

5. Дальнейшее развитие программных средств терминала может идти, во-первых, по пути организации подкачки с использованием внешней памяти (например, НМЛ), во-вторых, при возможности расширения МОЗУ, за счет передачи всех функций по первичной обработке графической информации с ведущей ЭВМ на терминал и, в-третьих, по пути перехода от языка диалога низшего уровня к языку более общего характера.

ЛИТЕРАТУРА

1. А. М. Ковалев, В. А. Котов, А. И. Лубков, А. С. Токарев. Графический дисплей «Дельта». — Автометрия, 1974, № 4.
2. А. Н. Гинзбург, Ю. И. Родионов. Графическая система дисплей «Дельта» — ЭВМ «Минск-32». Программное обеспечение. — Автометрия, 1974, № 4.
3. Ю. И. Родионов. Шаговый транслятор с графического макроассемблера. — В сб. «Вопросы построения системы сбора и обработки данных». Новосибирск, 1973.
4. А. Н. Гинзбург, Ю. И. Родионов. Структура программного обеспечения системы «Экран». — Автометрия, 1973, № 2.
5. J. H. Sexton. An Introduction to Data — Structures with Some Emphasis on Graphics. — Computer Bulletin, September, 1972.

Поступила в редакцию 27 декабря 1973 г.

УДК 681.3.06

А. Н. ГИНЗБУРГ, Ю. И. РОДИОНОВ

(Новосибирск)

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ГРАФИЧЕСКОЙ СИСТЕМЫ ДИСПЛЕЙ «ДЕЛЬТА» — ЭВМ «МИНСК-32»

Введение. Графическая система дисплей «Дельта» — ЭВМ «Минск-32», являясь универсальной по составу технических средств, предназначена для оперативного взаимодействия исследователя с системой автоматизации экспериментов; цифрового моделирования в режиме графического взаимодействия с ЭВМ, а также для решения широкого круга задач автоматизации проектирования (в микроэлектронике и т. п.).

Основные компоненты системы:

универсальная вычислительная машина «Минск-32», которая используется (в связи с тем, что она имеет достаточно емкую внешнюю память, большое быстродействие и развитое математическое обеспечение) как центральный вычислитель или ведущая ЭВМ;

графический дисплей «Дельта» [1], использующий в качестве дисплейного процессора стандартную мини-машину «Электроника-100»;

универсальное математическое обеспечение (МО).

Говоря об универсальности, мы, прежде всего, имеем в виду независимость комплекса программ от типа ведущей ЭВМ, а также и то обстоятельство, что состав программного обеспечения не накладывает принципиальных ограничений на класс доступных для решения задач. Наличие в системе двух универсальных вычислительных машин привело к тому, что программное обеспечение подразделяется на две части. Наиболее рациональным, по нашему мнению, было бы следующее распределение математического обеспечения. В мини-машине расположены все программы, выполняющие графические функции (формирование изо-

бражений, их модификация и т. п.) и обеспечивающие диалог оператора с ЭВМ. Ведущая машина, таким образом, освобождается от рутинной работы и используется по своему прямому назначению в качестве универсального вычислителя для решения проблемных задач. Однако в реализованном варианте из-за недостаточной емкости запоминающего устройства (4К 12-разрядных слов) в ЭВМ «Электроника-100» размещаются лишь автономный диспетчер терминала (АДТ), подробное описание которого содержится в [2], программа обмена с ведущей ЭВМ и массив индикации. Поэтому значительная часть работы, связанной с обработкой графической информации, возлагается на ведущую ЭВМ. Настоящая работа посвящена обсуждению комплекса программного обеспечения графической системы, размещенного в ведущей ЭВМ.

Общее описание программного обеспечения. Схема организации программ графического комплекса приведена на рис. 1. В верхней части рисунка показаны основные информационные массивы, внизу — основные программные блоки. В левой части рисунка обозначен комплекс МО ведущей ЭВМ, посередине — МО дисплейного процессора «Дельты». Справа изображен оператор с набором устройств, необходимых для приема и передачи графической и символьной информации. Сюда относятся функциональная клавиатура (ФК), буквенно-цифровая клавиатура (БЦК), световое перо (СП) и некоторые внешние устройства (ВнУ) терминала.

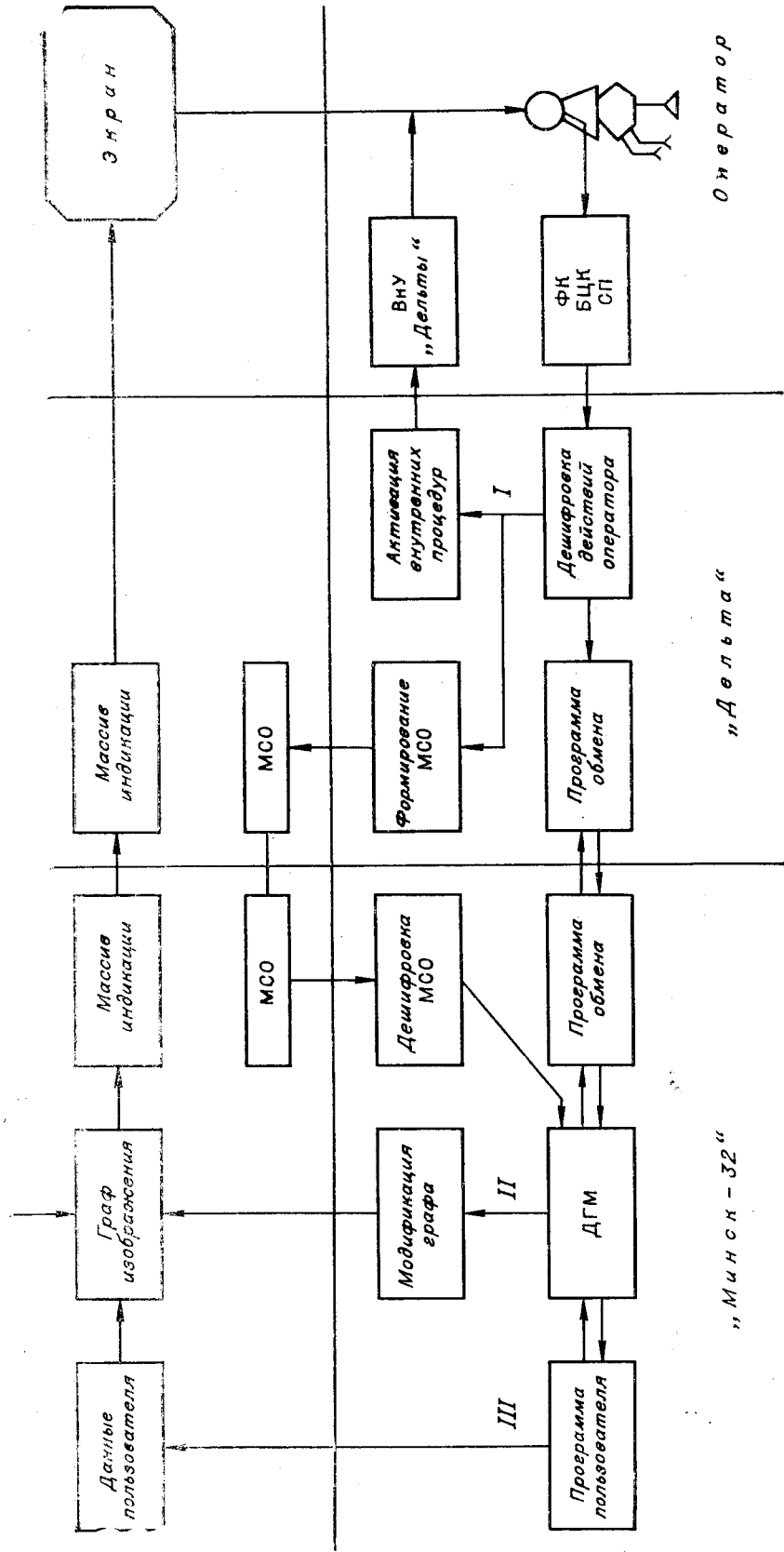
Предложена следующая логика обмена. Программа пользователя посредством набора графических процедур формирует описание изображения в виде графа на специально разработанном языке высокого уровня *GL1*. Далее с помощью диалогового графического монитора (ДГМ) производится трансляция описания изображения в массив индикации, который передается в терминал для отображения на экране дисплея и последующего редактирования изображения оператором. Кадр, принятый из ведущей ЭВМ, при этом не меняется, а редактирование сводится к добавлению новых линий, символов и т. п., которые высвечиваются и одновременно заносятся в специальный массив сообщений оператора (МСО), а также к занесению в МСО добавочной информации (указание пером, буквенно-цифровая информация и т. п.). Отредактированный МСО передается в ведущую ЭВМ и дешифруется с помощью ДГМ. ДГМ осуществляет все операции по модификации описания изображения, замыкая вторую цепь, по которой циркулирует информация. Первая цепь связана с обслуживанием автономных функций терминала, которые за счет возможностей параллельной работы могут выполняться и в режиме связи с ведущей ЭВМ. Это ввод — вывод на перфоленду, ввод — вывод на телетайп и т. п. Те компоненты сообщения, которые ДГМ расшифровать не в состоянии, он передает на расшифровку программе пользователя, которая, таким образом, не лишается возможности самостоятельно модифицировать изображение.

Получая информацию от ДГМ, программа пользователя может принимать решения о модификации своих собственных данных, о переходе к описанию изображения на языке *GL1* и воздействовать на изображение косвенным образом.

В результате появляется третья, последняя цепь циркуляции информации (на рис. 1 цепи обозначены римскими цифрами). При расшифровке действий оператора, а также при расшифровке МСО производится проверка на противоречивость сообщения. В случае ошибки соответствующая информация выводится на печать, а ошибочная компонента сообщения (или группа компонент) игнорируются.

Программное обеспечение, размещенное в ведущей ЭВМ, включает в себя следующие основные части.

1. Программы для организации связи ведущей ЭВМ с дисплейным терминалом.



Оператор

«Дельта»

«Минск-32»

Рис. 1.

2. Программы-генераторы графической информации, с помощью которых формируются такие элементы, как точки, векторы, тексты, маркер, оси координат и т. п.

3. Программы расшифровки и интерпретации МСО.

4. Административные программы, обеспечивающие выделение памяти и ее использование, операции по наименованию объектов и организации подпрограмм.

5. Программы для формирования изображений сложной структуры*, манипуляций с фрагментами изображений, трансляции с языка описания графических данных в массив индикации. Сюда относятся программы, создающие блоки (фрагменты) как с чисто графическими данными, так и включающие ссылки на подблоки (глубина вложения до 10 уровней); программы для удаления, перемещения, копирования, масштабирования, мультиплицирования, зеркального отображения, поворотов и т. п. фрагментов; вырезания окна; цикл программ для перевода с языка описания блоков графической информации в дисплейный файл.

6. Отладочные программы.

7. Специализированные прикладные программы для решения конкретных задач в режиме взаимодействия (прикладные задачи).

Перейдем к подробному рассмотрению наиболее важных, с нашей точки зрения, групп программ.

Программы для организации связи ЭВМ «Минск-32» — «Дельта» должны обеспечивать прием — передачу массивов данных с рабочей или служебной информацией, перевод данных в нужный формат, размещение их на требуемом месте в памяти или на внешних носителях, расшифровку служебной информации и запуск нужной программы обработки. К ним относятся.

Драйвер «Электроника-100» (программа низшего уровня иерархии, осуществляющая непосредственно обмен). Эта программа содержит команды ВВИ, работающие в стандартных тактах при установках, сопровождающихся обязательными прерываниями по началу и концу обработки команды на третий диспетчерский уровень. Кроме того, драйвер содержит ветви с опросом указателей интерфейса и в случае сбоя выдает сообщения на системный пульт оператора.

Форматор. Обмен ЭВМ «Электроника-100» — ЭВМ «Минск-32» ведется семиразрядными символами с контролем по четности. Форматор переводит принятый массив информации в формат, стандартный для ЭВМ «Минск-32».

Монитор. Эта программа на основании анализа служебной информации принимает решение о возобновлении работы прикладной задачи. Задача может состоять из нескольких подзадач, доступных монитору, которые хранятся в банке данных во внешней памяти. В мониторе имеется каталог банка, где перечислены имена массивов и место их размещения. Монитор, расшифровав служебную информацию, передает управление нужной секции (подзадаче) прикладной задачи. Язык монитора — это расширение ФОРТРАНа исполнительными макрооператорами.

Графические программы — это, прежде всего, набор процедур для формирования любых дисплейных кодов. За счет расширения более сложными графическими форматами (например, вывод текста, графиков, осей, координат, маркера) в совокупности с некоторыми административными программами они позволяют сконструировать программное обеспечение для работы с неструктурированными изображе-

* Выражения «изображения сложной (иерархической) структуры», которые будут неоднократно употребляться в тексте, используются в смысле, принятом в [3] (см. также [4]).

ниями. Для организации структурированных изображений (ниже везде речь идет именно о таких) достаточно использовать следующие графические подпрограммы: *VECT* — отобразить отрезок; *TEXT* — отобразить текст; *MOVE* — переместить луч без высвечивания; *MARK* — отобразить маркер.

Программы расшифровки МСО обеспечивают возможность достаточно изолированного диалога «человек — ЭВМ». МСО состоит из собственно сообщения оператора (*MESSAG*) и таблицы сообщения (*MSGTBL*) и формируется в терминале оператором. Сообщение оператора представляет собой последовательность компонент, каждая из которых может быть: а) сообщением от светового пера (адрес выделенного элемента и набор адресов подпрограмм при работе со вложенными структурами); б) кодом функциональной клавиатуры; в) строкой алфавитно-цифровых символов; г) строкой векторов; д) строкой точек; е) признаком «конец сообщения».

Для удобства расшифровки сообщения (и в связи с тем, что длина некоторых компонент не определена) к нему добавляется таблица сообщения, в которой о каждой компоненте имеется дополнительная информация: тип компоненты, ее начальный адрес в сообщении, длина компоненты.

Длина сообщения может быть не более 150 элементов, количество компонент сообщения — не более 20. МСО доступен любой программе, написанной на ФОРТРАНе через общий блок *COMMON/MESAGE/MSGTBL* (3,20), *MESSAG* (150).

Оператор, формирующий МСО, естественно, имеет возможность его редактировать. Полностью сформированное сообщение пересылается из терминала в ведущую ЭВМ для анализа и принятия решения.

Пример МСО дан в таблице.

Комплекс программ для формирования сложных изображений основан на представлении графической информации с помощью структуры типа «сеть» [3], что дает пользователю аппарат для работы с иерархическими изображениями. Как и в [4], изображению ставится в соответствие граф, в котором оконечные узлы содержат собственно графическую информацию, а остальные узлы вместе со связями служат для организации всех фрагментов в единое целое и привязки их к нужным точкам экрана. Для каждого узла имеются списки входящих и выходящих ветвей. Вся графическая информация записывается в массивах, запасенных в специальном *COMMON*-блоке. Каждое слово массива делится на три байта, но сами байты уже не связаны в списки. При этом значительно ускоряется обработка информации по сравнению с комплексом, приведенным в [4], хотя оказывается несколько сложнее организовать динамическое распределение памяти. Приняты две формы представления графической информации в памяти ЭВМ «Минск-32»:

на языке высокого уровня *GL1* информация размещается в *COMMON*-блоке/*GRAPH/NODE* (3,50), *BPAN 1* (3,50), *BPAN 2* (50), *MODE* (2,20), *PG* (400) (рис. 2);

на языке команд дисплея «Дельта» (массив индикации) информация размещается в *COMMON*-блоке/*DELTA/DF* (700).

В блоке *GRAPH* выделены отдельно массив узлов *NODE* (3,50), допускающий запись до 50 описаний фрагментов изображений, *BPAN1* (3,50) — массив выходящих ветвей (запись до 50 копий фрагментов), *BPAN2* (50) — массив входящих ветвей, *MODF* (2,20) — массив модификаций (до 20 фрагментов или их копий могут модифицироваться одновременно), *PG* (400) — массив страниц графической информации, *DF* (700) — массив для записи оттранслированного дисплейного файла (длина ~ 2100 байт).

Коротко опишем структуру массивов.

Массив узлов. Узел на графе изображения соответствует понятию

Номер компоненты	Таблица сообщения		Номер элементов	Сообщение
2	Код БЦК	3 — признак	5	E
		2 — начало	6	L
		8 — длина	7	T
			8	A
			9	—
3	Точки	5 — признак	10	X_1
		10 — начало	11	Y_1
		6 — длина	12	X_2
			13	Y_2
			14	X_3
			15	Y_3
4	Векторы	4	16	X_1
		16	17	Y_1
		4	18	X_2
			19	Y_2
5	Перо	1	20	Начало пера Адрес начала блока и вложенных подблоков при работе со структурированными изображениями
		20	21	
		11	22	
			23	
			24	
			25	
			26	
			27	
			28	
			29	
			30	
6		
..		
17	Конец	0	91	Первый свободный элемент в сформированном сообщении
		91	150	
		0		
18				
19				
20				

фрагмента (блока) изображения. Для его записи используются три элемента массива $NODE(3,50)$ (см. рис. 2, а).

Массив выходящих ветвей. Выходящая ветвь (см. рис. 2, б) соответствует понятию копия фрагмента. Ветви, выходящие из некоторого узла, объединяются в кольцевой список.

Если копия не модифицирована, то соответствующий байт-модификатор пуст; в противном случае в нем хранится ссылка на модификатор

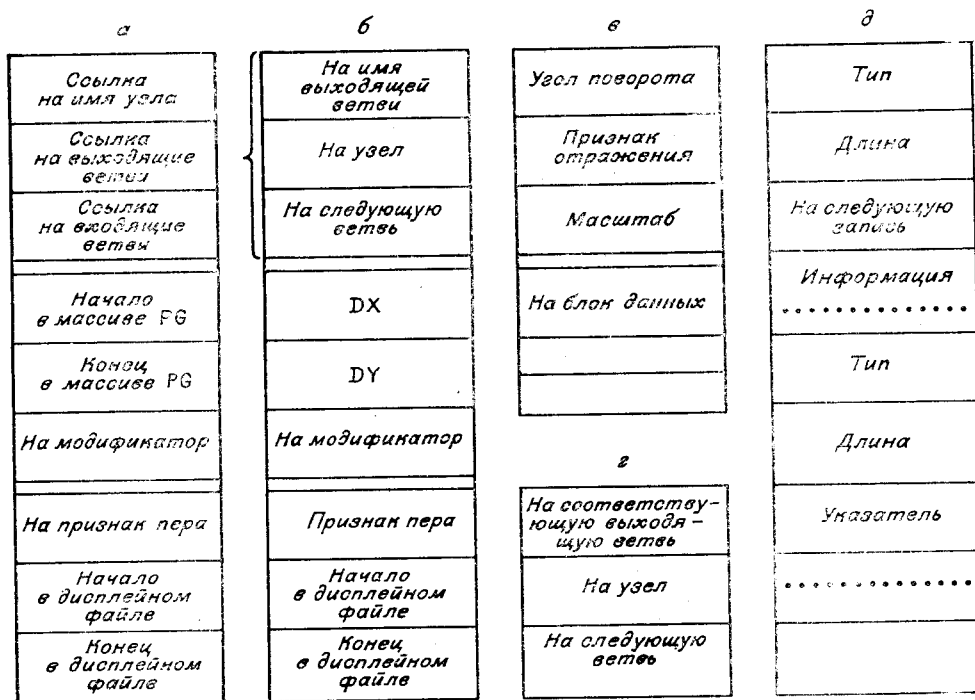


Рис 2.

(см. рис. 2, в). Признак пера — это двоичный шестизрядный код. Назначение разрядов следующее:

- α_1 — погасить объект — высветить объект;
- α_2 — объект мерцает — объект не мерцает;
- α_3 — метки нужны — метки не нужны;
- α_4 — перо активировано — перо деактивировано;
- α_5 — пером можно пользоваться — пером нельзя пользоваться;
- α_6 — объект модифицирован — не модифицирован.

Ссылки на начало и конец в дисплейном файле нужны для облегчения трансляции при модификации изображения (для того чтобы избежать трансляции всего изображения целиком).

Массив входящих ветвей. Входящая ветвь (см. рис. 2, г) имеет указатель на узел, из которого идет ссылка на подблок, в остальном она не содержит новой информации относительно соответствующей выходящей ветви. Однако использование списка входящих ветвей значительно облегчает организацию алгоритмов для исключения подблоков или блоков изображения из структуры.

Массив страниц графической информации. Оконечные узлы на графе имеют ту же структуру, что и остальные узлы, но в их записи есть две особенности (см. рис. 2, а): 1) ссылка на блок выходящих ветвей пуста; 2) есть две добавочные ссылки: на начало и конец в массиве *PG*. В массиве *PG* (см. рис. 2, д) накапливается соответственно графическая информация (уже не структурированная).

При записи графической информации указатель служит для организации списка. «Тип» определяет тип информации (текст, векторы и т.п.) и длину записи.

Приведем перечень типов возможных записей: 1 — запись текста, 2 — запись линии, 3 — запись невысвечиваемой линии, 4 — запись маркера, 0 — запись текста дисплейной команды в коде «Дельта», 7 — запись признака «Конец страницы с графической информацией».

A, 4004	}	Вход в блок	}
0000			
4003	}	Деактивация пера (если требуется)	
4026			
4004	}	Входная метка (если требуется)	
+2			
0			
	}	Признак входной метки	
4023			
4040	}	Установка начальных координат	
<i>DX</i>			
<i>DY</i>			
4006	}	Обращение к <i>L</i> -й графической подпрограмме (подблоку)	
<i>L+1</i>			
4023	}	Возврат в начало блока (по координатам)	
- <i>DX</i>			
- <i>DY</i>			
4026	}	Выходная метка подблока	
4004			
+2			
1			
	}	Признак выходной метки	
...			
	}	Другие подблоки	
4004			
A	}	Выход из блока	

Рис. 3

Массив модификаторов (см. рис. 2, в). Возможны следующие модификации изображения: смещение фрагмента; включение и исключение фрагмента в структуру, объединение фрагментов; копирование, поворот, отражение, масштабирование, мультиплицирование, окно; работа с пером; модификация графического элемента.

Массив имен. Массив имен западается в *COMMON*-блоке *NAMES*: *COMMON/NAMES/NAM1(50), NAM2(50), NN1, NN2*.

Здесь

NAM1 — массив, содержащий имена блоков;

NAM2 — массив имен подблоков (ветвей);

NN1, NN2 — счетчики, указывающие на первую свободную строку в массиве *NAM1* и *NAM2* соответственно.

Использование имен облегчает создание процедур, использующих поиск фрагмента в структуре.

Массив индикации (дисплейный файл). В связи с тем что состав дисплейных команд позволяет формировать подблоки изображений (дисплейные подпрограммы), структура массива индикации, соответствующая структурированному массиву, описывающему изображение на языке *GL1*, оказывается достаточно сложной. Принятая структура для блока в массиве индикации показана на рис. 3; вся необходимая содержательная информация приведена в комментариях; слева приведены соответствующие коды команд дисплея «Дельта» [1].

В принятой структуре массива индикации модификация фрагмента реализуется достаточно просто. Так, для смещения фрагмента достаточно изменить координаты на входе и выходе из подпрограммы. Реализация большинства других манипуляций с фрагментами аналогична.

При работе со структурированным изображением предварительно с помощью процедуры *BEGIN* происходит установка всех переменных в исходное состояние. Затем начинается формирование графических подблоков из элементов изображения с постепенным продвижением к вершине графа (к полному изображению). На этом этапе наиболее сложной оказывается процедура *ADDBLK(NAMBL, NAMBR, X, Y)* — включения уже сформированных подблоков в текущий блок. (Эта процедура включает подблок с именем *NAMBL* под именем *NAMBR* в те-

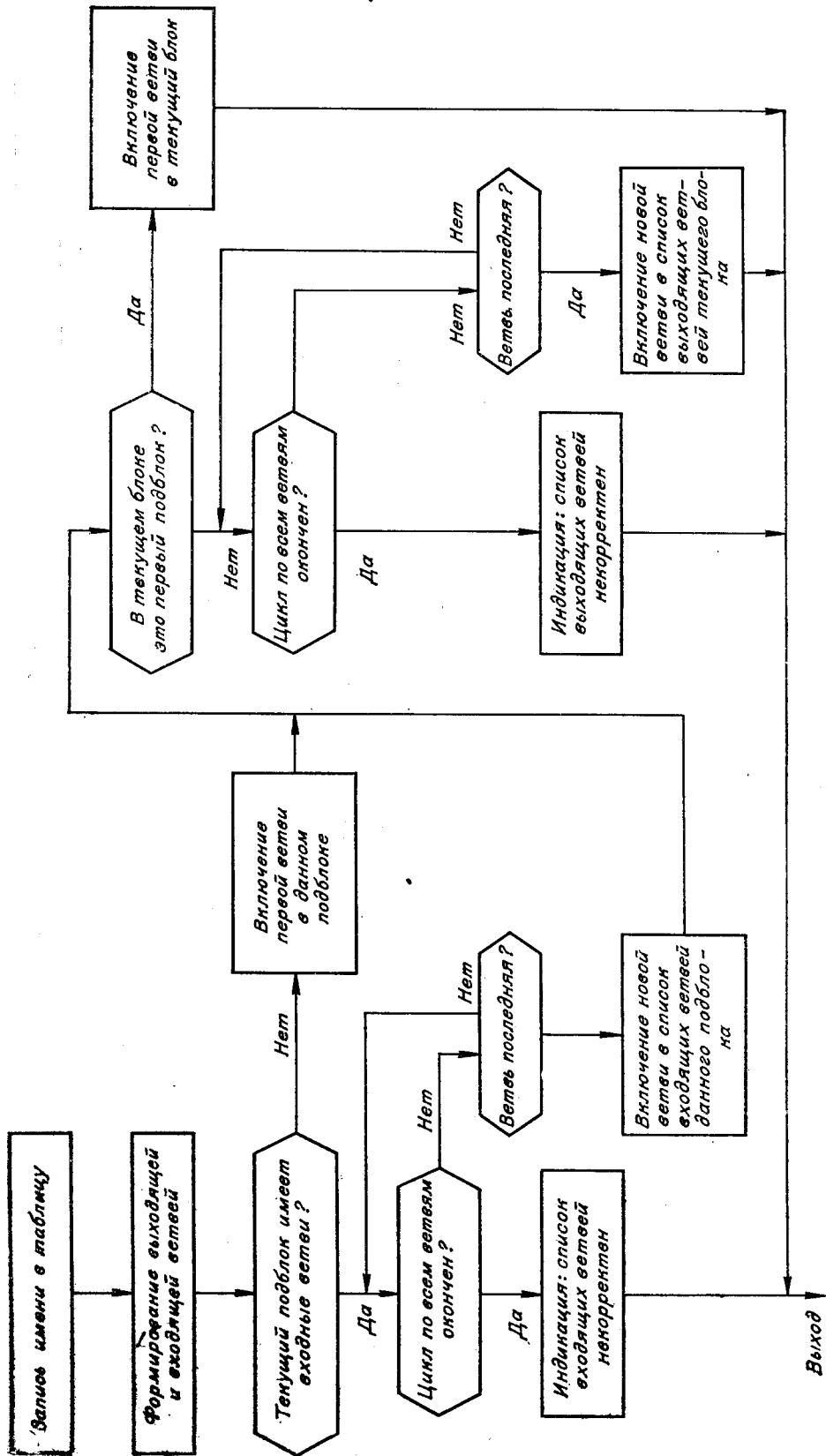


Рис. 4.

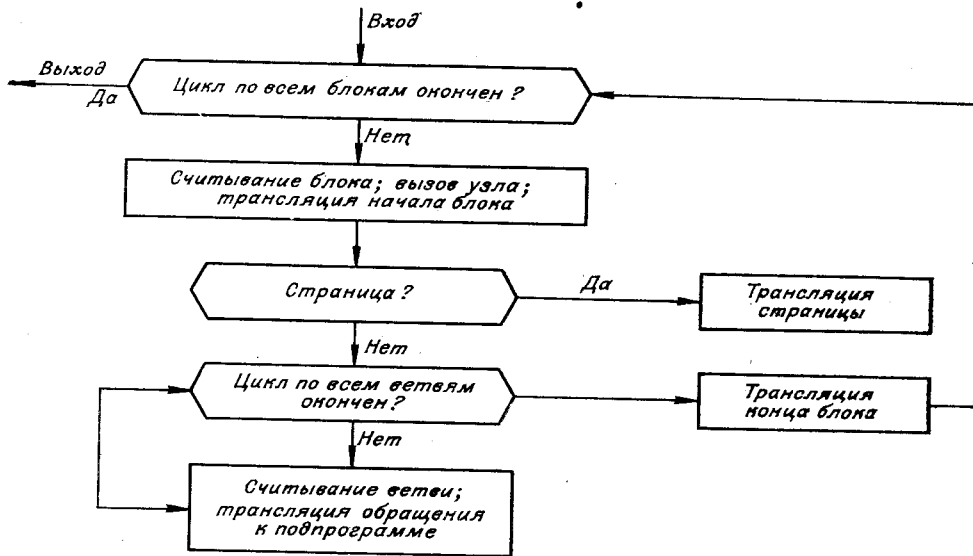


Рис. 5.

кущий блок с привязкой к (X, Y) относительно начала блока. Блок-схема процедуры *ADDBLK* приведена на рис. 4.

Формирование каждого блока начинается процедурой *BLOCK (NAME)* — открыть блок; заканчивается процедурой *ENDBLK* — закончить определение блока.

Трансляция описания изображения в массив индикации.

Из оказанного ранее ясно, что модификация любого фрагмента

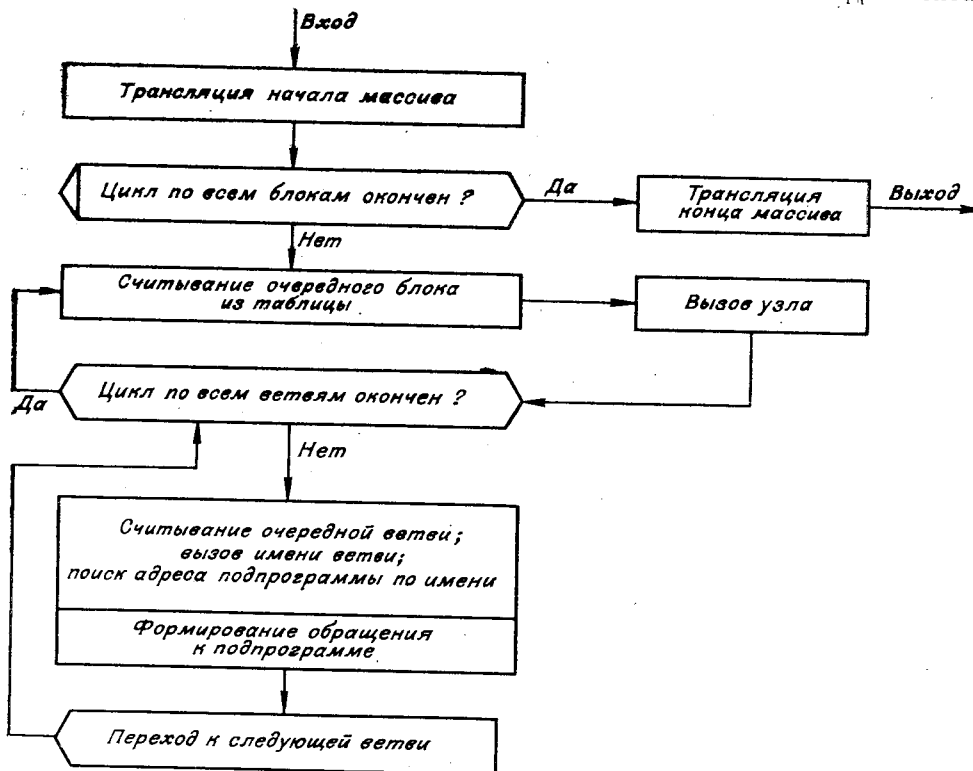


Рис. 6.

изображения сводится к заполнению необходимой информацией (об угле поворота, о смещении и т. п.) модификатора и фиксации ссылки на него в соответствующей выходной ветви. Эта информация либо вводится в текст с исходной программой, либо, чаще всего, поступает через МСО от оператора дисплея. Таким образом, наиболее важной частью МО являются процедуры, транслирующие модифицированное описание изображения в массив индикации, и программа анализа МСО. Последняя на основе анализа МСО осуществляет поиск в структуре необходимых фрагментов, модификацию их, создание новых фрагментов и т. п.

Трансляция изображения происходит в три прохода. На первом проходе (рис. 5) формируется массив индикации с пропущенными ссылками на дисплейные подпрограммы. На втором проходе (рис. 6) пропущенные в массиве индикации ссылки восстанавливаются. На третьем проходе в массив индикации вносятся изменения, связанные с модификацией изображения.

Заключение. Описанное программное обеспечение, за исключением процедур формирования дисплейных кодов и программы обмена, выполнено на ФОРТРАНе и поэтому не зависит от типа ведущей ЭВМ. Являясь универсальным по своему назначению, оно позволяет решать любые задачи, однако наиболее удобным оказывается при автоматизации проектирования. Это связано с тем, что в последнем случае аппарат структурирования изображений весьма эффективен.

Дальнейшие работы должны быть направлены на то, чтобы обеспечить возможность использования архивов изображений, расширить услуги по применению световых кнопок, а также освободить программу пользователя от необходимости дешифровать сообщения оператора.

ЛИТЕРАТУРА

1. А. М. Ковалев, В. Н. Котов, А. А. Лубков, А. С. Токарев. Графический дисплей «Дельта». — Автометрия, 1974, № 4.
2. А. Н. Гинзбург, Ю. И. Родионов. Автономный диспетчер графического терминала «Дельта» — Автометрия, 1974, № 4.
3. J. H. Sexton. An Introduction to Data-Structures With Some Emphasis on Graphics. — Computer Bulletin, September, 1972.
4. А. Н. Гинзбург, Ю. И. Родионов. Структура программного обеспечения системы «Экран». — Автометрия, 1973, № 2.

Поступила в редакцию 8 января 1974 г.

УДК 528.7 : 778.35 : 522.61 : 539.1.073 : 771.534 : 531 : 429 : 681.3.01 : 681 : 39 : 681.327.521 : 522

**Л. В. БУРЫЙ, В. П. КОРОНКЕВИЧ, Ю. Е. НЕСТЕРИХИН,
А. А. НЕСТЕРОВ, Б. М. ПУШНОЙ, С. Е. ТКАЧ, А. М. ЩЕРБАЧЕНКО**
(Новосибирск)

ПРЕЦИЗИОННЫЙ ФОТОГРАММЕТРИЧЕСКИЙ АВТОМАТ

Результаты обработки фотограмметрической информации обычно представляют собой совокупность статистик двумерного поля фотографических плотностей. Фотограмметрические массивы имеют весьма большой объем, так что ввод полного массива в ЭВМ представляется целесообразным лишь в особых случаях. Избыточность фотографических изображений, обусловленная наличием априорной инфор-