

УНИВЕРСАЛЬНАЯ СИСТЕМА ДВУМЕРНОГО ОТОБРАЖЕНИЯ
ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

О.В.Гартман, В.Л.Катков, Г.А.Панкеев, В.И.Фителев

Одним из важных направлений в машинной графике является создание универсальных систем для диалогового решения задач, использующих двумерные изображения [1,2,3]. Примерами могут служить проектирование, расчет и оптимизация электронных схем; создание и корректировка блок-схем программ; расчет сетевых графиков, кинематических схем, схем трубопроводов и т.д.

В программировании вообще и особенно в машинной графике можно выделить три категории людей, работающих с программным обеспечением. Первая категория - системные программисты, которые создают трансляторы с графических языков, пакеты графических подпрограмм и метасистемы и обеспечивают привязку графических устройств к ЭВМ. Вторая категория - это программисты применения, которые, используя средства, созданные системными программистами, проектируют системы, ориентированные на конкретные приложения. Третья категория - это пользователи, работающие в тех областях науки и техники, для которых были созданы системы, и применяющие эти системы для решения своих специализированных задач. Универсальная система двумерного отображения - УСИДО - является графической метасистемой и предназначена для программистов применения, которые, воспользовавшись ею, могут построить конкретные графические системы. Основные графические средства, используемые в большинстве графических систем, уже имеются в УСИДО, недостающие - создаются с ее помощью. Расчетные программы для конкретных задач также можно включать с помощью УСИДО в строящуюся графическую систему.

Благодаря предоставленным графическим и диалоговым средствам, пользователь имеет возможность строить на дисплее нужные ему рисунки.

сущи, задавать физические и другие параметры, производить требуемые расчеты и получать результаты в виде графиков или таблиц, корректировать рисунки и вновь производить расчеты. Законченные проекты пользователь может выводить на регистрирующие устройства (графопостроители, устройства микрофильмирования и т.д.), получая проектные документацию.

В каждой графической системе предусмотрены удобные средства для хранения созданных рисунков и введенных новых графических элементов.

Кроме этого, в УСИДО заложена возможность изменения и расширения ее собственных программных модулей, что позволяет легко модернизировать систему.

§1. Организация данных в УСИДО

При решении задач машинной графики главным объектом, над которым работает пользователь, является изображение, создаваемое им на экране дисплея. Изображение, как правило, состоит из нескольких картин, имеющих некоторое смысловое значение для пользователя. В процессе диалога с ЭВМ он модифицирует картины или их части, создает новые, стирает старые или запрашивает нужную ему информацию относительно этих картин (расчет электрической цепи, определение прочности различных узлов конструкции и т.д.). При изменении и создании новых картин пользователь оперирует частями картин, называемыми подкартинами. Подкартина, в свою очередь, состоит из более мелких подкартин. Таким образом, картина представляет собой иерархическую структуру образов. Поскольку для работы с картиной необходимо иметь ее представление в ЭВМ, то таким представлением является иерархическая структура данных [4-6]. Любое изменение на экране вызывает изменение и в структуре данных, поэтому организация структуры данных и работа с ней являются одним из основных вопросов в системе машинной графики.

В УСИДО используется структура данных, оптимально учитывающая особенности задач, работающих с двумерными объектами, которые состоят, в основном, из повторяющихся графических элементов [7-9].

Графическая структура данных представляет собой древовидно-кольцевую структуру, состоящую из блоков узлов, терминальных элементов, блоков связей, текстов и блоков для положительной неграфической информации.

Все блоки узлов, которые соответствуют картинкам, введенным пользователем, образуют верхний уровень структуры и являются "корнями деревьев", определяющих картины пользователя. Само дерево состоит из блоков узлов, соответствующих подкартинам, и терминальных элементов, определяющих графические стандартные элементы, используемые в картине. Причем, независимо от числа использований одного графического элемента, в структуре заводится один терминальный элемент. Все блоки узлов одного уровня связаны в двусторонний кольцевой список. Каждый блок узла может иметь кольцо связей, кольцо точек и кольцо текстов, которые принадлежат подкартине, соответствующей этому блоку узла. Помимо этого, существует кольцо межкартинных связей, обеспечивающих связь между точками различных подкартин. Пример структуры данных приведен на рис. 1 (БУ - блок узла, БТ2 - блок точки, БС - блок связи, БТС - блок текста, ТЭ - терминальный элемент).

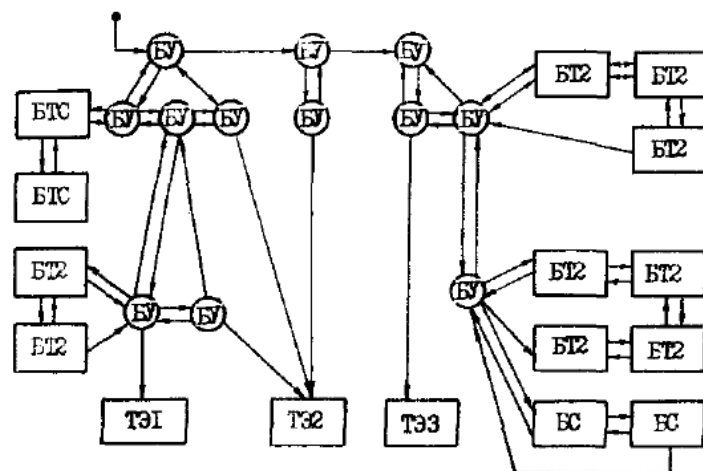


Рис. 1

Блок узла используется для идентификации подкартин и для указания информации о составе подкартин. Помимо указателей на списки точек, связей и текстов, на соседние блоки узла и терминальные элементы, в блоке узла содержится имя подкартин и информация о преобразованиях.

Весь свой рисунок пользователь создает на листе с размерами, заданными им самим. Неподвижная система координат, в которой ва-

числяются все координаты рисунка, жестко связана с листом, начало координат находится в левом нижнем углу, ось x направлена по горизонтали вправо, ось y — по вертикали вверх. За единицу измерения принят 1 мм. Поскольку весь лист может не поместиться на экране дисплея, то задается окно в рабочем поле экрана, на котором изображается рисунок. Пользователь может передвигать окно по листу, наблюдая и разрисовывая различные его части. Сам лист можно увеличивать или уменьшать в процессе работы. Единица измерения при этом остается неизменной.

Каждая подкартина представляется в ЭВМ некоторым поддеревом структуры данных. "Корень" этого поддерева — узел с названием подкартины. Все координаты в блоках поддерева задаются относительно системы координат, определяемой этим корневым узлом. Система координат подкартины определяется следующим образом. Вначале эта подвижная система совмещена с неподвижной системой координат. Затем подвижная система проходит дерево структуры с самого корня по ветке, содержащей данный узел, при этом над ней выполняются все преобразования, указанные во встречающихся блоках узлов: перенос начала координат на $\Delta x, \Delta y$, поворот системы координат на угол $\Delta \theta$ вокруг начала, масштабирование по осям X и Y относительно начала координат. Таким образом, получается изображение картины в неподвижной системе координат. Та часть картины, которая попадает в окно, после клипирования высвечивается на экране.

Подобная организация преобразования систем координат в структуре позволяет с минимальными затратами включать и исключать картины друг из друга, объединять картины и манипулировать ими.

Информация о преобразовании в узле состоит из координат перемещения $\Delta x, \Delta y$; угла поворота $\Delta \theta$, коэффициентов масштабирования по осям K_x и K_y ; координат начала рамки, ограничивающей подкартину, α_x, α_y ; размеров рамки $\Delta x, \Delta y$.

Координаты рамки задаются в той же относительной системе координат, что и вся подкартина. Рамка используется для того, чтобы проверить, попадает данная подкартина в окно или нет. Это позволяет значительно уменьшить время прохождения по структуре при создании дисплейного файла [10].

С блоком узла может быть связана неграфическая информация, содержащая какие-либо топологические или физические данные относительно картины, представленной данным узлом.

Блок точки возникает в структуре в тех случаях, когда в подкартине рисуется точка. С точкой может быть связана неграфическая информация (например, физические параметры и т.д.), на которую в блоке точки имеются ссылки. Кроме этого, блок точки может иметь ссылки на блоки связей, для которых данная точка является граничной, а также на блок текста, для которого данная точка является началом текста. Координаты в блоке точки задаются относительно системы координат узла, к которому принадлежит эта точка.

Блок связи используется для определения в структуре связывающих линий или в тех случаях, когда на некоторые две точки изображения накладываются какие-либо ограничения (например, совпадение координат X или постоянство расстояния и т.д.). Так же, как для узла и точки, для связи может существовать неграфическая информация, задающая физические параметры этой связи. Информация об ограничении, задаваемая в блоке связи, — это числовое имя, указывающее характер связи.

Текст так же, как и точки, включается в подкартину, т.е. в узел. С текстом можно производить простейшие операции редактирования, следовательно, его размеры не фиксированы; поэтому сам текст отделен от блока текста и находится в шаблоне текста. С текстом иногда связывается неграфическая информация, и всегда указывается точка, являющаяся началом текста. Эта точка находится в вершине прямоугольника, задающего область, в которой размещается и редактируется текст.

Терминальный элемент содержит графическое описание стандартного элемента, которое используется при построении картины. Графическое описание представляет собой запись на внутреннем графическом языке, из которой далее получаются дисплейный файл или команды для графопостроителя.

Если стандартный элемент используется в картине несколько раз, то на терминальный элемент будет указывать несколько узлов.

Для сопоставления подкартин, точек, связей, текстов с участками дисплейного файла в буфере дисплея, где они отображаются, вводятся таблицы соответствия, которые позволяют при указании световым пером на графический объект быстро и легко идентифицировать в структуре данных блок, соответствующий этому объекту.

Память для данных пользователя делится на две части, в одной из которых располагаются различные таблицы, со-

бочие ячейки и рабочие буфера, а другая предназначена для организации структуры данных пользователя.

В связи с тем, что памяти ОЗУ, как правило, не будет хватать для структуры рисунка пользователя, в системе организованы подкачка и откачка этой памяти на барабаны, а следовательно, и виртуальная адресация элементов структуры. С этой целью введена таблица приписки, в которой при подкачке осуществляется коррекция адресов, а ссылки в самой структуре не изменяются. Память, используемая для структуры данных, делится на сегменты одинаковой длины, которые и пересылаются на барабан. В каждый момент в памяти находятся свободные и занятые куски. Все занятые куски доступны по ссылкам, начиная с вершины какой-либо картины. Все свободные куски объединяются в список свободной памяти. Так как структура растет динамически, то в УСЦДО предусматривается возможность запроса памяти различной длины для новых блоков и отказа от памяти из-под блоков структуры, ставших уже ненужными. В том случае, когда структура в процессе работы вырастает до таких пределов, что не помещается в оперативной памяти ЭВМ даже после объединения всех разрозненных свободных кусков в один, включается механизм откачки, обеспечивающий перенос части структуры из сегмента, к которому дольше всего не обращались, на барабан.

Перед проектированием УСЦДО были промоделированы различные алгоритмы управления памятью при работе со структурой и выбран оптимальный для класса задач, решаемых УСЦДО [11].

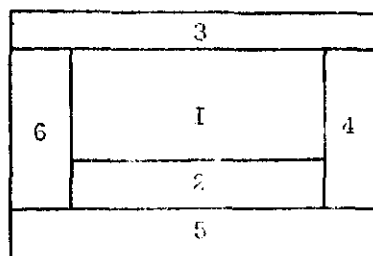


Рис. 2

куска листа, с которым он работает. Область I используется для графического вывода результатов счета структурного изменения, интерак-

Для отображения разноплановой информации при работе с пользователем экран дисплея делится системой на шесть частей (рис.2). В области I пользователь строит картину из различных графических элементов, точек, линий и текстов. Границы области I постоянны, за исключением нижней: она подвижна и изменяется пользователем. Область I является "окном", через него пользователь наблюдает

мер, в виде графиков, гистограмм и т.д. Область 3 используется для сообщений, выдаваемых системой, и для ввода данных с помощью алфавитно-цифровой клавиатуры. Область 4 заполняется световыми кнопками, называемыми "меню", которое используется для указания действий, требуемых пользователем. В области 5 изображаются стандартные графические элементы, из которых пользователь строит свои картины. Так как графических элементов может быть много и все они не поместятся в области 5, то при создании стандартных графических элементов происходит автоматическое разбиение их на листы. Каждый лист содержит столько графических элементов, сколько их помещается в области 5. Часть действий пользователь может задавать нажатием на клавиши функциональной клавиатуры. В области 6 высвечиваются номера функциональных клавиш и имена действий, выполняемых при нажатии на эти клавиши, доступные пользователю в каждый момент.

С помощью УСЦДО формируются подсистемы специального применения. Для каждого отдельного применения существует постоянная библиотека, в которую входят:

- список всех программ УСЦДО и модулей с параметрами;
- список состояний подсистемы;
- список листов со стандартными графическими элементами;
- таблица стандартных графических элементов;
- таблица ограничений для связей, используемых в данной подсистеме.

Библиотеку для каждого применения задает и модифицирует программист применения. Пользователь указывает, с какой задачей он желает работать, и УСЦДО по библиотеке формирует нужную конфигурацию подсистемы для работы с этой задачей. Каталог библиотеки, имеющийся в УСЦДО, позволяет быстро найти участок библиотеки для данного применения, сформировать нужную конфигурацию и перенести ее в быструю внешнюю память (на диски) для дальнейшего использования.

Каждый пользователь подсистем, созданных с помощью УСЦДО, имеет в своем распоряжении два архива: временный и постоянный.

Временный архив располагается на быстрой внешней памяти (дисках, барабанах) или на рабочей ленте. Во временном архиве пользователь может запоминать созданные картины, например, для дальнейшего использования в других местах. При этом запоминается вся структура картины, за исключением таблиц соответствия и ограничений. Поскольку элементы структуры картины разбро-

саны по памяти, то перед запоминанием во временном архиве структура картины плотно упаковывается и все внутренние ссылки перенастраиваются относительно начала этой компактной структуры. Таким образом, во временном архиве картины хранятся в виде компактных блоков памяти. При вызове картины из временного архива внутренние ссылки перенастраиваются по адресу, с которого помещается структура в памяти. Временный архив имеет свое оглавление и сохраняется только в течение одного сеанса работы с системой.

П о с т о я н н ы й а р х и в предназначен для длительного хранения информации пользователя. Здесь могут храниться листы с изображениями, созданные пользователями и записанные ими в архив; временные архивы пользователей, списки листов со стандартными графическими элементами, которые создал для себя пользователь, помимо библиотечных графических элементов, предоставленных ему системой. В отличие от временного архива, листы с изображениями запоминаются в постоянном архиве с полной неупакованной структурой, определяющей все изображения пользователя. Постоянный архив располагается на магнитной ленте, имеет свой каталог. В архиве предусмотрена защита информации от других пользователей.

§2. Структура системы

УСИДО состоит из восьми функциональных блоков: инициализации, монитора, управления структурой данных, построения стандартных графических элементов, рисования, архива, блока программиста применения, связи с графическими устройствами.

Б л о к и н и ц и а л и з а ц и и обеспечивает считывание УСИДО с ленты в оперативную память ЭВМ и на барабаны. На диск считываются первые четыре состояния, необходимые для предварительной работы по идентификации пользователя и указания подсистемы, с которой желает работать пользователь. Блок инициализации находит в библиотеке УСИДО ту подсистему, которая нужна пользователю, и создает в памяти ЭВМ конфигурацию, необходимую для работы с данной системой. По окончании работы с УСИДО блок инициализации выдает на печать всю собранную за время сеанса статистическую информацию о работе системы.

М о н и т о р организует всю работу по обеспечению диалога пользователя с системой, подкачку нерезидентом в оперативную память ЭВМ, а также сбор статистики о всех параметрах системы (ко-

личество обращений к процедурам системы, количество элементов с внешней памятью; объем памяти, занятой под структуру данных).

Б л о к у п р а в л е н и я с т р у к т у р о й д а н н ы х обеспечивает все действия со структурой и памятью, которые требуются пользователю при работе за экраном дисплея. Организация памяти и структуры данных была описана выше. Блок управления структурой данных выделяет куски памяти требуемой длины для заведения новых элементов структуры, включает в список свободной памяти куски памяти, старые ненужные структуре данных, обеспечивает подкачку элементов структуры во внешнюю память и обратно. В этом блоке обеспечивается заведение узлов, точек, связей, текстов и терминальных элементов, включение их в различные списки и поддеревья структуры данных; исключение блоков из списков и поддеревьев структуры. Одной из основных функций блока управления структурой данных является способность структурирования картин, а именно: включение одной картины в другую, исключение подкартинки из картины (в этом случае подкартинка начинает существовать как самостоятельная картина), соединение двух картин в одну, что позволяет в дальнейшем выполнять все манипуляции над этими картинками одновременно. Все эти действия требуют сохранения формы рисунка и его расположения на экране, что приводит к просмотру структуры, в которую входят данные картины, и пересчету информации о преобразовании в узлах, соответствующих этим картинкам.

Б л о к п о с т р о е н и я с т а н д а р т н ы х г р а ф и ч е с к и х э л е м е н т о в обеспечивает пользователя и программиста применения средствами для рисования графических элементов, требуемых для создания различных изображений.

Построение стандартного графического элемента осуществляется в области I экрана дисплея. Описание стандартного графического элемента в памяти ЭВМ состоит из набора таких команд внутреннего графического языка, как:

- изобразить точку с абсолютными координатами X, Y ;
- переместить луч (перо) в точку X, Y ;
- провести линию из текущего положения в точку X, Y ;
- изобразить заданный текст с точки X, Y в прямоугольнике $\Delta X, \Delta Y$;
- изобразить окружность с центром в точке X, Y и радиусом R ;
- изобразить дугу окружности по трем точкам: $X_1, Y_1; X_2, Y_2; X_3, Y_3$;

- изобразить эллипс по центру в точке X, Y и двум полуосям a, b.

При создании стандартного графического элемента строится простейшая табличная структура, в которой хранятся описание этого элемента.

Блок построения стандартных графических элементов обеспечивает следующие действия пользователя:

- задание точек с помощью пересечения вертикальной и горизонтальной линий, которые перемещаются при указании пользователя световым пером на горизонтальный или вертикальный ряд рисок;

- задание точек с помощью координат, введенных пользователем с алфавитно-цифровой клавиатуры, или с помощью маркера, перемещаемого по экрану дисплея джойстиком;

- проведение линии между двумя последними заданными точками, причем изображение этих точек исчезает с экрана дисплея, и они удаляются из графического элемента;

- задание окружности по двум точкам, одна из которых определяет центр окружности, а вторая - ее радиус; после изображения окружности обе точки удаляются из графического элемента;

- задание окружности или дуги окружности по трем точкам (графически на экране дисплея окружности представляют собой правильные двенадцатиугольники, чего вполне хватает для наглядности изображения);

- задание эллипса по центру и полуосям (центр задается точкой, обе полуоси - координатами X, Y другой точки, графически эллипс изображается шестнадцатиугольником);

- задание текста двумя точками прямоугольника, в который вписывается текст, высвечиванием в этом прямоугольнике курсора и вводом текста с алфавитно-цифровой клавиатуры.

Пользователь имеет также возможность запоминать различные текущие положения луча в стеке и возвращать луч в эти положения в любое время подобно тому, как это было сделано в системе ИРИС [12].

В графическом элементе пользователь может задать любое число точек привязки, использующихся в дальнейшем при построении картин для соединения графических элементов связями в заранее определенных точках.

Кроме этого, блок построения графических элементов обеспечивает стирание любых частей и всего создаваемого элемента; при -

связывание графическому элементу имени, по которому он может в дальнейшем идентифицироваться и вызываться на экран дисплея; запись графического элемента в текущий лист для стандартных графических элементов (при этом изображение созданного элемента помещается в области 5 экрана дисплея); вызов заданного по имени графического элемента в область I и редактирование его по требуемой форме; стирание графических элементов, заданных по имени, из листов с этими элементами.

Если при записи графического элемента в лист и высвечивании его в области 5 элемент размерами по вертикали не помещается в этой области, то он автоматически масштабируется относительно своего начала. Если очередной графический элемент не помещается в области 5 по горизонтали (т.е. много места занимают предыдущие элементы), то текущий лист автоматически закрывается и заводится новый лист, в который заносится графический элемент.

Блок рисования является основным блоком УСМО, обеспечивающим действия пользователя по созданию изображения на экране дисплея. Блок рисования в значительной мере связан с блоком управления структурой данных, так как любое изменение изображения немедленно отражается на структуре данных.

Блок рисования обеспечивает следующие действия пользователя:

- Задание длины и ширины листа, на котором будет строиться изображение. В первоначальный момент считается, что лист совпадает с областью I на экране дисплея. Все цифровые и текстовые данные здесь и в других директивах задаются с помощью алфавитно-цифровой клавиатуры.

- Задание длины и ширины области I в координатах листа. При этом физические размеры области на экране не изменяются, за исключением площади куска листа, изображаемого в области I.

- Присваивание листу буквенно-цифрового имени для дальнейшей записи его в архив. Здесь и в дальнейшем длина имени не превышает шести символов.

- Увеличение/уменьшение размеров листа по X и Y в любое число раз (вещественное). Аналогично можно изменить масштаб единицы измерения в листе по осям X и Y. В первоначальный момент за единицу измерения берется 1 мм.

- Перемещение окна, определяемого областью I, по листу в любое место. Кроме этого, задание непрерывного движения окна по листу в любом направлении, горизонтальном или диагональном направ-

лениях и с тановка этого движения. Окно при этом перемещается каждый раз на 10 единиц измерения в листе.

- Вызов в область 5 по номеру или имени элемента любого листа с графическими элементами для построения картины из стандартных графических элементов.

Для построения изображения пользователь определяет начало картины, которая в этот момент становится текущей, и может присвоить ей буквенно-цифровое имя.

Тем же способами, что и в блоке построения стандартных графических элементов, здесь строятся точки в области I. Две точки могут быть соединены линией. В заданной точке области I можно поместить стандартный графический элемент, указанный в области 5 световым пером или заданный по имени, причем в структуре данных для этого графического элемента будет создан терминальный элемент, если его ранее не было в структуре.

Для того чтобы манипулировать с картинами (или подкартинами), пользователь должен идентифицировать эту картину, что он и делает либо указывая световым пером на эту картину, либо задавая ее имя с помощью алфавитно-цифровой клавиатуры. Такие элементы картины, как точки, линии, тексты, пользователь может идентифицировать лишь световым пером.

Поскольку картины у пользователя тесно структурированы, ему предоставляется возможность объявлять некоторые подкартины недоступными (или доступными) для указания световым пером, тем самым поднимается уровень подкартин, идентифицируемых световым пером.

С любой картиной пользователь может осуществлять следующие действия:

- передвинуть в указанную точку;
- передвинуть на ΔX и ΔY ;
- повернуть вокруг начала на угол, заданный в градусах с алфавитно-цифровой клавиатуры;
- увеличить/уменьшить по осям X и Y отдельно относительно начала в любое число раз;
- скопировать ее в любой точке области I, при этом все дерево, соответствующее этой картине, будет скопировано и включено в общую структуру.

Картину также можно погасить на экране, не убирая из структуры; стереть с экрана и убрать из структуры данных; можно заново высветить погашенную картину.

Пользователь может изобразить текст таким же образом, как в блоке построения стандартных графических элементов. Любой ранее созданный текст может быть изменен.

Как уже упоминалось выше, картины могут включаться одна в другую, соединяться в одну, исключаться одна из другой.

Пользователь может соединить линией первую и последнюю точки привязки двух подкартин или любые две точки привязки. Если в системе предусмотрены другие ограничения для точек, то по номеру ограничения он может задать нужную связь между двумя точками. При дальнейших преобразованиях картин связь, заданная между точками, остается, если это возможно.

При высвечивании картины на экране дисплея в блоке рисования происходит создание дисплейного файла по структуре данных [10]. При этом всегда сначала проверяется, попадает ли прямоугольник, ограничивающий подкартину, в область I, и если нет, то эта подкартина далее не рассматривается. Если хотя бы часть картины попадает в область I, то осуществляется преобразование систем координат, производится просмотр поддерева подкартины и создается дисплейный файл для точек, линий, текстов и графического описания терминальных элементов. При создании дисплейного файла изображение клиппируется областью I (см. [13]).

Блок архива обеспечивает все возможные действия пользователя с постоянным и временным архивами системы. Такими действиями являются: запись и считывание из постоянного архива листа с изображениями пользователя; считывание и запись временного архива и листов со стандартными графическими элементами; замена и стирание листов с изображениями, временного архива и архива со стандартными графическими элементами. Кроме этого, пользователь может просмотреть на экране каталог архива, чтобы вспомнить имена нужных ему элементов.

Блок программиста применения является отдельной подсистемой и обычным пользователем не используется. С этим блоком работает программист применения в тех случаях, когда он хочет сделать новую подсистему или изменить уже существующую. Блок программиста применения обеспечивает следующие возможности:

- заведение новой подсистемы, которую хочет создать программист применения;

- вызов для редактирования из библиотеки УСЦДО уже имеющейся системы;

- запись, стирание и замена подсистемы в библиотеке УСЦДО.

При заведении новой подсистемы в нее сначала автоматически включаются состояния и программы УСЦДО, обеспечивающие диалоговую и графическую работу. Далее программист применения уже может добавлять, изменять и выкидывать программы, состояния, листы со стандартными графическими элементами, таблицу ограничений.

При вводе новой программы в подсистему программист применяет указывая ее имя, адрес в ОЗУ ЭВМ, с которого она должна работать, и место, откуда можно взять эту программу (ввод с перфокарт, считывание с магнитной ленты, длина). После этого программа записывается в подсистему либо вместо любой другой программы, либо на новое место.

При создании новых и редактировании старых листов со стандартными графическими элементами используются процедуры блока построения стандартных графических элементов. В системе различаются листы со стандартными графическими элементами, созданные программистом применения и обычным пользователем. Пользователь не может редактировать графические элементы, созданные программистом применения (они защищены от пользователя), в то время как программист имеет доступ к графическим элементам пользователя.

Блок программиста применения обеспечивает возможность ввода в подсистему новых состояний и редактирования старых. Это необходимо при включении в подсистему новых графических возможностей или программ расчета, требуемых при решении задач, для которых создается подсистема. При этом в состоянии можно изменять любой элемент, входящий в его описание. Отредактированное или вновь созданное состояние может быть записано в подсистему как новое или вместо другого состояния.

При работе с программами и состояниями имеются такие программы и состояния, которые "закрыты" для редактирования программистом применения, так как они обеспечивают правильную работу системы, и без них система не может существовать.

Для того чтобы установить новые связи между точками изображений, программист применения должен скорректировать таблицу ограничений и ввести программы, обеспечивающие эти связи.

Для каждого ограничения можно изменить его номер, имена программы, соответствующих данному ограничению, и записать измененное ограничение вместо любого другого в таблице ограничений.

Помимо тех возможностей, которые предоставлены программисту применения для модификации и создания новых подсистем, в УСЦДО предусмотрена возможность модификации самой УСЦДО системным программистом. Для этого существует особое состояние, в которое можно попасть, только имея специальный шифр. В этом состоянии системный программист может изменить любую программу и любое состояние УСЦДО и заменить старую УСЦДО новой. Здесь же он может изменить шифр, по которому разрешен доступ в данное состояние.

Блок связи с графическими устройствами позволяет пользователю выдавать созданные им рисунки на различные регистрирующие устройства. Пользователь может нарисовать лист с изображениями на графопостроителе. Если имя листа задано с алфавитно-цифровой клавиатуры, то этот лист имеется в постоянном архиве, в противном случае берется текущий лист. Любая подкартина, указанная пользователем, может быть выведена на графопостроитель. Содержимое области 2 экрана дисплея также можно нарисовать на графопостроителе.

Все указанные выше объекты могут быть по желанию пользователя отображены на устройстве микрофильмирования.

Необходимо отметить, что при выводе изображений на регистрирующие устройства обеспечивается большая точность рисунка. Предварительно дисплейный файл или структура изображения преобразуются в запись на внутреннем графическом языке, с которого далее идет перекодировка на требуемое устройство. При этом команды рисования окружности, дуги окружности, эллипса создают более точные изображения, чем на дисплее. Если в устройстве есть круговой интерполятор [14], то он используется для рисования окружностей, иначе окружности и эллипсы представляются шестидесятиугольниками.

§3. Организация диалога

При работе с УСЦДО пользователю невозможно предоставить сразу все команды, существующие в системе, так как во-первых, их слишком много, а, во-вторых, действия пользователя в каждый небольшой отрезок времени ограничиваются небольшим набором команд. Поэтому работа с системой ведется по диаграмме состояний и 2 [15,16]. Каждое состояние определяет средства и команды системы, доступные пользователю, приведенные в это состояние. После выполнения команды, заданной пользователем, система переходит в

состояние, которое является состоянием перехода для этой команды по диаграмме состояния. Таким образом, диаграмма состояний представляет собой граф, вершинами которого являются состояния, дугами — переходы из состояния в состояние, причем дугам соответствуют процедуры, выполняемые при задании команд.

Описания состояний для каждой подсистемы хранятся в библиотеке УСИД.

Описание состояния включает в себя:

- номер состояния;
- длину описания;
- набор световых кнопок, для каждой из которых задается имя, имя программы, соответствующее этой кнопке, номер состояния, в которое необходимо перейти после выполнения программы;
- набор функциональных клавиш, для каждой из которых задается имя и номер, имя программы, соответствующее этой клавише, номер состояния, в которое необходимо перейти после выполнения программы;
- набор команд, задаваемых с алфавитно-цифровой клавиатуры, для каждой команды задается ее имя, имя программы и номер состояния перехода;
- сообщение, высвечиваемое на экране, когда система переходит в данное состояние;
- признак разрешения на указание световым пером или прерывание от джойстика, имя программы, обрабатывающей это указание, и номер состояния перехода;
- признак ввода параметров с алфавитно-цифровой клавиатуры, при этом в область 3 экрана дисплея выводится курсор, что позволяет пользователю вводить данные в ЭВМ;
- номер областей экрана дисплея, которые должны быть "почищены" при переходе в данное состояние.

В каждом состоянии может быть описано не более 16 световых кнопок и не более 10 функциональных клавиш, что связано с ограничением в дисплейном буфере места, отведенного для областей 4 и 6 экрана дисплея.

Работу системы можно представить себе следующим образом. После инициализации система переходит в первое состояние, при этом из внешней памяти в ОЗУ считывается описание состояния, по которому на экране дисплея создается изображение све-

товых кнопок, функциональных клавиш и сообщений. Система в этот момент находится в режиме ожидания действий от пользователя. Как только пользователь произвел некоторое действие (указал световым пером на любой объект, послал прерывание от джойстика или нажал на функциональную клавиатуру), монитор, получив прерывание от дисплея, анализирует, от какого устройства ввода пришло прерывание. Если было прерывание от светового пера, то монитор определяет, на какой объект указал пользователь: если на изображение в области 1, то адрес элемента в структуре данных заносится в специальный стек операций; если на риску в вертикальном или горизонтальном наборах, то отрисовывается линия, исходящая из этой риски, запоминаются новые координаты точки пересечения горизонтальной и вертикальной линий. После этого монитор переводит систему в состояние перехода, соответствующее процедуре обработки прерывания от светового пера, которое указано в описании данного состояния. Если пришло прерывание от джойстика, то монитор считывает и запоминает координаты маркера, находящегося на экране дисплея, и оставляет систему в текущем состоянии. Если пользователь указал световым пером на световую кнопку или нажал на функциональную клавишу, то монитор по описанию состояния определяет имя программы, соответствующей этой световой кнопке или функциональной клавише, и запускает эту программу. На время счета программы в правом верхнем углу экрана дисплея высвечивается специальный символ, показывающий пользователю, что система команду поняла и выполняет ее. После окончания работы программы монитор переводит систему в состояние, указанное как состояние перехода для данной световой кнопки или функциональной клавиши. После этого система переводится в режим ожидания действия от пользователя. Цикл работы системы показан на рис. 3.

Все директивы языка диалога имеют следующий синтаксис: < список параметров > < имя директивы >. Параметрами являются числовые и символические значения,

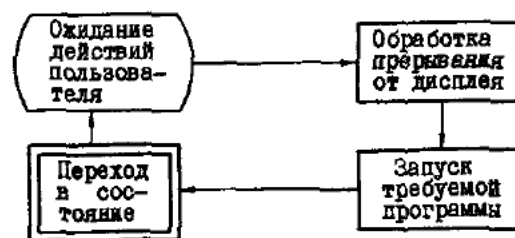


Рис. 3

которые задаются с помощью алфавитно-цифровой клавиатуры и вводятся в специальный буфер приема сообщений. При указании на какой-нибудь элемент изображения адрес блока этого элемента в структуре данных заносится в специальный стек операций и тоже является параметром для директивы. Выполнение директивы начинается с выбора параметров из буфера приема сообщений, из стека операций и из специальных рабочих ячеек (например, для получения координат точки от маркера или пересечения горизонтальной и вертикальной линий). При этом в стеке операций, начиная с его верхушки, ищем элемент такого типа, который подходит директиве в качестве параметра. Далее, указанный элемент изображения на экране дисплея начинает мерцать. Если пользователь нажатием на специальную функциональную клавишу подтверждает правильность выбора элемента изображения, то директива выполняется, в противном случае директива отменяется. При выполнении директивы всегда проверяется правильность задания параметров, в случае ошибки сообщение о неверных параметрах выводится на экран дисплея, и выполнение директивы прекращается. Если при выполнении директив встречаются критические ситуации, такие как переполнение некоторых таблиц, внешней памяти или равенство нулю определителя матрицы преобразования при манипуляциях с картинками, система прекращает работу, сообщая об этом пользователю и выдавая на печать всю статистическую информацию о работе системы.

УСИДО содержит в настоящее время 28 состояний, и продолжается пополнение ее новыми состояниями, в основном содержащими инструкции по работе с системой. Предполагается, что на каждом "рабочем" состоянии в системе будет находиться 1-2 состояния, в которых содержится инструкция о параметрах и функциях действий, указанных в "рабочем" состоянии.

Определение набора директив и их функций в каждом состоянии определяет степень удобства пользования системой [17] и не является однозначным. Только опыт работы с системой покажет оптимальное распределение директив по состояниям и определит степень удобства задания параметров для этих директив.

§4. Реализация УСИДО

При реализации УСИДО использовалась следующая аппаратура: ЭВМ БЭСМ-6, графический дисплей ЕС-7064, мультиплексор ввода-вывода (МП), обеспечивающий интерфейс между БЭСМ-6 и графическими устрой-

ствами ЕС ЭВМ, графопостроители ЕС-7051 и ЕС-7052 [14]. Конфигурация комплекса показана на рис.4.

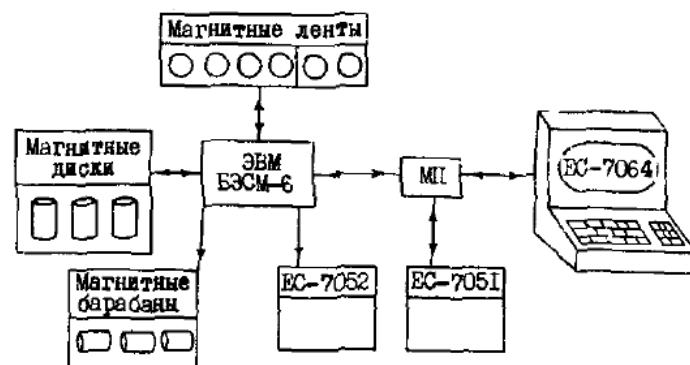


Рис. 4

Привязка графических устройств через мультиплексор к ЭВМ БЭСМ-6 осуществлялась в рамках операционной системы, созданной в Институте прикладной математики АН СССР [18]. При этом были обеспечены диалог с ОС ИМ на дисплее, а также считывание и вывод изображения на дисплей, обработка прерываний от дисплея и вывод изображений на графопостроители.

УСИДО написана на ЯРМО — языке высокого уровня для БЭСМ-6 [19]. Объем системы составил 52К слов БЭСМ-6 без учета памяти под данные, таблицы, рабочие ячейки и т.д.

Объем системы и особенности языка ЯРМО отразились на организации системы. Из более чем четырехсот процедур УСИДО были выделены наиболее часто работающие процедуры и объединены в резидент, т.е. часть системы, постоянно находящуюся в оперативной памяти БЭСМ-6. Остальные процедуры были объединены в нерезидентные сегменты, находящиеся на барабанах. Поскольку в ЯРМО не было возможности сегментирования программы, в монитор были включены специальные процедуры, обеспечивающие подкачку нерезидентов в память БЭСМ-6 при вызове процедур (в том числе и при рекурсивном вызове). Так как в ЯРМО нет действий с вещественными числами, то их пришлось ввести с помощью процедур, что вызвало значительное увеличение времени работы процедур, связанных с пересчетом матриц преобразования.

Время реакции системы - от 1-2 секунд до минуты, в зависимости от выполняемых действий и сложности созданного изображения. Около 80% всех ответов системы происходят менее чем за 10 секунд.

Описанный вариант УСИДО является экспериментальным. Перед тем, как создать систему, была разработана ее модель. Результаты моделирования учитывались при проектировании системы, однако было неясно, правильно и полностью ли в модели учтены все особенности систем машинной графики. Поэтому в УСИДО заложен сбор статистики, позволяющей оценить эффективность работы системы. Одним из главных вопросов является удобство пользования системой. Эксплуатация экспериментального варианта УСИДО позволит сделать выводы относительно достоверности моделирования систем машинной графики и модифицировать УСИДО для удовлетворения требований и пожеланий пользователей.

Хотя настоящий вариант и является экспериментальным, с помощью УСИДО создаются рабочие подсистемы для конструирования блок-схем программ и изображений типа "мозаики".

Пример создания блок-схемы, изображенной на рис.3, приводится в приложении.

Л и т е р а т у р а

1. BELADY L.A., BLASGEN M.W., EVANGELISTI C.J., JENNISON R.D. A computer graphics system for block diagram problems. - "IBM system journal", 1971, v.10, N 2, p.43-58.

2. MORSE S.P. Interactive graphic system for modeling symbolic networks. - "Computer design", 1973, v.12, N 2, p.32-44.

3. WISEMAN N.E., CHENEY C.J., ETHERTON M., HILEZ J.O., LEMKE H.U. RAINBOW - multi-purpose CAD system. - "Software practice and experience", 1972, v.2, N 3, p.65-88.

4. Van DAM A. Data and storage structures for interactive graphics. - "SIGPLAN Notices", 1971, v.6, N 2, p.14-36.

5. WILLIAMS R. A survey of data structures for computing graphics systems. - "Computing surveys", 1971, v.3, N 1, p.3-58.

6. SUTHERLAND I.E. SKETCHPAD: a man-machine graphical communication system. - "AFIPS conference proceedings", SJCC, 1963, v.23, p.329-362.

7. GRAY J.C. Compound data structures for computer - aided design: a survey. - In: Proceedings ACM 20th National conference. S.I., Thompson books, 1967, p.355-376.

8. ABRAMS M.D. Data structures for computer graphics. - "SIGPLAN Notices", 1971, v.6, N 2, p.37-62.

9. ETHERTON M. Data structures for a network design system. - "Computer journal", 1971, v.14, N 4, p.36-53.

10. WISEMAN N.E. A note on compiling display file from a data structure. - "Computer journal", 1968, v.11, N 2, p.141-153.

11. КАТКОВ В.Л., ФИШЕЛЕВ В.И. Выбор алгоритма управления памятью с помощью моделирования. - "Программирование", 1975, № 5, с. 56-63.

12. БРЫНДИН Е.Г., КАБАННИК Л.Г., КАТКОВ В.Л., КОЖУХИН Г.И., ФИШЕЛЕВ В.И., ФИШЕЛЕВА М.И., ШИШОВА Н.А., ЧЕРЕВИКИН А.Ю. Система отображения графической информации ИРИС. - В кн.: Системное программирование. (Материалы Всесоюзного симпозиума, март 1973 г.) Ч.I. Новосибирск, ВЦ СО АН СССР, 1973, с.35-63.

13. SUTHERLAND I.E., SPROULL R.F. A clipping divider. - "AFIPS conference proceedings", FJCC, 1968, v.33, p.765-782.

14. БАСОВ Е.П., АБРАМОВ В.Н. Графические регистрирующие устройства ЕС ЭВМ. М., "Статистика", 1977.

15. NEWMAN W.M. A system for interactive graphical programming. - "AFIPS conference proceedings", SJCC, 1968, v.34, p.287-298.

16. BOULIER P., GRAS J., JANCENS P., LEMAIRE A. META VISU - general purpose graphic system. - In: Graphic languages. Proceedings of the IFIP working conference on graphic languages. Amsterdam, North Holland, 1972.

17. EVANGELISTI C.J., MORSE S.P. Graphical modelling using contextually implied function. - "Computer journal", 1971, v.14, N 4, p.21-35.

18. КРИКОВ В.А., МАКЛАНИН О.А., МАЛАХОВ-КАМАРТАН К.К. Инст - рукция по использованию операционной системы ОС ИИМ. М., ИИМ АН СССР, 1975.

19. ГОЛОДОВОВ В.И., ЧЕБЛАКОВ Б.Г., ЧИНИН Г.Д. Машинно-ориентированный язык высокого уровня для ЭВМ БЭСМ-6. - В кн.: Развитие программного обеспечения БЭСМ-6. М., ВЦ АН СССР, 1975, с.50-51.

Поступила в ред.-изд.отд.
7 июля 1977 года

Приложение

ПРИМЕР ПОСТРОЕНИЯ БЛОК-СХЕМЫ

После вызова УСИДО и инициализации системы построения блок-схем пользователь получает возможность начать построение изображения. При этом в области 5 экрана дисплея высвечены стандартные графические элементы для построения блок-схем (схема 1).

После задания координат точки с алфавитно-цифровой клавиатуры и указания на световую кнопку ТЧКРД пользователь получает изображение точки. Затем, указав световым пером на графический элемент и на световую кнопку ВЫБЭЛ, пользователь получит графический элемент в указанной точке (схема 2).

Таким же образом или задавая точки с помощью маркера, пользователь заносит и другие графические элементы в область 1. Далее, задавая точки на границах этих элементов, он соединяет их линиями, указывая на световую кнопку СОЕТЧ. Кроме этого, в нужные точки заносится элемент, указывающий направление стрелки (схема 3).

В следующем состоянии пользователь указывает световым пером на подкартинки, изображающие стрелки, задает с алфавитно-цифровой клавиатуры угол поворота в градусах, указывает на световую кнопку ПОВЕРН и получает требуемое изображение (схема 4).

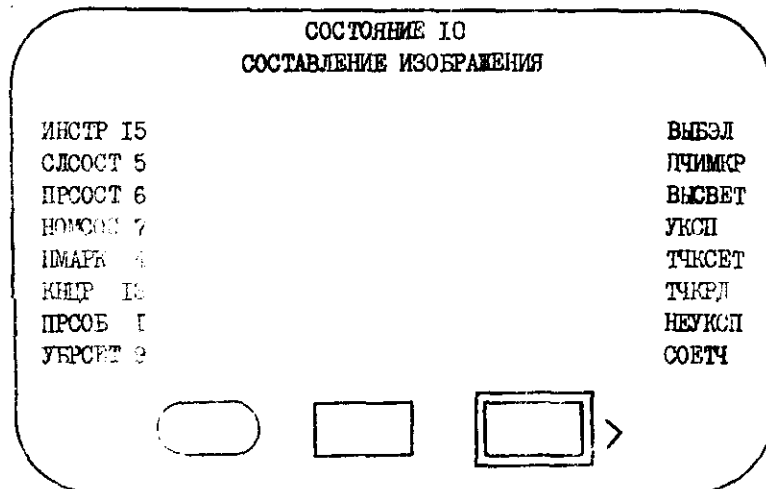


Схема 1

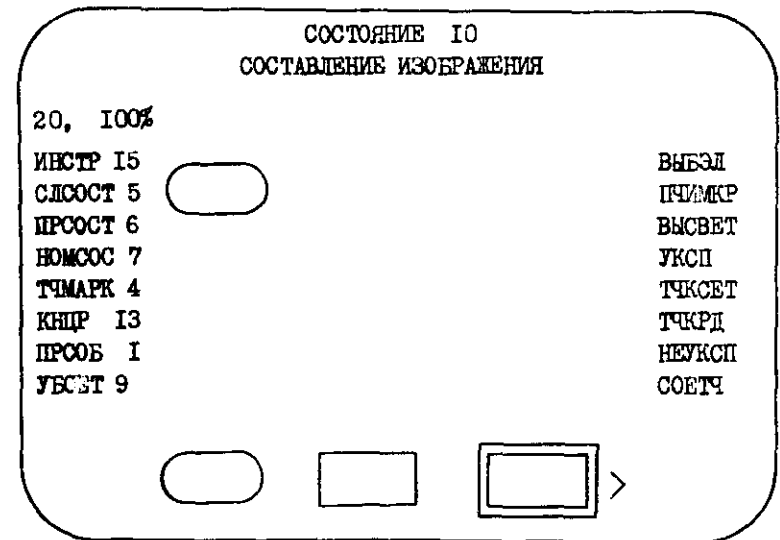


Схема 2

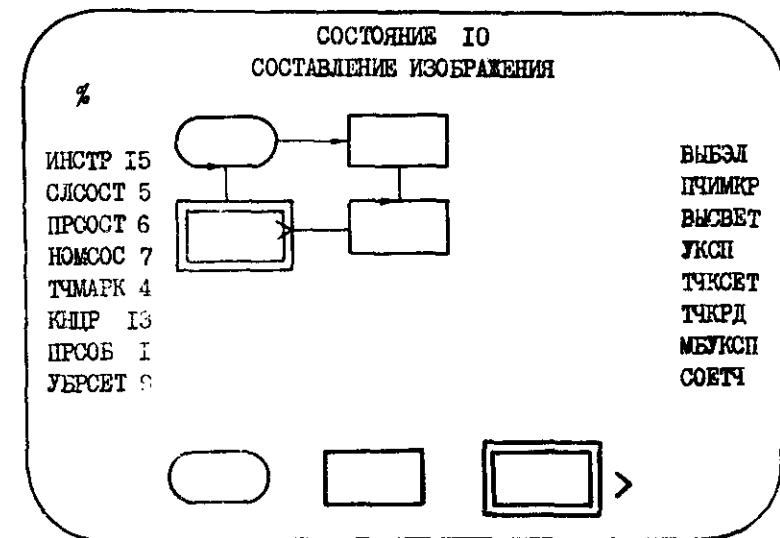


Схема 3

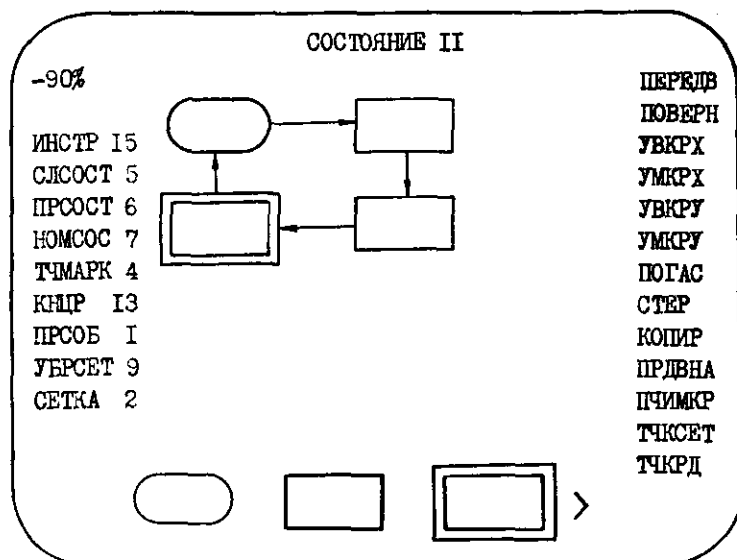


Схема 4

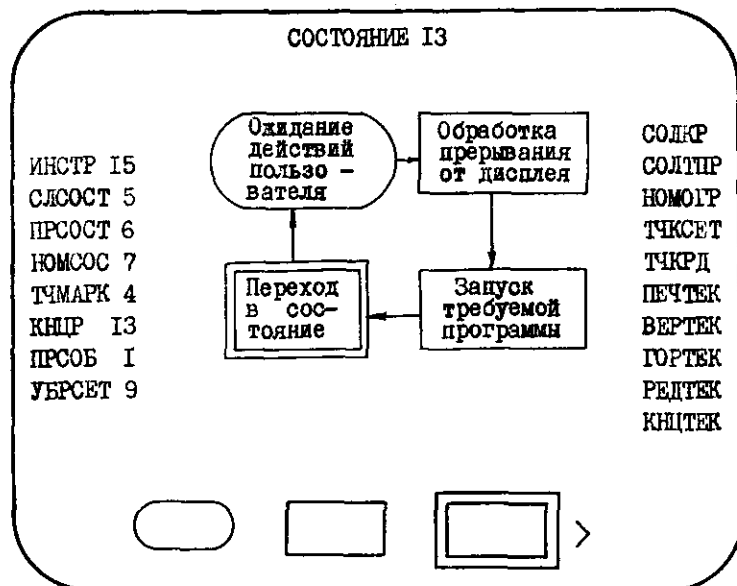


Схема 5

Перейдя далее в состояние I3, пользователь задает верхнюю левую точку, вертикальные и горизонтальные размеры текста, вводит текст с алфавитно-цифровой клавиатуры и получает окончательное изображение (схема 5).

Нажатием на функциональную клавишу I3 пользователь заканчивает работу с УСИДО.