} (« n ») заключается в прибавлении n в ячейку p_k , будем писать $f_{p_k}^* \rightarrow n$. Если некоторое число помещается соответственно в группу 2, 3, 4 кода команды, то это будет отмечаться постановкой индекса I, II, III у этого числа. Если по тексту будет ясно, что речь идет о числе, помещенном в определенную группу, то индекс будем опускать.

Работа по составлению программы вычислений по заданному алгоритму в коде машины ГИФТИ состоит из следующих шести этапов: 1) составление проекта размещения числовых информации в ЗУ; 2) составление схемы счета; 3) составление таблицы зависимости операторов счета от параметров; 4) развертывание схемы счета в схему программы; 5) преобразование схемы программы; 6) расписывание программы по приказам.

Мы ограничимся здесь рассмотрением вопросов, связанных с пятым этапом. Наиболее существенные моменты по другим этапам пояснены на примерах, а здесь мы ограничимся следующими указаниями. При составлении схем счета надо стремиться к тому, чтобы количество операторов счета в ней не было меньше числа параметров. В этом случае к каждому оператору счета можно «прикрепить» ячейку переадресации, что обеспечивает более экономную реализацию операторов переадресации. Если в схеме счета имеется выражение $\prod_{i=g(k)}^{i=n} A_i$, где i и k — параметры, а n — константа, то при составлении таблицы зависимости операторов от параметров надо учесть зависимость A от обоих параметров. При развертывании схемы счета в схему программы операторы переадресации будем вводить всегда после соответствующих операторов счета.

Преобразования схем программ имеют целью выбор операторов управления, реализующихся меньшим числом приказов. В коде машины ГИФТИ содержание ячеек переадресации не зависит от размещения переадресуемого приказа в ЗУ. Это открывает некоторые новые возможности для преобразований. Такие преобразования удобно выполнить в несколько шагов.

Первый шаг — введение операторов типа \bar{A} и r_A . Во всех случаях, когда в схеме программы встречается произведение операторов вида $F_A(i)$, к оператору A прикрепляется ячейка переадресации a и произведение $A \cdot F_A(i)$ заменяется оператором \bar{A}^i . Если число таких комбинаций окажется больше числа ячеек переадресации, то схема программы разбивается на участки с меньшим числом комбинаций $A \cdot F_A(i)$. В конце каждого участка содержание ячеек переадресации отправляется на запоминание, а в начале вновь вводится в них. Такой случай встречается ниже, в примере 3.

Восстановление путем умножения на нуль производится в тех случаях, когда оператор восстановления, имеющий индекс a , стоит правее любого другого оператора, имеющего тот же индекс. В этих случаях $R_A(i)$ заменяется на r_A .

Второй шаг — определение содержания ячеек переадресации в момент проверки логических условий. Операторы F и R , оставшиеся в схеме программы после первого шага, определяют всю последовательность операций над содержанием ячеек переадресации и позволяют определить это содержание в момент проверки логических условий.

Рассмотрим достаточно общий случай участка схемы программы, на котором имеются N параметров i_1, i_2, \dots, i_N и операторы, зависящие, вообще говоря, от всех параметров. (Если оператор не зависит от какого-либо параметра, то шаг переадресации его по этому параметру равен нулю.) Вычислим $[a]_k$ — содержание ячейки переадресации a , «привязанной» к оператору счета \bar{A}^s ($s \leq N$) в момент проверки логического условия $p(i_k)$. Для этого перенумеруем все l циклов, охватывающих цикл по параметру i_k , присвоив им номера $k+1, k+2, \dots, k+l$. Будем считать, что при каждом прохождении знака \sqcup_{i_k} счет повторений по i_k начинается заново, т. е. после каждого прохождения $\sqcup_{i_k}, i_k = 0$. В момент прохождения любого знака или оператора, стоящего не-

посредственno после \sqcup_{i_v} , значение i_s увеличивается на единицу. Учитывая, что накопления в $[a]_k$ происходят лишь в результате выполнения операторов переадресации, расположенных в циклах, охватывающих цикл по i_k , и что $[a]_0=0$, получим: если оператор \bar{A}^{is} стоит внутри цикла по i_k , то

$$\begin{aligned}[a]_k = & t_{i_k} [i_k - (i_k)_0 + 1] \cdot 1 [i_k - (i_k)_0 + 1] + \\ & + \sum_{v=k+1}^{v=k+l} t_{i_v} [i_v - (i_v)_0] \cdot 1 [i_v - (i_v)_0].\end{aligned}\quad (2.1)$$

Если цикл по i_s охватывает цикл по i_k , а оператор \bar{A}^{is} расположен правее цикла по i_k , то

$$[a]_k = \sum_{v=s}^{v=k+l} t_{i_v} [i_v - (i_v)_0] \cdot 1 [i_v - (i_v)_0].\quad (2.2)$$

Если цикл по i_s охватывает цикл по i_k , а оператор \bar{A}^{is} расположен левее цикла по i_k , то

$$\begin{aligned}[a]_k = & t_{i_s} [i_s - (i_s)_0 + 1] \cdot 1 [i_s - (i_s)_0 + 1] + \sum_{v=s+1}^{v=k+l} t_{i_v} [i_v - (i_v)_0] \cdot 1 [i_v - \\ & - (i_v)_0].\end{aligned}\quad (2.3)$$

Третий шаг — уточнение констант восстановления и некоторые упрощения схемы программы. Обозначим через $[a]_k$ содержание ячейки a , вычисленное по формулам (2.1) — (2.3) при условии, что $p_n(i_k)=1 (n=1, 2, 3 \dots)$, но без учета накоплений.

Уточнение констант восстановления сводится к определению $[a]$, после чего удается несколько упростить схему программы. Для схем программ характерно наличие таких структур:

$$\dots p_n(i_k) \sqcup_k R_A(i_k) \dots \dots \dots \dots F_A(i_{k+1}) p_{n+1}(i_{k+1}) \sqcup_{k+1} \dots \dots \quad (2.4)$$

Действие оператора $R_A(i_k)$ сводится к вычитанию константы $[a]_k$, а действие оператора $F_A(i_{k+1})$ к прибавлению константы $t_A(i_{k+1})$ в ячейку a . Поместив разность этих констант в ячейку z_A , можно заменить операторы $R_A(i_k)$ и $F_A(i_{k+1})$ оператором $f_A(z_A)$. Если $[z_A]=0$, то операторы $R_A(i_k)$ и $F_A(i_{k+1})$ исключаются. Если число повторений в каком-то цикле зависит от параметров, то при выполнении операторов $R_A(i_k)$ и $F_A(i_{k+1})$ приходится иметь дело не с константами, а с какими-то функциями параметров. $[z_A]$ тоже будет зависеть от этих параметров, и в схему программы необходимо ввести операторы, обеспечивающие соответствующее изменение $[z_A]$. В некоторых случаях удается применить другой прием, который будем называть замещением. Допустим, что на участке схемы (2.4) от $p_n(i_k)$ до $p_{n+1}(i_{k+1})$ имеется несколько операторов и среди них операторы A и B , такие, что $[a]_{k+1}=[b]_{k+1}$. Тогда оператор $f_A(z_A)$, расположенный между $p_n(i_k)$ и $p_{n+1}(i_{k+1})$, можно заменить оператором $f_A(b)_a$. Аналогичные преобразования допустимы и в отношении операторов $F_A(i_k)$ и $r_A(i_k)$.

*Четвертый шаг — «привязка» логических условий к операторам счета и введение операторов f^*_a .* В ячейках переадресации накапливаются величины, пропорциональные числу повторений оператора счета, входящего

го в цикл. Содержание ячеек $[a]$, $[b]$, $[c]$ и т. д. в момент, когда $p_n(i_k) = 1$, будем обозначать $[a]_k$, $[b]_k$, $[c]_k$ и т. д. Для проверки $p_n(i_k)$ используется определенная ячейка переадресации, и у каждого логического условия вместо зависимости от параметра надо указать зависимость от оператора счета.

Пусть условие $p_n(i_k)$ выполняется как $p_n(A)$. Тогда $[p_n] = [a]_k$. Величина $[a]_k$ зависит не только от i_k , но и от параметров при переадресации, по которым происходит накопление. В схему программы надо ввести операторы $f_{p_n}^*$ по этим параметрам, разместив их после соответствующих знаков \perp . В большинстве случаев удается „привязать“ логические условия к операторам так, что $[a]_k$ зависит только от одного накапливающего параметра, т. е. $[a]_k = x_1 + x_2 i_k + x_3 i_s$, а $[a]_k = x_1 + x_2 m + x_3 i_s$, где x_1 , x_2 , x_3 и m – константы. Тогда $[p_n] = x_1 + x_2 m$, а $f_{p_n}^* \rightarrow x_3$ и проверка $p_n(A)$ сводится к проверке того, что $[a] = [p_n]$. Если же $x_2 > x_3 \neq 0$, и $i_s \leq m$, то $x_1 + x_2 m \geq x_1 + x_3(m-1) + x_3 i_s$, и оператор $f_{p_n}^*$ не вводится, а проверка $p_n(A)$ сводится к проверке $a > [p_n]$, где $[p_n] = x_1 + x_2 m$.

Заканчивая описание преобразований, отметим, что приведенные рекомендации не претендуют на полноту. Так, например, возможны преобразования, связанные с автоматическим изменением кодов операций, в частности при суммировании знакопеременных рядов. Для того чтобы обеспечить чередование операций сложения и вычитания, коды этих операций (k_+ и k_-) выбраны в машине ГИФТИ так, что $k_+ = k_- + 2^n$, где n – число разрядов в коде команды. Так как код операции занимает в коде команды крайние левые разряды, то переносы в разряд $n+1$ не аккумулируются. Поместив в ячейку переадресации число 2^n , получим нужное чередование кодов операций.

В заключение коротко рассмотрим основные вопросы, связанные с построением стандартных подпрограмм, которые предполагается широко использовать при программировании на машине ГИФТИ. При построении таких подпрограмм желательно разместить в СЗУ всю подпрограмму, а также вспомогательные и оперативные ячейки. Если число ячеек СЗУ оказалось недостаточным для размещения всей подпрограммы, то в СЗУ помещается часть подпрограммы с максимальной повторяемостью. При этом в подпрограмму надо ввести команды безусловной передачи управления, обеспечивающие передачу управления от одной части подпрограммы к другой.

Кроме того, в подпрограмму вводятся приказы, формирующие вспомогательные числа, и приказы, обеспечивающие передачу группы приказов из ОЗУ в СЗУ. Если в программе есть приказы, передающие управление приказам, передаваемым в СЗУ, то надо подменить номера отсылок, заменив номера ОЗУ соответствующими номерами СЗУ.

3. ОСНОВНЫЕ ПРИНЦИПЫ ПОСТРОЕНИЯ ЦЕНТРАЛЬНОГО УПРАВЛЕНИЯ МАШИНЫ ГИФТИ

Центральное управление (ЦУ) состоит из набора генераторов управляющих импульсов и набора функциональных устройств. В машине имеются следующие генераторы управляющих и синхронизирующих сигналов:

1. Генератор тактовой частоты, построенный на базе 2048 импульсов, записанных на одной из дорожек магнитного барабана, и дающий на выходе непрерывную серию тактовых импульсов t с расстоянием между ними 5 мксек.

2. Генератор импульсов окончания малого цикла и цикла. Тактовые импульсы разбиваются на группы по 32 и 64 импульса. Время прохождения 32-тактовых импульсов называется малым циклом, а время прохождения 64-тактовых импульсов — циклом. Генератор импульсов окончания цикла построен на базе триггера, на счетный вход которого подаются импульсы t_{32} , записанные на одной из дорожек магнитного барабана (следующие друг за другом через 0,16 мсек) и фиксирующие окончание малого цикла. Выходы триггера, кроме того, используются для индикации четности и нечетности номера малого цикла относительно начала отсчета.

3. Генератор импульса начала отсчета. На одной из дорожек магнитного барабана наносится одиночный импульс, совпадающий по образующей с одним из t_{32} . Этот импульс используется как отсчетный для нумерации ячеек.

4. Генератор управляющих импульсов. Для управления различными функциональными устройствами необходимо иметь большинство импульсов, входящих в малый цикл. При генерации этих импульсов используется линия задержки, составленная из динамических триггеров, на вход которой подаются t_{32} . На выходе n -го динамического триггера получается импульс t_n ($1 \leq n \leq 31$). Совпадение t_n и сигнала «нечет» дает t_{n+32} .

5. Генератор потенциалов главного цикла. Главным циклом называется время, необходимое для выполнения одного приказа. Это время в машине ГИФТИ не фиксировано и зависит от выполняемой операции, расположения чисел и т. д. В качестве генератора потенциалов главного цикла используется самовосстанавливающийся счетчик с дешифратором. На вход счетчика подаются t_{32} или импульсы завершения определенных действий. Выходы дешифратора поочередно активизируются и дают серию следующих один за другим потенциалов. Восстановление исходного состояния счетчика происходит в момент перехода к новому главному циклу.

В машине ГИФТИ имеются следующие функциональные устройства:

1. Счетчик команд (СК). Устройство для запоминания номера команды, которая должна быть выполнена в данном главном цикле.

2. Регистр текущей команды (РТК). Устройство для запоминания кода команды, которая должна быть выполнена в данном главном цикле. СК и РТК реализованы в виде регистров с рециркуляцией на магнитном барабане.

3. Устройство для запоминания кода операции (РКО), кода ячейки переадресации и управляющего кода переадресации (РК п/а) и выделения сигналов, соответствующих данной операции, данному управляющему коду и т. д. Эти устройства выполнены в виде цепочки статических триггеров, соединенных в сдвиговый регистр с дешифраторами.

4. Схема предварительного определения характера поисков (СПО), которая в момент, предшествующий поиску, определяет, в каком ЗУ будет производиться поиск, и генерирует нужный сигнал.

5. Регистр поиска (РП), т. е. устройство для запоминания номера дорожки, если поиск происходит в ОЗУ, или запоминания номера числа, если поиск происходит в СЗУ, и выделения сигналов, соответствующих данному номеру. Схемно РП выполнен так же, как и РКО.

6. Схема совпадения номера числа в ОЗУ, которое необходимо найти, и номера числа, начала которого в данный момент подошло к магнитной головке.

7. Генератор сигналов разрешения записи или чтения (РЗЧ) и разрешения операций, связанных с переадресацией (РЗ п/а). Эти генераторы выполнены на статических триггерах.

8. Схемы подключения сумматора к РТК и СЗУ для выполнения сложений, связанных с проведением переадресации.

В ходе исполнения одного приказа необходимо произвести следующие действия: 1) по номеру команды, хранящемуся в СК, найти ее и передать в РТК; 2) при наличии соответствующего указания в группах 5 и 6 сложить РТК с содержанием одной из ячеек переадресации и результат направить в РТК; 3) произвести поиск числа, указанного в адресе 1-ом, и передать его в АРУ; 4) произвести поиск числа, указанного в адресе 2-ом, и передать его в АРУ; 5) произвести операцию, код которой указан в РКО; 6) произвести отсылку результата из АРУ в ЗУ; 7) прибавить единицу в СК; 8) если производилась операция передачи управления и выяснилась необходимость передачи, то код 1-го адреса РТК передать в СК; 9) при наличии соответствующего указания в группах 5 и 6 прибавить в ячейку переадресации константу переадресации.

Выполнением этих действий управляет генератор потенциалов главного цикла, вырабатывающий последовательно 8 потенциалов: П₁, П₂, ..., П₇, П₈. Номера потенциалов и вышеперечисленных действий с 1 по 7 совпадают, указания 8 и 9 выполняются во время действия потенциала П₈.

Продолжительность действия отдельных потенциалов главного цикла, за исключением П₇, заранее неизвестна. Автоматизм работы машины достигается за счет того, что переход от потенциалов П₁, П₃, П₄ и П₆ к следующим происходит после поступления на вход генератора главного цикла импульсов завершения поиска в ЗУ, переход от потенциалов П₂ и П₈ — после поступления импульсов завершения операций, связанных с переадресацией, переход от потенциала П₅ к П₆ происходит после поступления импульса завершения операции в АРУ. Все эти импульсы синхронизованы с началом цикла. Относительно продолжительности действия отдельных потенциалов укажем, что если поиск происходит в ОЗУ, то его средняя продолжительность 32 малых цикла, т. е. приблизительно 5 мсек. При определении времени поиска в СЗУ и времени операции по переадресации следует учесть, что в СЗУ период рециркуляции равен циклу, а работа машины синхронизируется малыми циклами. Поэтому будем считать, что среднее время поиска в СЗУ равно 1,5 малых цикла,

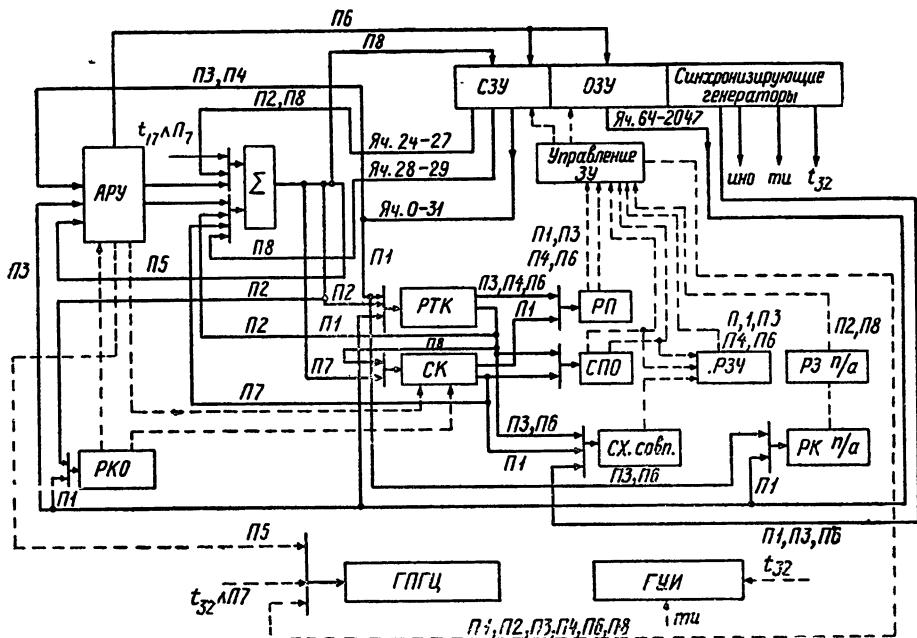


Рис. 1

т. е. 0,24 мсек. Говоря о продолжительности главного цикла в целом, отметим, что выработка потенциалов, ненужных для проведения данной операции, блокируется, например, если переадресация не проводится в данном приказе, то блокируется выработка потенциалов П₂ и П₈.

На рис. 1 изображена блок-схема машины ГИФТИ. На этой схеме управляющие сигналы показаны пунктирными линиями, пути передач числовoy информации и кодов команд — сплошными. Около линии поставлены номера потенциалов, которые показывают, при наличии каких сигналов активизируются те или иные связи.

ЛИТЕРАТУРА

1. А. М. Гильман. К проекту вычислительной машины последовательного типа. Конференция «Пути развития советского математического машиностроения и приборостроения», Секция универсальных цифровых машин, I, М., 1956.
2. Х. Рутисхаузер, А. Шпейзер, Э. Штифель. Вопросы ракетной техники, 1952, вып. 2—5.
3. Л. А. Люстерник, А. А. Абрамов, В. И. Шестаков, М. Р. Шуря-Бура. Решение математических задач на автоматических цифровых машинах, 1952.
4. А. А. Ляпунов. УМН, 1955, 10, вып. 3 (65), 188.
5. А. И. Китов. Электронные цифровые машины, изд-во «Советское радио», 1956.
6. А. А. Ляпунов. См. настоящий журнал, стр. 106—109.

Исследовательский
физико-технический институт
при Горьковском университете

Поступила в редакцию
19 июня 1957 г.