

МАШИННАЯ ГРАФИКА
И ЕЁ ПРИМЕНЕНИЕ

СИБИРСКОЕ ОТДЕЛЕНИЕ АН СССР
ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР

МАШИННАЯ ГРАФИКА И ЕЕ ПРИМЕНЕНИЕ

Сборник научных трудов

Под редакцией Ю.А.Кузнецова



Новосибирск
1974

© ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР СО АН СССР

С О Д Е Р Ж А Н И Е

Стр.

ПРЕДИСЛОВИЕ.....	3
С.В.Горин, В.И.Дворжец, В.А.Дебелов, А.Я.Куртуков.	
Структура СМОГ БЭСМ-6.....	7
В.А.Дебелов. Буферная часть СМОГ.....	19
С.В.Горин. Базовая часть СМОГ.....	38
А.М.Володюков, В.И.Дворжец, А.В.Лукинцов. Генерация символов и чисел в СП СМОГ.....	47
Ю.Г.Земсков, В.П.Карачевский. Программное обеспечение графопостроителя для выпуска чертежей.....	66
В.И.Дворжец, Б.Е.Звенигородский. Алгоритмы вычерчива- ния изолиний.....	76
С.В.Горин, В.И.Дворжец. СМО Дисплея для М-222.....	88
Л.В.Васильева. Организация графического диалогового режима в ОСД-68, "Диспак" ЭВМ БЭСМ-6.....	96
Л.В.Васильева, В.М.Израйлев, П.А.Ким. Практические применения графического диалога в ОС ЭВМ БЭСМ-6.....	104
В.А.Василенко. Приближение сплайнами в евклидовом пространстве.....	114

ПРЕДИСЛОВИЕ

В последние годы одной из характерных тенденций развития средств вычислительной техники является все возрастающий интерес к устройствам графического отображения информации и графического взаимодействия с ЭВМ. Отечественная промышленность освоила серийное производство графопостроителей, а также алфавитно-цифровых и графических дисплеев различных моделей. Осваивается выпуск новых устройств с более широкими возможностями. Внедрение графических устройств в повседневную деятельность Вычислительных центров все актуальнее ставит вопрос о разработке систем их математического обеспечения. Изложению некоторых результатов в этой области, полученных сотрудниками Вычислительного центра СО АН СССР, и посвящен настоящий сборник.

Ю.А.Кузнецов

С.В.Горин, В.И.Дворжец, В.А.Дебелов, А.Я.Куртуков

СТРУКТУРА СМОГ БЭСМ-6

В сборнике [2] в общих чертах уже описывалась СМОГ для БЭСМ-6 в тот момент, когда она находилась в стадии доводки. В настоящей статье мы опишем сложившуюся структуру СМОГ, систему ее комплектования и внесения изменений, а также функционирование СМОГ.

I. Части системы и их взаимодействие

На рисунке I показана общая схема СМОГ БЭСМ-6 как она выглядит в настоящее время. Основной частью СМОГ является стандартная программа "СП СМОГ", которой присвоен индекс БI.(Такая же СП с индексом МI используется и в СМОГ М-220). Поскольку СП БI – программа в кодах БЭСМ-6, обращаться к ней можно из любого автокода и из любой системы программирования, разрешающей работу на автокоде или в машинных командах.

СП БI содержит все программы первого уровня СМОГ [I] плюс сервисные подпрограммы, обеспечивающие контроль входной информации и выдачу информации об ошибках.

На уровне машинных команд обращение к СП БI не представляет трудности. На уровне автокодов – то же самое.

Для языков более высокого уровня создаются специальные комплексы фиктивных процедур I уровня (SWOPPING'u). Эти

процедуры осуществляют пересылку параметров в настоящие подпрограммы первого уровня, тела которых находятся в СП Б1. Кроме того, некоторые из них выдают из СП Б1 необходимую информацию в программы пользователя.

Программы II уровня реализованы для каждой из систем АЛГОЛ, ФОРТРАН и АЛЬФА-6 в виде комплектов реальных процедур. Для работы в системе АЛГОЛ - ГДР используются ФОРТРАН-программы II уровня, так как в этой системе можно обращаться к ФОРТРАН-подпрограммам.

2. Программы II уровня

В настоящее время программы II уровня состоят из 19 подпрограмм, разбитых на 5 комплектов.

2.1. Комплект "ГРАФИКИ"

процедуры: 2.1.1. СИКОР
2.1.2. ГФ
2.1.3. ГФЛИН
2.1.4. ГФКИН
2.1.5. ГФП

описаны в работе [2].

2.2. Комплект "ВЕКТОРНЫЕ ПОЛЯ"

процедуры: 2.2.1. ПОЛЕ
2.2.2. ПОЛИН
2.2.3. КУПОЛ

описаны в работе [2].

2.3. Комплект "ПОВЕРХНОСТИ"

процедуры: 2.3.1. ПРИЗ
2.3.2. РИФТ
2.3.3. ОСПРИЗ
2.3.4. ПСПРИЗ

2.3.1 и 2.3.2 описаны в работе [2]

2.3.3 и 2.3.4 модернизация 2.3.2.

2.4. Комплект "ИЗОЛИНИИ"

процедуры: 2.4.1. ЛИНУР
2.4.2. ЛИНТ
2.4.3. КИНТ
2.4.4. ЛИНУРМ
2.4.5. КИНТМ

2.4.1, 2.4.2, 2.4.3 описаны в работе [2]
2.4.4 и 2.4.5 описаны в работе [13].

2.5. Комплект "ЛИНИИ ТОКА"
процедуры: 2.5.1. ТОК
2.5.2. ТОКИН
описаны в работе [2].

3. SWOPPING'и

Сейчас первый уровень включает 17 процедур.

- 3.1. КЛЮЧ
- 3.2. КАНАЛ
- 3.3. ИНФОР
- 3.4. ЛИСТ
- 3.5. ТРА
- 3.6. ЛИК
- 3.7. ШАГ
- 3.8. ПЕРО
- 3.9. СИМ
- 3.10. ЧИСЛО
- 3.11. ТЕКСТ
- 3.12. ДЕКАР
- 3.13. ТРАД
- 3.14. ДЕК
- 3.15. ДЕЛ
- 3.16. ЛИД
- 3.17. ХРАМ.

Тела этих процедур находятся в СП Б1. Процедуры SWOPPING'а занимаются только пересылкой информации в СП СМОГ и обратно.

Процедуры 3.1 и 3.4 - 3.16 описаны в работе [1]. Процедуры 3.2 и 3.3 описаны в работе [2] и настоящей статье.

Опишем работу SWOPPING'ов на примере SWOPPING'a АЛГОЛ. Процедуры 3.1 - 3.16 не обращаются непосредственно к СП Б1. Связь SWOPPING - СП осуществляется через служебную процедуру ХРАМ. Эта процедура по обращениям от остальных процедур SWOPPING'a формирует обращения к соответствующим им подпрограммам СП Б1.

Обращение к процедуре ХРАМ имеет следующий вид:

ХРАМ (ПОЧТА, ШТАМП);

Параметр ПОЧТА используется для пересылки информации.
Параметр ШТАМП задает режим работы (приказ) для процедуры.

А). $-100 \leq \text{ШТАМП} \leq 100$

В этом случае необходимо обратиться к подпрограмме ХРАМ в СП БИ (такой режим используется только при отладке СП и SWAPPING'a самими разработчиками).

Б). ШТАМП = 101

Сформировать параметр-значение в обращении к СП БИ.

В). ШТАМП = 102

Сформировать параметр-адрес в обращении к СП БИ.

Г). ШТАМП = 103

Сформировать параметр-номер подпрограммы в обращении к СП БИ и обратиться к СП.

Приведем пример работы с ХРАМом процедур ТРА и ЛИД.

Пример 1.

```
PROCEDURE ТРА (x,y,J);  
VALUE x,y,J; REAL x,y; INTEGER J;  
BEGIN REAL Н П ;  
ХРАМ (x,101);ХРАМ(y,101) ХРАМ(J,101);  
Н П :=5; ХРАМ (Н П ,103) END;
```

Сначала в обращении к СП формируются параметры-значения x, y, J, затем формируется номер подпрограммы (для ТРА он равен 5) и происходит обращение к СП.

Пример 2.

```
PROCEDURE ЛИД (x,y);  
REAL x, y;  
BEGIN REAL x1, y1, Ax, Ay, Н П ;  
x1 := x; y1:= y;  
DOWN K: ИА x1,15 ПА, ВИ 15, зч Ax, ИА y1, 15 ПА ,  
ВИ 15, зч Ay, UP;  
ХРАМ (Ax,102); ХРАМ (Ay,102);  
Н П := 16; ХРАМ (Н П , 103);  
x := x1; y := y1; END ;
```

В ячейках x_1 и y_1 запоминаются значения параметров x и y . Затем в ячейках Ax и Ay с помощью АВТОКОД-БЛОКА выбираются адреса переменных x_1 и y_1 . Далее, эти адреса передаются в процедуру ХРАМ для формирования параметров-адресов в обращении к СП. Потом с номером подпрограммы = 16 происходит обращение к СП. После этого в ячейках x_1 и y_1 будут находиться нужные значения и останется только переслать их обратно в x и y . (Напомним, что процедура ЛИД переводит листовые координаты x и y в декартовые, засыпая результат снова в x и y).

4. СП СМОГ Б1

Общее описание. СП Б1 выполняет функции первого уровня для всех систем программирования на БЭСМ-6. СП содержит тела процедур 3.1 - 3.17 и внутреннюю подпрограмму БУФЕР. СП состоит из шести модулей, каждый из которых содержит одну или несколько подпрограмм первого уровня СМОГ. Связь модулей между собой осуществляется при помощи общих переменных и обращений друг к другу. Модули написаны на автокоде БЕМШ и связываются при помощи редактора внешних связей автокода.

Список модулей:

- 4.1. ХРАМ
- 4.2. ЛИСТ
- 4.3. ТРА
- 4.4. ПЕРО
- 4.5. СИМ
- 4.6. БУФЕР.

Модуль ХРАМ выполняет сервисные функции и содержит тела подпрограмм ХРАМ, КЛЮЧ, ИНФОР, ЛИК, ДЕКАР, ДЕК, ДЕЛ и ЛИД.

Модуль ЛИСТ содержит тела подпрограмм КАНАЛ и ЛИСТ, то есть он выполняет настройку СП на канал, выделение ресурсов для рисования и стандартное оформление рисунка.

Модуль ТРА содержит тела подпрограмм ТРА и ТРАД. Он выполняет контроль выхода за границы листа, пересчет координат и производит обращения к БУФЕРу (см. работу [5]).

Модуль ПЕРО состоит из подпрограмм ШАГ и ПЕРО. Выполняет настройку СП на состояние пера прибора.

В модуль СИМ входят подпрограммы СИМ, ЧИСЛО и ТЕКСТ. Они обеспечивают генерацию символов, чисел и текстов (см. [6]).

Модуль БУФЕР содержит тело подпрограммы БУФЕР. Он реализует выдачу информации на графические устройства (см. [4]).

Связи подпрограмм. На рисунке 2 показана схема обращений подпрограмм СП друг к другу. Подпрограммы разбиты на несколько слоев (по признаку "к кому обращаются"). Подпрограммы ИНФОР, ЛИК, ДЕК и ХРАМ не обращаются к другим подпрограммам СП.

При обращении подпрограммы А к подпрограмме Б или к Б извне СП информация для Б передается двумя способами. Во-первых, А или пользователь засыпает в параметры Б необходимые значения или адреса. Во-вторых, Б может использовать любую ячейку некоторого массива ССК (состояние комплекса).

Этот массив содержит полную информацию о состоянии СП. В ССК хранятся параметры текущей декартовой системы координат, состояние прибора, номер канала, размеры листа и так далее.

Приведем список элементов ССК.

0. ССК - характеризует состояние СП в целом

$$ССК = \begin{cases} 1 & СП открыта для обращений \\ 0 & СП закрыта \end{cases}$$

I. ШИФР - шифр пользователя

2. ФХ } - размеры текущего листа в мм
3. ФУ }

4. ФУЛПР - режим работы на текущем листе

$$ФУЛПР = \begin{cases} 0 & \text{жесткая защита листа} \\ 1 & \text{мягкая защита листа} \end{cases}$$

5. ХЛИСТ } - текущие математические листовые координаты
6. УЛИСТ }

7. ХДЕК } - текущие математические декартовы координаты
8. УДЕК }

9. МХ } - текущие масштабы по осям Х и У
10. МУ }

II. ФОРМТ - формат текущего листа

12. КРИАГ - номер шага

13. НПЕРА - номер текущего пера

14. СОСП - состояние пера

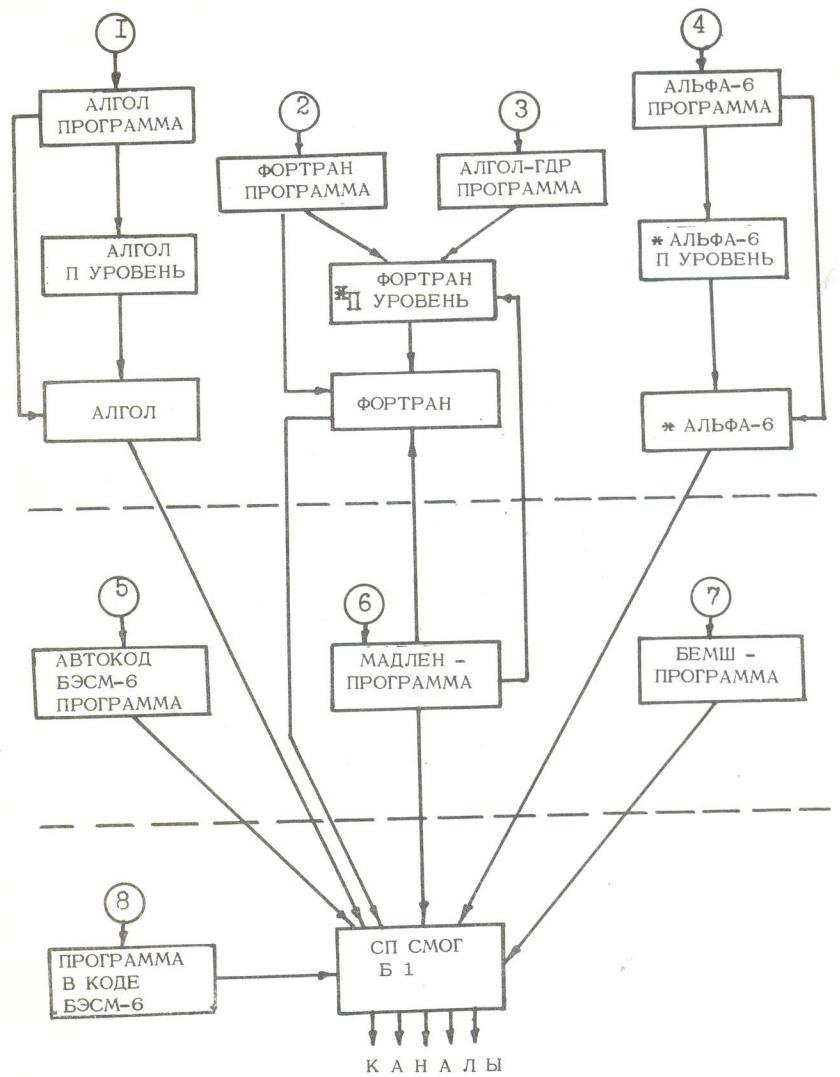


Рис. I

*.) Программы отлаживаются.

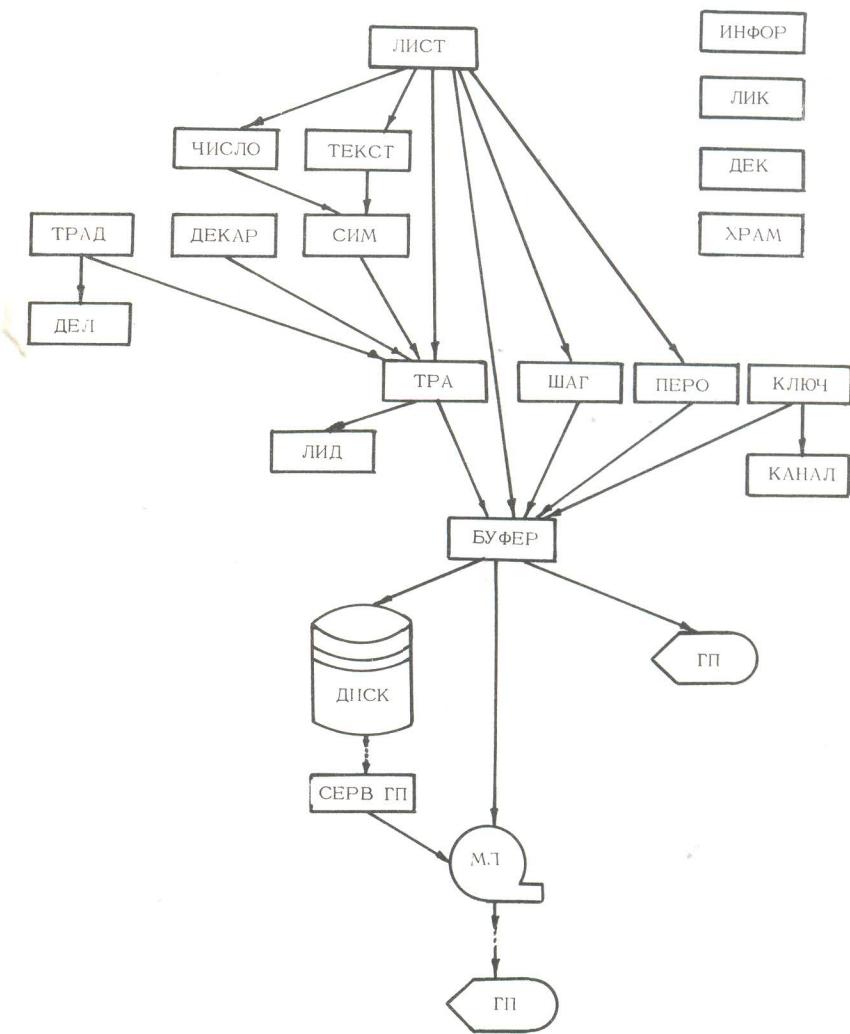


Рис.2

$$СОСП = \begin{cases} 0 - \text{перо поднято} \\ 1 - \text{перо опущено} \end{cases}$$

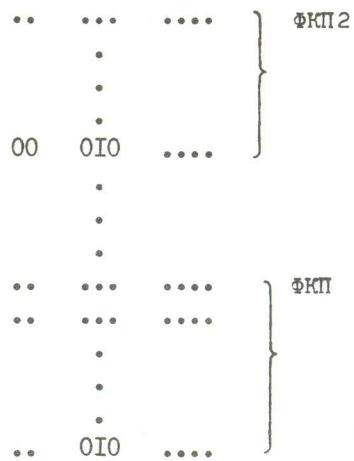
15. ДЛШАГ - длина шага
 16. ЧИСП - число перьев прибора
 17. MAXX } - размеры поля прибора
 18. MAXY }
 19. MAXKP - количество номеров шага у прибора
 20. ПРИБ - номер канала
 21. НЛИСТ - номер текущего листа
 22. ДЛИНА - длина траектории пера (в метрах)
 23. ФХЛ } - текущие физические листовые координаты пера
 24. ФУЛ }
 25. ВЛИСТ - указывает область, где находится перо (математические)

$$ВЛИСТ = \begin{cases} 0 - \text{перо "вне" листа} \\ \neq 0 - \text{перо "внутри" листа} \end{cases}$$

26. ЗАНБХ } - занятый бумажный фонд по осям Х и У
 27. ЗАНЕУ }
 28. MAXДЛ - максимальная длина траектории пера (в метрах)
 29. НННУП - номер подпрограммы СП, которая работает в данный момент
 30. ЗАНПБ - ширина текущей полосы листов
 31. РПП - шкала разрешения печати параметров.

Обращение к СП Б.1. Обращение к СП есть обращение к одной из подпрограмм З.1 - З.17. Оно имеет следующий общий вид:

15	24	Анач.СП
00	000	0000
00	066	0000
10	000	Номер СП
00	000	Номер подпр.СП
..
..
		ФКП 1
		{
..	010



После команды обращения к СП в левой половине ячейки указывается номер подпрограммы СП (от I до I7).

Следующие команды образуют список фактических параметров (ФКП). Каждый элемент этого списка состоит из произвольной последовательности команд 022 и 023, заканчивающейся командой 010. Все команды списка ФКП могут использовать любые индекс-регистры и адреса. Сформированный адрес команды 010 вычисляется СП и считается адресом очередного фактического параметра.

Ниже мы опишем работу внутренних блоков СП. Все такие блоки находятся в модуле ХРАМ.

Блок распаковки обращения. Работает сразу после входа в СП. Вычисляет адреса ФКП и номер подпрограммы, а затем устраивает переход по переключателю на блоки засылки ФКП в подпрограммы СП. Этих блоков столько же, сколько подпрограмм в СП. Если параметр используется как значение, то в подпрограмму засыпается значение параметра, иначе – адрес параметра. Затем управление передается на собственно подпрограмму СП.

Блок печати параметров обращений. Для упрощения работы с СП при отладке с помощью подпрограммы ИНФОР можно включать и выключать печать "внешних" (извне СП) и "внутренних" обращений к любой подпрограмме СП. Для реализации этого отладочного режима работы каждая

подпрограмма обращается к блоку печати обращений. Этот блок печатает параметры данной подпрограммы, если соответствующий признак равен единице.

Блок ХРАМ. Содержит тело подпрограммы ХРАМ. Эта подпрограмма позволяет записывать информацию в массив ССК. ХРАМ применяется только разработчиками при отладке СП.

Блок "ПЕЧАТЬ ССК". Используется подпрограммой ИНФОР и блоком печати ошибок (ОШ).

Блок ОШ. При обнаружении ошибки в обращении к некоторой подпрограмме СП происходит печать диагностического текста, печать параметров этой подпрограммы (независимо от заказа), печать ССК и печать текста: "УСТРАИВАЕТСЯ ИСКУССТВЕННОЕ ДЕЛЕНИЕ НА НУЛЬ".

После этого выполняется оператор $A := A / 0$ и управление передается на сбойную программу.

Генерация СП. Поскольку СП может использоватьсь в различных организациях, имеется возможность генерации вариантов СП. При настройке СП на операционную систему и приборы заказчика необходимо менять блоки вывода на графические устройства в подпрограмме БУФЕР и некоторые другие подпрограммы. Все необходимые изменения вносятся в готовую программу СП (результат трансляции с БЕМШ).

Развитие СП. СП Б1 имеет вполне законченный вид и легко адаптируется к различным приборам. Однако уже сейчас просматриваются некоторые направления развития СП. К ним можно отнести следующие:

А). Выделение модуля БУФЕР в отдельную СП и разработка языка общения с ним.

Б). Расширение возможностей модуля СИМ.

В). Усовершенствование системы генерации и внесения изменений.

Г). Добавление возможностей работы с диалоговыми графическими устройствами (дисплеями).

Предварительные попытки развития СП в указанных направлениях были проделаны и нашли свое отражение в статьях настоящего сборника [4] - [7].

Л и т е р а т у р а

1. Математическое обеспечение для граffопостроителей. I уро-
вень. Под ред. А.Я.Куртукова. Новосибирск, ВЦ СО
АН СССР, 1971.
2. Машинная графика и ее применение. - Сборник статей под
ред. Ю.А.Кузнецова. Новосибирск, ВЦ СО АН СССР,
1973.
3. ДВОРЖЕЦ В.И., ЗВЕНИГОРОДСКИЙ Б.Е. Алгоритмы вычерчивания
изолиний. - Настоящий сборник.
4. ДЕБЕЛОВ В.А. Буферная часть СП СМОГ. - Настоящий сборник.
5. ГОРИН С.В. Базовая часть СП СМОГ. - Настоящий сборник.
6. ВОЛОДОКОВ А.М., ДВОРЖЕЦ В.И., ЛУКИНЦОВ А.В. Генерация сим-
волов и чисел в СП СМОГ. - Настоящий сборник.
7. ГОРИН С.В., ДВОРЖЕЦ В.И. СМО дисплея для М-222. - Настоя-
щий сборник.

В.А.Дебелов

БУФЕРНАЯ ЧАСТЬ СМОГ

В качестве объекта описания в настоящей статье будет выступать буферная часть (БУФЕР) системы математического обеспечения графопостроителей (СМОГ) для БЭСМ-6. В некоторых деталях статья будет перекликаться с другими статьями сборника. Такое повторение кажется автору необходимым для обеспечения более или менее связного изложения. О первом варианте БУФЕРА СМОГ БЭСМ-6 можно прочитать в работе [1], об идеологии и строении СМОГ – в работах [2] и [3]. Здесь же система будет затронута настолько, насколько это необходимо для описания БУФЕРА.

СМОГ можно разбить на два слоя – ядро и БУФЕР (см.рис. I). Ядро содержит все основные алгоритмы системы. Эти алгоритмы минимальным образом привязаны к конкретной конфигурации устройств графического вывода (УГВ) обслуживаемых системой.

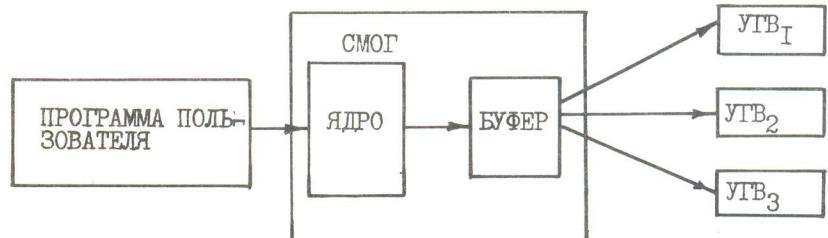


Рис. I. Схема функционирования СМОГ

Ядро всегда работает с "фиктивным" устройством, которое является в некотором смысле "общим знаменателем" всех используемых системой приборов. Роль такого устройства в системе играет БУФЕР. Из связей БУФЕР - УТВ₁ в каждый момент функционирует только одна. По приказу системы действующая связь может быть прервана, а одна из законсервированных станет функционирующей. И если для ядра смена прибора выразится в подаче команды БУФЕРу, то все действия, необходимые для перевода функционирующей связи в состояние консервации и наоборот, ложатся на него.

В настоящее время СМОГ для БЭСМ-6 обслуживает три прибора графического вывода. Приведем их основные характеристики. Считается, что читатель в какой-то мере знаком с приборами типа граffопостроитель. Для справки отсылаем к работе [4].

1. ГРАФОПОСТРОИТЕЛЬ (ГП) "BENSON-220" планшетного типа имеет четыре перодержателя. Прибор шаговый, возможные длины шага - 0,05 мм и 0,1 мм. Не имеет интерполятора, то есть за одно обращение к нему перо продвинется на один шаг по одному из восьми направлений (см.рис.2).

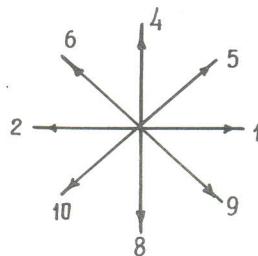


Рис.2

Имеется специальное устройство - раскодировщик "BENSON-411", с помощью которого на ГП можно разрисовывать магнитные ленты (МЛ) с графической информацией, оформленной надлежащим образом. Другими словами, записанные на МЛ команды устройство отрабатывает на ГП.

2. ФОТОПОСТРОИТЕЛЬ (ФП) "BENSON-320" относится к планшетному типу. Служит для получения слайдов. Рисует электронным лучом на фотопленке. Прибор шаговый и без интерполятора. Имеет восемь градаций яркости луча (на языке СМОГ обозна-

чает, что у ФП восемь перодержателей). Единственная возможная длина шага - 0.05 в условных единицах (см. [2]). Как и для ГП имеется устройство - раскодировщик "BENSON-412". Принцип работы тот же.

3. ГРАФОПОСТРОИТЕЛЬ ДГУ-2 рулонного типа имеет три перодержателя. Прибор шаговый, длина шага - 0.2 мм. В состав устройства входит интерполятор, который обеспечивает за одно обращение проведение отрезка прямой, размеры которого по осям координат не превышают 255 шагов.

На указанные три прибора информация в системе может выдаваться по шести каналам. Номер канала задает графический прибор, а также режим выдачи графической информации на этот прибор. В системе задействованы следующие каналы.

1. ГП-планшет OFF - LINE.
2. ФП OFF - LINE.
3. ГП-планшет OFF - LINE "Б".
4. ФП OFF - LINE "Б".
5. ГП рулонный ON - LINE.
6. Не задействован.
7. ГП рулонный OFF - LINE "Б".

I.I. Режим ON - LINE

Этот режим предполагает, что связь ЭВМ с прибором жесткая (см.рис.3).

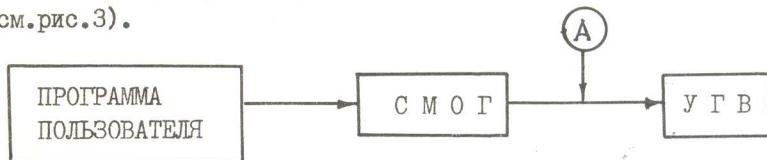


Рис.3

При обработке заданий пользователя система по мере выработки команд управления посыпает их непосредственно на УГВ. Режим ON - LINE используется для оперативного получения рисунков. Но на современных ЭВМ это невыгодно, так как на них в основном используется пакетная обработка задач. Скорость УГВ во много раз меньше скорости ЭВМ, поэтому невыгодно, что графическая задача зря занимает ресурсы вычислительной системы длительное время. Для этого необходима какая-либо буферизация

графической информации, иными словами, отделение счета от рисования. Для этой цели и введены следующие режимы.

I.2. Режим OFF - LINE

По указанной выше причине делается разрыв цепи, как показано на рисунке 3, в точке А. Графическая информация в виде управляющих команд не посыпается на прибор, а перехватывается и накапливается на магнитной ленте (МЛ) (см.рис.4).

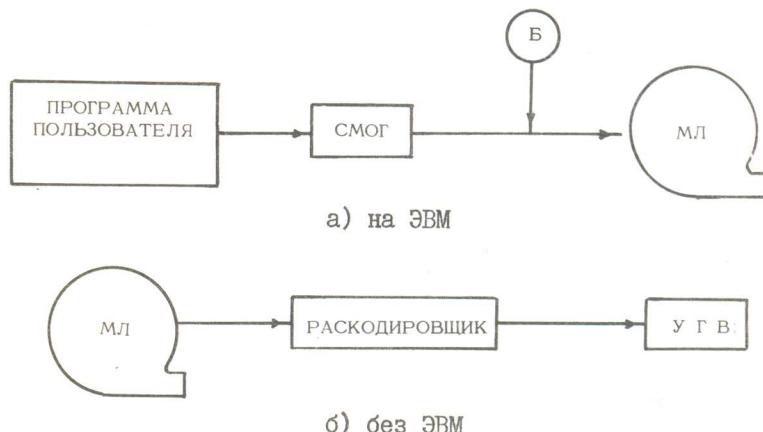


Рис.4

Специальное устройство – раскодировщик – позволило разбить получение рисунка на две фазы. Первая – насчет графической информации, эта фаза выполняется на ЭВМ. Вторая – наиболее медленная – выполняется без затрат машинного времени на автономном устройстве. Такие устройства есть только для планшетного ГП и для ФП, поэтому режим OFF - LINE и используется только для них. Данный режим имеет преимущество перед ON - LINE еще и в том, что не надо дополнительного машинного времени для получения копий рисунков. Недостатком же является то, что информацию для раскодировщика необходимо детализировать вплоть до пошаговых движений, при этом ее объем

сильно возрастает по сравнению с входной информацией СМОГ.

С точки зрения БУФЕРа, устройствами графического вывода являются уже не ГП и ФП, а магнитофон "BENSON - 430", на котором записывается на МЛ подготовленная для этих приборов графическая информация. В первом варианте СМОГ (см. [I]) для ГП и для ФП использовались разные магнитные ленты. Поэтому оператору иногда приходилось по нескольку раз менять бобины даже во время решения одной задачи. В настоящем варианте рисунки по каналам I и 2 записываются на одну и ту же магнитную ленту, но в ее каталоге отмечается, для какого они прибора. Эта информация необходима для оператора прибора-раскодировщика.

I.3. Режим OFF - LINE "Б"

При дальнейшем развитии системы возникает задача организации архивов рисунков, подрисунков, узлов и тому подобное. Но информация, накопленная на МЛ при использовании режима OFF - LINE, не подходит для этих целей ввиду своего большого объема, а также из-за жесткой ориентации на определенный графический прибор. Вопрос организации архивов должен решаться в рамках всей системы СМОГ и здесь рассматриваться не будет. Зато задача БУФЕРа - организация вывода графической информации на стандартный носитель ЭВМ в компактном виде, допускающая простой перевод в управляющие команды для УГВ, решена и будет нами рассмотрена. Схема, изложенная на рисунке 4, опять прерывается в точке Б. О выборе этой точки в программе БУФЕРа будет сказано при разборе алгоритма каналов 3, 4 и 7. На рисунке 5 показан технологический процесс получения рисунков при режиме OFF - LINE "Б", который задействован в настоящее время на БЭСМ-6. Для каналов I и 2 он проходит в три этапа:

I. Вывод графической информации в компактном виде на магнитный диск (МД).

II. Перепись информации с МД на ту же МЛ, что и при режиме OFF - LINE. Естественно, что при переписи идет перекодировка информации.

III. Разрисовка МЛ, аналогично второй фазе OFF - LINE. Здесь уже два этапа используют ЭВМ, но на втором этапе в от-



Рис. 5.

личие от первого, работает одна и та же системная программа СЕРВГП. Она производит перепись МД ↔ МЛ сразу для всех рисунков, которые накопились на диске за определенный промежуток времени (смена, сутки и так далее). При пакетной обработке такой режим предпочтительнее режима OFF - LINE, так как в OFF - LINE магнитофон "BENSON - 430" должен быть включен все время, а по техническим условиям он выдерживает не более четырех часов непрерывной работы. Вторая положительная черта OFF - LINE "Б" заключается в возможности одновременного решения сразу нескольких графических задач! В то время как в режиме OFF - LINE первая из них, выходя в решение, занимала магнитофон до своего окончания, независимо от веса ее графической части. "Одновременность" решения достигается с помощью принципов параллельного программирования, изложенных в статье Дийкстры в работе [5].

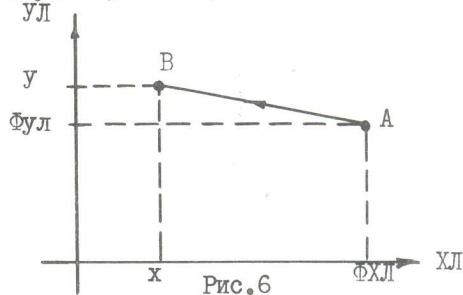
Все эти соображения верны и для канала 7. В этом случае процесс получения рисунков проходит два этапа. В качестве стандартного носителя используется обычная магнитная лента.

Перейдем к описанию алгоритма БУФЕРа, реализованного в СМОГ для БЭСМ-6 с описанной выше конфигурацией графических приборов и каналов вывода информации из них.

2. Алгоритм буфера

2.1. Входной язык БУФЕРа

Основная задача БУФЕРа – обеспечить для СМОГ проведение прямых на конкретных устройствах. Каждое рабочее обращение к БУФЕРу позволяет продвинуть перо графопостроителя в заданную точку на листе бумаги. БУФЕР имеет три параметра: x, y, j . Их назначение покажем на примере (см.рис.6). Переменные ФХЛ, ФУЛ в каждый момент содержат координаты пера на листе (точка А). Тогда по рабочему обращению БУФЕР выдаст на ГП команды продвижения пера из точки А в точку В. Причем перо поднято, если $j = 0$, и опущено, если $j = 1$.



БУФЕР работает только с физическими координатами – миллиметрами, поэтому Х и У должны быть выражены в них (о разных системах координат см. в работе [6]).

В варианте СМОГ БЭСМ-6 БУФЕР и ЯДРО имеют общие переменные (см.работу [3]). Введем те из них, значения которых существенно влияют на алгоритм или поясняют его.

1. НПЕРА – номер передержателя (интенсивности подсветки).
2. ДЛШАГА – длина шага в мм.
3. ПРИБОР – номер действующего канала вывода графической информации.
4. ФХЛ, ФУЛ – координаты пера на листе в мм.

5. ФОРМАТ – формат листа бумаги, необходимый для рисунка.
6. ДЛИНА – общая длина пути, пройденного пером.
7. МАХДЛ – максимальное значение ДЛИНЫ, заказанное пользователем.

Значение ДЛИНА насчитывается в БУФЕРе, и как только оно превышает значение МАХДЛ, управление передается на блок системы – ОЩДЛ. Начальные значения переменных ФХЛ и ФУЛ нулевые. В процессе работы БУФЕР их пересчитывает. Например, для рисунка 6:

$$\begin{aligned} \text{ФХЛ новое} &= X, \\ \text{ФУЛ новое} &= Y. \end{aligned}$$

Для сообщения в БУФЕР об изменениях значений НПЕРА и ДЛШАГА программы системы используют служебные обращения БУФЕР (X, 2, -9) и БУФЕР (X, 3, -9) соответственно. Значение параметра X для всех служебных обращений несущественно. Еще одно обращение информативного характера – БУФЕР (X, 1, -9) обозначает, что пользователь закончил рисовать лист (рисунок) и перешел к следующему (или начинает первый лист). И, наконец, последний тип служебного обращения – БУФЕР (X, -9, -9) сообщает БУФЕРу, что работа СМОГ окончена и необходимо перевести в состояние консервации функционирующую связь СМОГ-УГВ. На этом описание входного языка закончим и перейдем к самому алгоритму.

БУФЕР фактически является комплексом программ. Он состоит из головной программы, интерполирующей (МУР), и набора построенных по одинаковому принципу программ обработки информации для каждого графического устройства, обслуживаемого системой.

2.2. Головная программа

Из рисунка 7 видно, что головная программа управляет работой БУФЕРа. Она состоит из блока анализа служебных обращений (рис.8) и блока обработки (рис.9). Поясним назначение переменных на указанных блок-схемах.

ПРИЗ1, ПРИЗ2, ПРИЗ3 – индикаторы, в них отмечается наличие соответствующего служебного обращения.

КС, КН содержат номер канала и обычно равны. При смене канала КС равна номеру работавшего до настоящего момента ка-

нала, а КН - номеру, к которому необходимо подключить СМОГ.

ЦЕЛО - индикатор; равен 1, если еще не активизировался ни один канал (то есть БУФЕР еще не работал).

КОНЕЦ - индикатор, устанавливается в 1 обращением БУФЕР ($X, -9, -9$). В этом случае переменная КН получает значение -9, чтобы направить алгоритм на смену канала.

Наличие переменных ПРИ1, ПРИ2, ПРИ3 кажется избыточным. Это справедливо в какой-то мере для БУФЕРА как подпрограммы СМОГ, но он в системе мыслился как самостоятельный блок (нулевой уровень матобеспечения графопостроителей). В последнем случае обращения вида БУФЕР ($X, Y, -9$), где $Y = 2, 3$, становятся необходимыми. Увеличивая в данном случае диапазон изменения Y , такие обращения можно использовать для передачи таких параметров, как НПЕРА, ДЛШАГА, ПРИБОР и так далее. Для этого используется незадействованный параметр X в обращении БУФЕР ($X, Y, -9$).

2.3. Слуга (пар)

Это программа - коммутатор, она передает параметр ПАР из головной программы в программу Π_i , если функционирует связь БУФЕР - УГВ $_i$. Другими словами, i - это номер работающего канала (значение переменной КС). Выше отмечалось, что программы Π_i , $i = 1, \dots, n$, должны быть организованы одинаково. В зависимости от значения ПАР Π_i должна выполнять следующие приказы головной программы.

1. Открыть канал. Для каналов, где счет и разрисовка разделены, такой приказ особенно важен. По нему происходит считывание каталогов с внешних устройств (МЛ или МД), определение свободного места и тому подобное.

2. Закрыть канал. Перед отказом от канала необходимо произвести соответствующие отметки в каталогах, чтобы не "потерять" только что накопленную информацию.

3. Выдать очередной квант информации в канал.

4. Открыть лист (рисунок).

5. Закрыть лист.

Все приказы, за исключением третьего, относятся к работе с архивами на МЛ или МД. Третий приказ - рабочий, только он

занимается выдачей графической информации. Для некоторых каналов один или несколько приказов могут быть пустыми, то есть по ним не будет выполняться никаких действий. Например, для канала 5 (программа Π_5) первые два приказа пустые. По четвертому обнуляется значение переменной МЕТРЫ (путь пера в миллиметрах в пределах одного листа), а по пятому приказу выдается сообщение пользователю о том, что лист отрисован и длина линии такая-то. Наиболее громоздкой получается программа Π_1 для каналов 1 и 2. Так как графическая информация имеет очень большой объем, то часто она не помещается на одной МЛ и приходится переносить ее на другую, третью и так далее. Поэтому третий приказ – самый сложный.

Наиболее интересный случай – программа Π_3 для каналов 3 и 4. В этом случае в памяти ЭВМ может находиться несколько программ, каждая из которых работает с этими каналами. Основная трудность заключается в том, как исключить "одновременную" работу с каталогами МД в то время, когда архивный диск доступен им всем. В ОС ДИАПАК (и ЛИСПАК) имеются экстракоды захвата и освобождения диска. Они-то и разрешают создавшуюся трудность, так как между обращениями к этим экстракодам никакая другая задача не сможет работать с архивным диском. На МД две зоны отводятся под каталоги К1 и К2. Первое слово К1 содержит количество листов (рисунков) на диске. На информацию о каждом листе отводится по два слова. В первом из них шифр пользователя, во втором – сведения о листе:

- а) номер канала – 3 или 4,
- б) формат бумаги для рисунка,
- в) длина пути пера в метрах,
- г) номер первой зоны архива (НЗОНЫ) с информацией для данного листа,
- д) дата записи в архив.

Для каждого листа в каталоге К2 находится односвязный список. Номер первого его элемента равен НЗОНЫ. Содержимое каждого элемента не только определяет преемника, но и номер следующей зоны с информацией для этого листа. Такое строение каталога К2 позволяет записывать информацию в зоны, не обязательно следующие друг за другом.

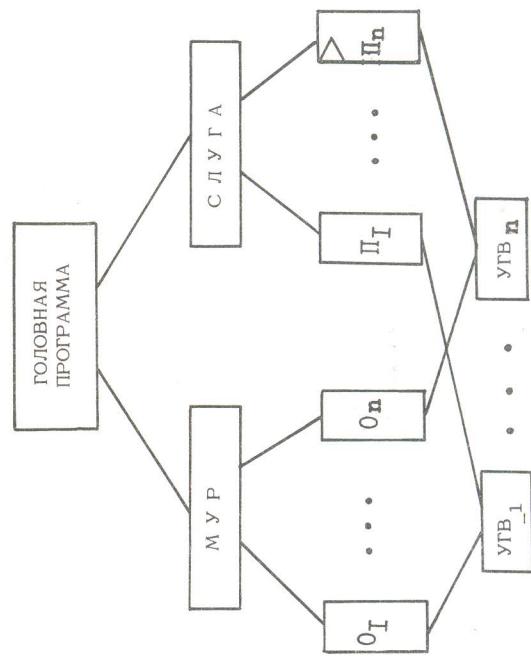


Рис.7. Функциональная схема БУФЕРа

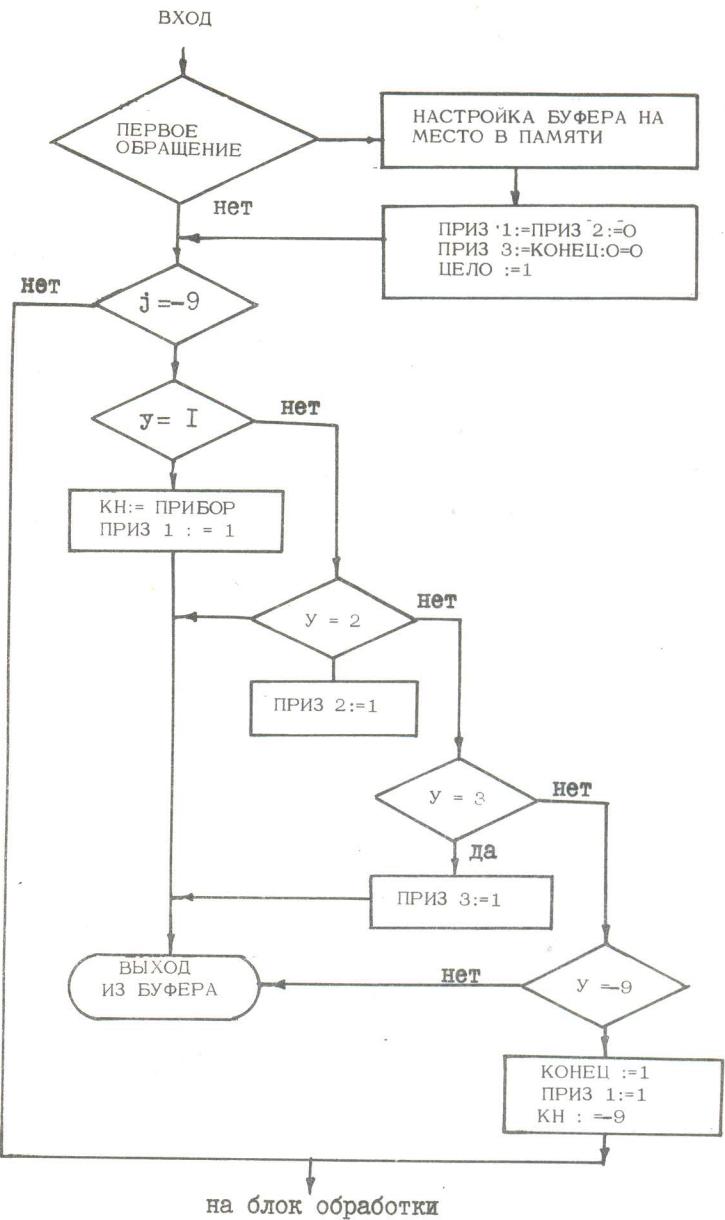


Рис.8. Блок-схема анализа служебного обращения

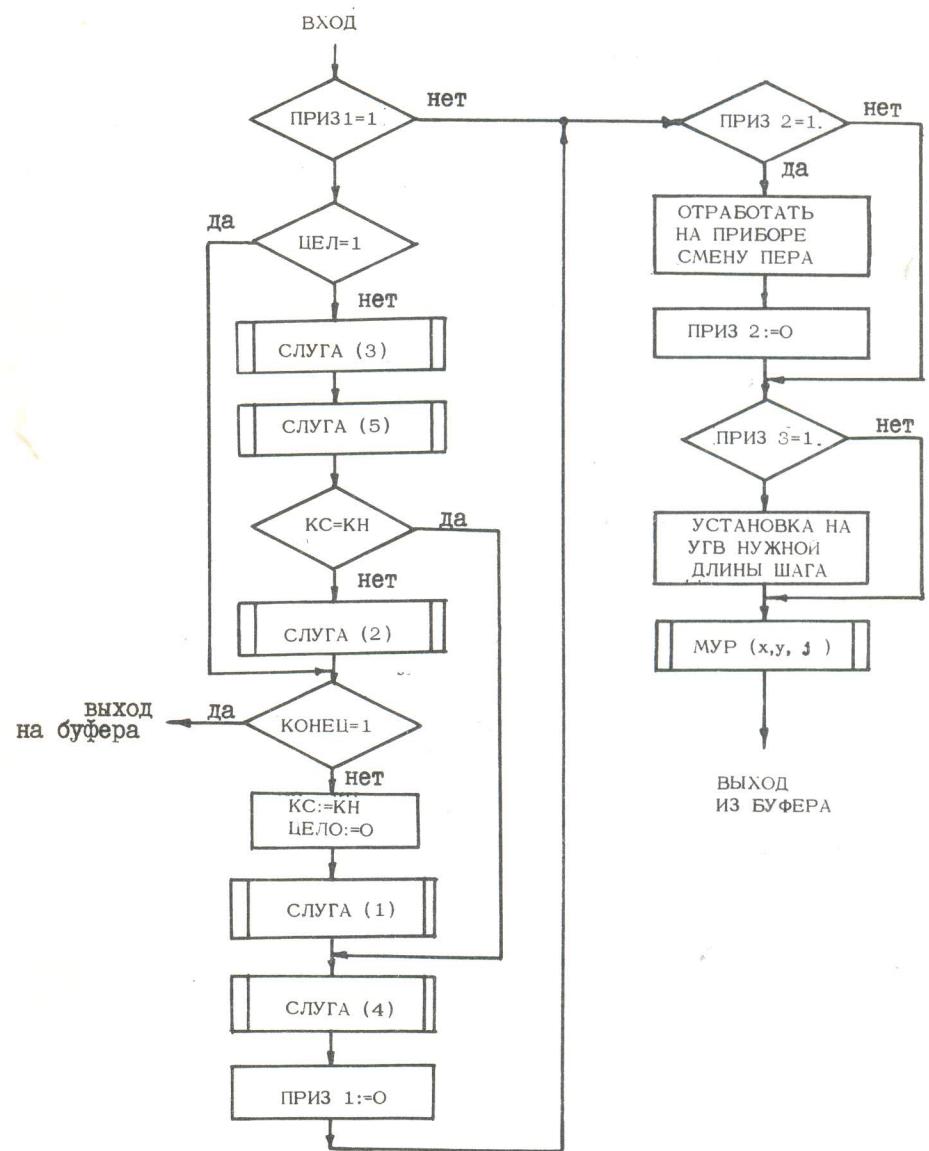


Рис.9. Блок-схема блока отработки

2.4. МУР (X, Y, J).

Эта также основная подпрограмма БУФЕРа, она имеет непосредственный доступ к УТВ для режима ON - LINE или через соответствующую программу Π_1 в других режимах. Пример, показанный на рисунке 6, можно отнести и к МУРу. Другими словами, МУР выполняет рабочие обращения. Основная его функция – интерполяция прямой.

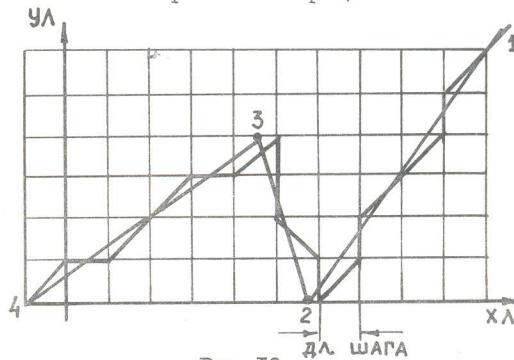


Рис.10

На рисунке 2 было показано, что все обслуживаемые системой приборы имеют только восемь направлений движения. Задача интерполятора состоит в том, чтобы, используя дискретные направления и фиксированную длину шага, обеспечить наименьшую погрешность при проведении прямой. Принцип интерполяции показан на рисунке 10. Координаты X и Y не обязательно должны нацело делиться на длину шага. Возникающая при этом ошибка, как видно из рисунка, никогда не превышает половины шага интерполяции. Если учитывать, что минимальная толщина линии (пера) современных графопостроителей порядка 0.1 мм, то на глаз интерполяцию на рисунке обнаружить трудно.

Результатом работы МУРа не обязательно должна быть информация, детализированная с точностью до шагов. Вид ее зависит от потребности того или иного канала. На рисунке II приводится блок-схема МУРа. Выделенная точка А показывает, где информация разделяется по каналам.

Переменная СОСП (состояние пера) равна 0, если перо поднято, и 1, если опущено. Перед тем как попасть на УТВ информация проходит последний этап – обработку программой O_1 .

I. Подпрограмма O_1 , получая с выхода интерполятора по одному коду очередного движения, накапливает их в рабочем массиве. После того как накопится информации полностью на зону

МЛ, управление передается программе Π_1 с параметром ПАР=3 (запись зоны на МЛ). Так как информация для каналов 1 и 2 имеет одинаковый вид (с точки зрения БУФЕРа), то O_1 работает и для канала 2.

2. Подпрограмма O_3 . На ее вход информация приходит в виде целых значений приращений, выраженных в шагах - ПХ и ПУ. Как уже отмечалось в первой половине статьи, выходная информация БУФЕРА в режиме OFF - LINE "Б" должна быть компактной и в то же время достаточно простой для распаковки. Первоначально предполагалось исходную информацию для кодировщика брать с выхода интерполятора. Но этим мы ограничивали круг устройств графического вывода, которые можно было бы использовать в режиме OFF - LINE "Б". Так как отрабатывать информацию по одному шагу на рулонном ГП (см.рис.5б) было бы расточительно (прибор интерполирующий), поэтому оказалось удобнее выбрать точку А - вход интерполятора. Второй довод в пользу такого выбора - это возможность унификации кодировок, применяющихся в системе (см. [7]).

2.5. Кодирование графической информации

Каждый элемент кодировки содержит либо координаты отрезка, который надо провести на приборе, либо одну из управляющих команд, таких, например, как "поднять перо", "опустить", "сменить шаг" и тому подобное. Управляющие команды, кроме первых двух, появляются довольно редко (в большинстве рисунков не более одного раза), поэтому длина элемента для них может быть большой, и это не скажется на эффективности кодировки.

Из опыта работы с графопостроителем замечено, что наиболее часто используются линии длиной от 1 мм до 10 см. Для таких линий необходимо, чтобы элемент кодировки имел по возможности меньшую длину.

За основу кодировки взята тетрада - 4 двоичных разряда. Каждый элемент содержит целое число тетрад, но не меньше двух. Наиболее часто встречающиеся случаи кодируются элементами с одной управляющей тетрадой, остальные - двумя.

Итак, элементы с одной управляющей тетрадой T_1 . Она разбивается на две части: p и q , как показано на рисунке 12.

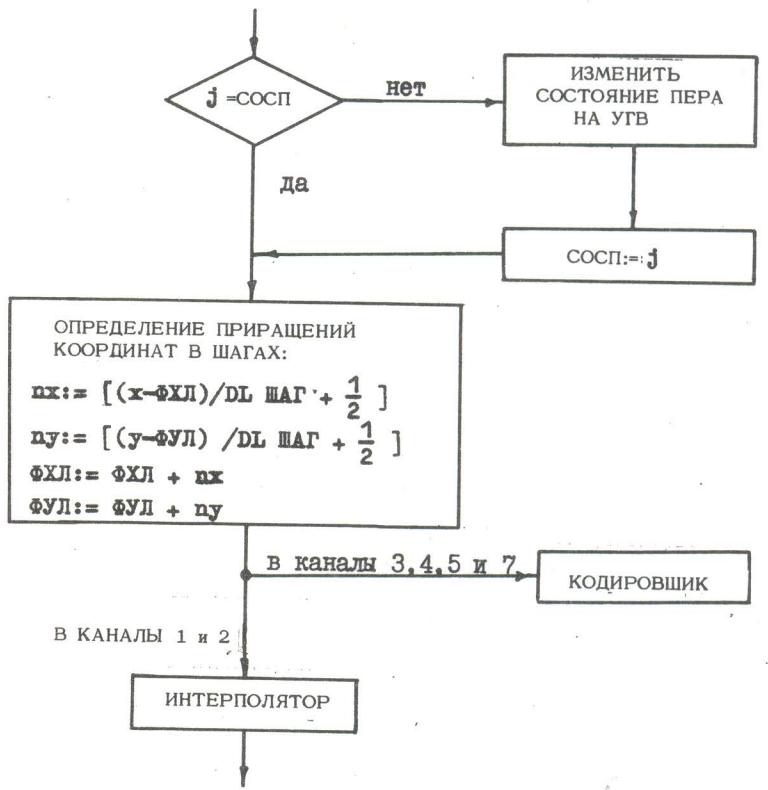


Рис. II. Блок-схема МУРа

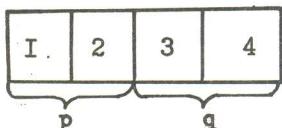
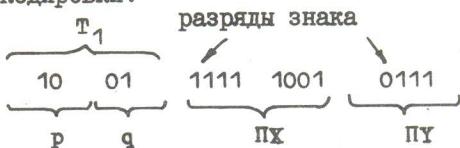


Рис.12

Ясно, что они могут иметь целые значения от 0 до 3. Если $p+q \neq 0$, то следом за T_I идут p тетрад со значением приращений по X

(ПХ) и q тетрад со значением приращений по Y (ПУ). Знак приращения помещается в первый разряд первой тетрады, если оно не нуль. Поясним на примере.

$\text{ПХ} = -17T_8$, $\text{ПУ} = +7$, тогда $p = 2$ и $q = 1$; элемент кодировки:



В таблице приводится зависимость между значением p и длиной ПХ в шагах без учета знака.

P	0	1	2	3
ПХ max	0	7	27	2047
ПХ min	0	1	8	128

Если взять шаг прибора равным 0,05 мм, то отрезки, приращения которых не превышают 10 см, будут кодироваться элементами с одной управляющей тетрадой.

Если $T_I = 0$, то необходима еще одна управляющая тетрада $-T_2$. Здесь возможны следующие виды элементов.

а) $T_2 = 8, 9, 10$. Тогда элемент, кроме управляющих тетрад, содержит ($T_2 - 4$) тетрады с ПХ и столько же с ПУ. В данном случае мы отводим одинаковое число тетрад для ПХ и ПУ, но это оправдывается тем, что это линии достаточно большой длины (больше 10 см).

б) $T_2 = II$. Команда подъема пера. Этот элемент состоит только из двух управляющих тетрад T_I и T_2 .

- в) $T_2 = T_2$. Команда опускания пера.
- г) $T_2 = T_2$. Команда смены пера. Номер нового пера находится в двух тетрадах, следующих за T_2 .
- д) $T_2 = T_2$. Команда смены длины шага. Константа с новой длиной в нормализованном виде занимает T_2 тетрад, следующих за T_2 .
- е) $T_2 = T_2$. Конец информации.
- ж) $T_2 < 8$. Элементы с этими значениями T_2 БУФЕР не вырабатывает. Они являются резервными. Дополнительные сведения об этой кодировке приводятся в работе [7].

З а к л ю ч е н и е

Модульная структура БУФЕРа позволяет довольно быстро подключить к СМОГ новое устройство графического вывода. Достаточно для этого написать программки О и П для него (см.рис. 7), что обычно не очень сложно. Предполагается дальнейшее развитие алгоритма головной программы. В настоящее время подключение нового УТВ ведет к увеличению объема БУФЕРа, то есть пользователь вынужден жертвовать все большим объемом памяти. Поэтому предполагается головную программу оформить как резидент, а соответствующие программы обработки вызывать в память при смене канала.

Л и т е р а т у р а

1. Куртуков А.Я., Горин С.В., Дворжец В.И., Дебелов В.А. Система для БЭСМ-6. - Сб. Машинная графика и ее применение. Новосибирск, 1973.
2. Куртуков А.Я. Система математического обеспечения для задач с графическим выводом. - Сб. Машинная графика и ее применение. Новосибирск, 1973.
3. Горин С.В., Дворжец В.И., Дебелов В.А. Структура СМОГ БЭСМ-6. - Настоящий сборник.
4. Математическое обеспечение графопостроителей (I уровень). Под ред. А.Я.Куртукова. ВЦ СО АН СССР, 1971.

5. Диикстра Э. Взаимодействие последовательных процессов.- Сб. Языки программирования. Под ред. Ф.Женюи. М., "Мир", 1972.
6. Горин С.В. Базовая часть СМОГ.- Настоящий сборник.
7. Володюков А.М., Дворжец В.И., Лукинцов А.В. Генерация символов и чисел в СП СМОГ.- Настоящий сборник.

С.В.Горин

БАЗОВАЯ ЧАСТЬ СМОГ

В данной статье будет описана структура базовой части СМОГ, то есть набора подпрограмм, обеспечивающих проведение отрезка прямой, защиту области вне листа и работу с декартовой системой координат. Для более полного понимания статьи полезно знакомство с работами [1], [2], хотя оно и не является обязательным.

При работе со СМОГ пользователю отводится ограниченная прямоугольная область, и перо не должно выходить за ее пределы. Задача защиты области вне заказанного листа носит среди разработчиков название ФУЛПРУФ (от английского foolproof). Пусть нам надо соединить прямыми точки 1,2,3,4,5,6 в порядке возрастания номеров. Возможно несколько вариантов решения этой задачи (см.рис. I-3).

Первые два варианта довольно просты в реализации. Выбор последнего существенно усложняет блок ФУЛПРУФ (блок СП СМОГ занимается контролем выхода за лист). Тем не менее был выбран именно этот вариант из-за его естественности и высокой информативности. Кроме того, преимуществом этого варианта является возможность получать часть рисунка простым увеличением масштаба.

В выделенной области вводится листовая система координат. Это прямоугольные координаты с ориентацией осей по сторонам

•3

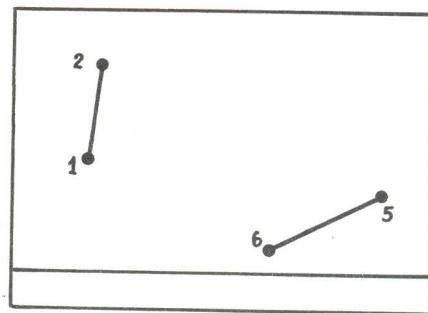


Рис.1. Линии, выходящие за лист, вообще не чертятся

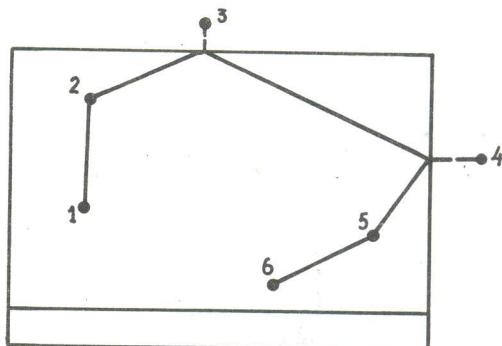


Рис.2. Точки, лежащие вне листа, стягиваются на границу

•3

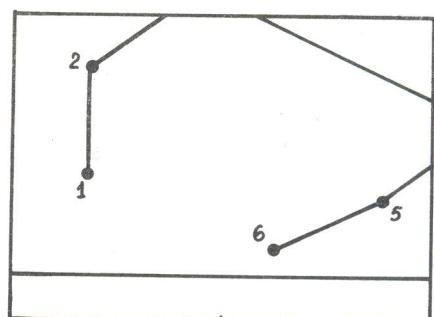


Рис.3. Участки линий, выходящие за пределы листа, просто не чертятся

листа и с миллиметром в качестве единицы измерения по обеим осям. Начало координат находится в левом нижнем углу листа. По желанию пользователя на выделенной области может вводиться и декартова система координат. В отличие от листовой системы координат, масштаб по осям указывается самим пользователем и начало координат не имеет жесткой связи с листом. Ориентация осей всегда совпадает с листовой. Для "закрепления" такой системы на листе используется "точка привязки", то есть указываются координаты одной и той же точки в листовой и декартовой системе координат. Точка выбирается произвольно. Декартовы системы координат могут сменять друг друга на одном и том же листе. Для задания декартовой системы координат служит подпрограмма ДЕКАР. Ее параметрами являются: масштабы (число миллиметров в единице измерения) по осям X и Y, листовые и декартовы координаты точки привязки. При отсутствии обращения к подпрограмме ДЕКАР декартова система совпадает с листовой, то есть считается, что введена декартова система координат с масштабами, равными единице, по осям. Началом координат является левый нижний угол листа. Для преобразования декартовых координат в листовые и наоборот служат подпрограммы ДЕЛ и ЛИД (см. [1]). Для понимания работы описанных подпрограмм необходимо знакомство с еще одной подпрограммой СМОГ – подпрограммой ХРАМ. Это внутренняя подпрограмма СП СМОГ (см. [4]), предназначенная для выполнения служебных функций по входу и выходу из СП. Кроме того, она содержит вышеописанные процедуры по работе с декартовыми координатами, а также все константы, переменные и массив ССК. Массив ССК (Состояние Комплекса) содержит 32 переменные, характеризующие состояние комплекса в каждый текущий момент. Подробно он описан в работе [4]. В частности, в ССК хранятся масштабы по осям, а также листовые и декартовы координаты пера. При заказе листа в масштабы заносятся единицы, а листовые и декартовы координаты зануляются. Подпрограмма ДЕКАР изменяет масштабы, подводит перо в точку привязки и записывает указанные листовые и декартовы координаты в ССК. Тем самым декартова система координат фиксируется на листе. Подпрограммы ЛИД и ДЕЛ для пересчета координат из одной системы в другую используют переменные из ССК. Формулы пересчета имеют вид:

Подпрограмма ЛИД (листовые координаты в декартовы)

$$Хд = (Хл - ХЛИСТ) / МХ + ХДЕК$$

$$Уд = (Ул - УЛИСТ) / МУ + УДЕК$$

Хл, Ул - листовые координаты, параметры подпрограммы;

Хд, Уд - декартовы координаты, результат работы подпрограммы.

Подпрограмма ДЕЛ (декартовы в листовые)

$$Хл = (Хд - ХДЕК) * МХ + ХЛИСТ$$

$$Ул = (Уд - УДЕК) * МУ + УЛИСТ$$

Хд, Уд - декартовы координаты, параметры подпрограммы;

Хл, Ул - листовые координаты, результат работы подпрограммы.

В обоих случаях

ХЛИСТ, УЛИСТ - листовые координаты пера

ХДЕК, УДЕК - декартовы координаты пера

МХ, МУ - масштабы по осям

Все переменные из ССК.

Перейдем к описанию работы основной рисующей подпрограммы СМОГ - ТРА. С ее помощью можно соединять отрезком две произвольные точки листа. Назовем начальную точку точкой ухода, а конечную - точкой прихода. Координаты точки ухода, вообще говоря, должны задаваться положением пера перед обращением к ТРА. Однако рассмотрим небольшой пример. Пусть нам надо соединить прямыми точки АВС. Как уже говорилось, должен получиться рисунок 4.

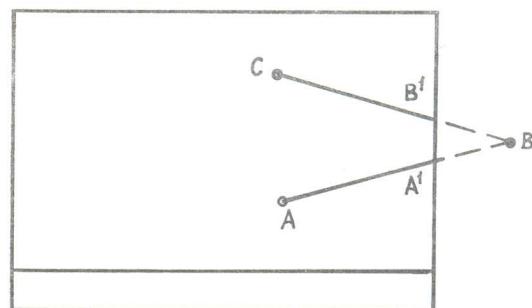


Рис.4.

При попытке провести линию АВ реально проводится только АА'. Теперь мы хотели бы провести линию ВС. Точкой ухода является точка В, хотя на самом деле перо находится в точке А'. Таким образом, приходим к понятию физических и математических координат. Под физическими координатами понимаются координаты точки реального нахождения пера. Математические координаты – это те, которые были поданы последний раз на входе ТРА. Заметим, что абсолютно все передвижения пера производятся только через обращения к ТРА. Они различаются только для случая, когда поданные координаты указывают точку вне указанного листа. Тем не менее для нормальной работы необходимо всегда хранить как те, так и другие координаты. Упомянутые выше ХЛИСТ, УЛИСТ, ХДЕК, УДЕК суть текущие математические листовые и декартовы координаты пера. Физические координаты также хранятся в массиве ССК. Физические координаты всегда листовые. К этому вопросу мы еще вернемся при описании блока ФУЛПРУФ подпрограммы ТРА. Итак, математические координаты пера являются координатами точки ухода. Координаты точки прихода задаются первыми двумя параметрами ТРА. Значение третьего параметра равно нулю при поднятом пере и единице – при опущенном. Первые два параметра ТРА могут трактоваться и как приращения к текущим координатам. Тогда значение третьего параметра равно двум для поднятого и трем – для опущенного пера. С помощью подпрограммы ТРА можно работать с декартовой системой координат. В этом случае первые два параметра понимаются как декартовы координаты или приращения, а третий параметр увеличивается на 4. Подпрограмма ТРА состоит из трех блоков.

I. Блок нахождения координат точки прихода. Анализируя значение третьего параметра, легко получить способ задания координат точки прихода. Либо это явные координаты, и тогда все очевидно, либо их легко получить добавлением приращений к математическим координатам из массива ССК. Преобразование листовых координат в декартовы и наоборот производится через обращение к подпрограммам ЛИД и ДЕЛ. В результате получаем листовые и декартовы координаты точки прихода и состояние пера. В дальнейшем работа ведется только с листовыми координатами. Листовые координаты точки ухода и точки прихода перепи-

сываются в рабочий массив, а в ССК записываются листовые и декартовы координаты точки прихода. Затем начинает работу блок 2.

2. Блок ФУЛПРУФ. Поиск точек пересечений со сторонами листа. В настоящее время количество различных типов графических устройств весьма велико. Это шаговые и интерполирующие, рулонные и планшетные граffопостроители, фотопостроители и так далее.

Описываемый графический комплекс обладает достаточной универсальностью. Все подпрограммы работают с некоторым абстрактным прибором, который включает в себя по возможности все свойства графических устройств. Роль такого прибора выполняет подпрограмма БУФЕР (см. [1], [2], [3]).

Для пользователя, работающего с комплексом, отрезок прямой проводит ТРА и только ТРА. На самом деле реальное проведение отрезка производит подпрограмма БУФЕР. На входе ей подаются координаты точки прихода и состояние пера. Координаты точки ухода суть физические листовые координаты из ССК.

Подпрограмма ТРА работает с математическими координатами. Подпрограмма БУФЕР всегда работает только с физическими. В этом их основное различие. Подпрограмме БУФЕР подаются координаты точки, про которую точно известно, что она лежит внутри заказанного листа. В зависимости от возможностей прибора БУФЕР должен привести перо из точки ухода (физические листовые координаты пера) в указанную точку. Подпрограмме ТРА подаются математические координаты, то есть координаты точки, куда пользователь хотел бы привести перо. Результатом работы ТРА должны быть координаты точки, возможно, отличной от заданной, но лежащей внутри листа. Такая точка отыскивается согласно описанной выше задаче ФУЛПРУФ. Координаты найденной точки (если она вообще существует) подаются на вход подпрограммы БУФЕР, и перо приводится в нужную точку. Таким образом, ТРА фактически только контролирует выход за пределы заказанного листа и организует нужные обращения к БУФЕРу. Поэтому блок 2 в первую очередь проверяет, лежат ли обе точки (ухода и прихода) внутри заказанного листа. Если это так, то происходит обращение к подпрограмме БУФЕР, и работа заканчивается. В противном случае может возникнуть одна из четырех ситуаций.

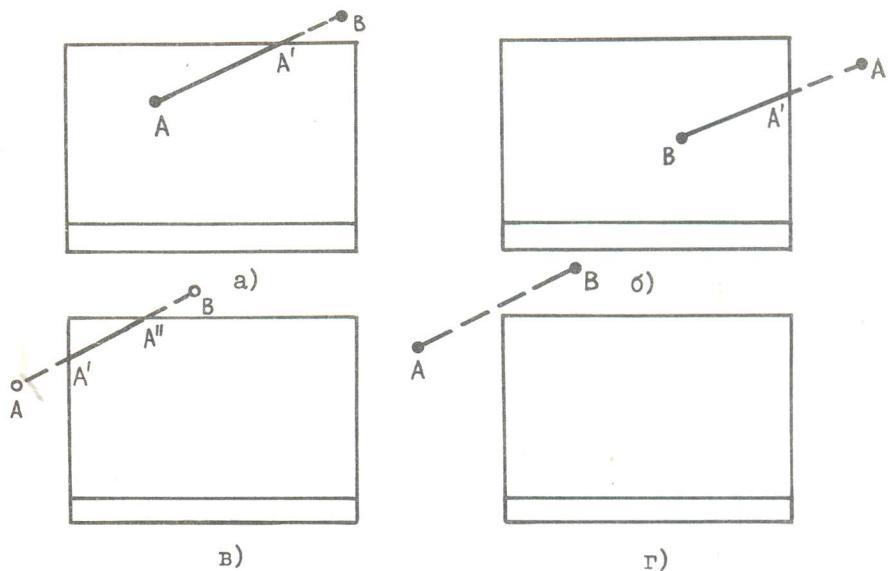


Рис.5

Все известные алгоритмы решения задачи ФУЛПРУФ основаны на одном принципе: необходимо каким-либо образом выделить стороны, точно имеющие пересечения с отрезком, и найти точки пересечения. Вообще говоря, в некоторых частных случаях эта задача может быть решена довольно просто. Для шаговых приборов этот блок может быть встроен в интерполятор, то есть на каждом шаге нужно проверять выход за лист и при достижении края листа прекращать рисование. Команды проверки могут входить во временную задержку и не будут влиять на время рисования.

В процессе создания СМОГ было опробовано несколько вариантов блока ФУЛПРУФ. В результате предпочтение было отдано, может быть, не самому быстрому и не самому малому по числу команд (для варианта БЭСМ-6 ТРА занимает 270_8 ячеек, для М-220-230₈), но зато наиболее наглядному варианту.

Блок 2 должен найти пересечения со сторонами листа и перейти к блоку 3. Основной подпрограммой блока 2 является подпрограмма ТАПЕР.

Подпрограмма ТАПЕР предназначена для нахождения координат точки пересечения отрезка с заданной стороной листа или ее продолжением, а также для отбраковки точек пересечения, лежащих на продолжении стороны листа и тех точек, которые найдены повторно (отрезок проходит через угол листа). Одно обращение к ТАПЕРу находит (или не находит) точку пересечения с одной заданной стороной листа. Поскольку сторон у листа четыре, то и обращений к ТАПЕРу в общем случае будет четыре. Но обращение к ТАПЕРу будет идти лишь до тех пор, пока не будут найдены две различные точки пересечения. Найденные точки пересечения и их число заполняются. При обращении к подпрограмме ТАПЕР производятся следующие действия.

1) Проверяется число уже найденных точек пересечения. Если их две и они не совпадают (совпадение может быть в случае прохода через угол листа), то переходим к блоку 3.

2) Проверяется, лежат ли точки ухода и прихода в разных полуплоскостях, разделяемых заданной стороной листа. Если нет, то работа ТАПЕР заканчивается.

3) Иначе – отыскивается точка пересечения отрезка с заданной стороной листа.

4) Проверяется, лежит ли найденная точка на заданной стороне листа, так как возможен случай пересечения отрезка с продолжением стороны. Если не лежит, то работа ТАПЕР заканчивается.

5) Иначе – точка запоминается и число найденных точек увеличивается. Работа ТАПЕР заканчивается.

Фактически блок 2 состоит из четырех обращений к подпрограмме ТАПЕР.

3. Блок подготовки обращений к БУФЕРу.

Итак, точки пересечений со сторонами листа найдены. Рассматриваются три основные ситуации:

3.1. Точек пересечения нет. Случай г) рисунок 5. Это означает, что обе точки лежат вне листа и работа ТРА заканчивается.

3.2. Точка пересечения одна. Это может означать одно из трех:

3.2.1. Выход за лист. Случай а)-рисунок 5. Обращение к БУФЕРу-перо приводится в точку А' и работа ТРА заканчивается.

3.2.2. Вход в лист. Случай б)-рисунок 5. Обращениями к БУФЕРу перо приводится сначала в точку А" (в поднятом состоянии), а затем в точку В. Работа ТРА заканчивается.

3.2.3. Точка ухода и точка прихода лежат вне листа, то есть отрезок проходит через угол листа. Работа ТРА заканчивается.

3.3. Точек пересечений две. Случай в)-рисунок 5. Обращениями к БУФЕРу перо приводится сначала в точку А' (в поднятом состоянии), а затем в А". Работа ТРА заканчивается.

В заключение заметим следующее. Контроль выхода за пределы заказанного листа производится всегда. Пользователь по своему желанию может задать два режима контроля: мягкий и жесткий. При мягком режиме отрезки прямых, лежащие вне листа, просто игнорируются. При жестком режиме первая же попытка выйти за пределы листа является ошибкой. Перо приводится в точку пересечения со стороной листа, на печать выдаются координаты точек ухода и прихода, и СМОГ свою работу заканчивает.

Л и т е р а т у р а

1. Математическое обеспечение для граffопостроителей. I уровень. Под редакцией А.Я.Куртукова. Новосибирск, 1971, с.82.
2. Куртуков А.Я. Система математического обеспечения для задач с графическим выводом.-Сб. Машинная графика и ее применение. Новосибирск, 1973. с.14.
3. Дебелов В.А. Буферная часть СМОГ. - Настоящий сборник.
4. Горин С.В., Дворжец В.И., Дебелов В.А., Куртуков А.Я. Структура СМОГ БЭСМ-6. - Настоящий сборник.

А.М.Володюков, В.И.Дворжец, А.В.Лукинцов

ГЕНЕРАЦИЯ СИМВОЛОВ И ЧИСЕЛ В СП СМОГ

Выход символов в СП СМОГ осуществляется подпрограмма СИМ, вывод чисел – подпрограмма ЧИСЛО. Обращения к ним описаны в работе [I].

I. Генерация символов

Во время разработки СП СМОГ основные споры вызывали дискуссии по проблемам описания и отрисовки символов. Эта проблема разбивается на две связанные задачи:

А). Кодировка движений в некотором массиве МКС (Массив Кодировки Символов). МКС формируется в момент генерации варианта СП СМОГ.

Б). Разработка декодирующего алгоритма (декодера), который реализуется в подпрограмме СИМ.

Выбрав некоторый тип кодировки символов (и следовательно, декодера), мы надолго определяем "кодовую политику" СМОГ в области кодировки графической информации. Действительно, было бы очень заманчиво использовать одну и ту же схему кодирования для символов и для рисунков общего вида.

Первоначально эта проблема была отодвинута более срочными задачами, однако сейчас представляется возможным взглянуть на нее с позиций опыта, приобретенного за период разработки СМОГ.

В этой статье мы рассмотрим только те три вида кодировок, которые остались в поле нашего зрения после проведенного за эти годы отбора.

I.1. Кодировка с переменной длиной элемента. Каждый элемент задает одно движение с помощью параметров ТРА (x , y , J).

I.2. Контекстно зависимая кодировка с фиксированной длиной элемента.

I.3. Кодировка с подрисунками и переменной длиной элемента.

Какие же требования можно предъявить к кодировке? Очевидны два: минимальная память для МКС и минимальное время работы декодера. Кроме того кодировка должна удовлетворять принятым стандартам на изображение символов и некоторым эстетическим требованиям. Важным достоинством кодировки может являться легкость включения кодировок новых символов и изменения кодировок старых символов. Имея в виду эти основные требования, приступим к описанию кодировок I.I-I.3.

Во всех рассматриваемых случаях МКС состоит из последовательности элементов, каждый из которых содержит некоторую информацию для декодера. Чаще всего элемент задает одно движение пера по отрезку прямой, однако в кодировках имеются также элементы с некоторой служебной информацией для декодера.

Такие элементы используются как приказы-повторители, признаки поворота траектории движения пера, признаки смены состояния пера, указатели подрисунков и так далее. В кодировках может использоваться также один специальный элемент-разделитель, который служит признаком начала кодировки некоторого символа или рисунка.

Для кодирования движений пера используется некоторая условная координатная сетка (УКС). Декодер должен преобразовать условные координаты (УК) в листовые и произвести необходимые преобразования координат для реализации поворота и наклона символа к осям листовой системы. После этого нужно обратиться к ТРА для отрисовки отрезка. Рассмотрим два способа кодирования перемещений пера по УКС: в координатах (от некоторой начальной точки) и в приращениях (от каждой очередной точки). Так как для второго способа в большинстве случаев требуется меньшая длина элемента кодировки, то этот способ и был реали-

зован во всех кодировках I.I – I.3.

Кодировка I.I. Подробно кодировка описывается в работе [2].

Здесь хотелось бы отметить следующие ее особенности.

I.I.1. Простота: отсутствие сложных элементов, каждый элемент кодирует одно движение пера или один приказ на смену состояния пера.

I.I.2. Универсальность: незначительная разница между длинами элементов, кодирующих большое и малое перемещение пера. Соответственно, разница в памяти для кодирования различных типов рисунков (с одинаковым числом, но не длиной отрезков) также незначительна.

I.I.3. Малая эффективность при наличии часто повторяющихся частей рисунка. Ясно, что кодировка, позволяющая записывать эти части только один раз, будет занимать меньше памяти, чем I.I.

I.I.4. Большая длина кодировки, чем I.2 в том случае, когда большинство движений производится между соседними узлами УКС ("на один шаг").

Кодировка I.2. При разработке этого типа кодировки основное внимание уделялось вопросу экономии памяти, так как необходимо было обеспечить идентичность кодировки вариантов для М-220 и БЭСМ-6. В связи с этим основным требованием решено было выбрать минимальный размер элемента кодировки. Ограничились только восемью видами движений, кодируемых одним

элементом (рис. I). Каждое такое "элементарное" движение соединяет два соседних узла УКС, то есть такие узлы, координаты которых различаются не более чем на одну условную единицу (у.е.) по обеим осям. Для кодировки восьми элементарных движений необходим как минимум трехразряд-

Рис. I

ный элемент ($2^3=8$). Однако нельзя обойтись без некоторых служебных элементов (разделителей, повторителей и так далее).



Поэтому для кодировки I.2 был принят четырехразрядный элемент. Ясно, что это обстоятельство существенно ограничивает возможности кодировки при ее использовании в целях записи "больших" движений. Сейчас мы склоняемся в пользу кодировки типа I.3. Но об этом позже.

Несмотря на свою узкую специализацию, кодировка I.2 представляет и сейчас определенный интерес как наиболее оптимальная по памяти из I.1 – I.3 для целей кодирования символов. В кодировках более перспективного типа хотелось бы использовать положительные стороны I.2, по возможности избавившись от ее недостатков.

I.2.1. Описание кодировки I.2. МКС кодировки I.2 состоит из четырехразрядных элементов (тетрад), упакованных по II (в варианте M1) или I2 (в варианте B1) в ячейку. После выбора размера элемента сразу встал вопрос о методе поиска кодировки нужного символа в МКС. В первом варианте кодировки I.2 последовательность тетрад, кодирующую рисунок символа (ПТС), пытались поместить в одну ячейку, что сразу упростило алгоритм поиска нужной ПТС. Однако плохие изобразительные возможности такого способа кодирования заставили нас от него отказаться. В окончательном варианте ПТС следуют подряд, одна за другой. При этом начальный элемент ПТС (тетрада-разделитель) не обязательно находится в начале ячейки. Здесь проблема поиска нужной ПТС усложняется. Нами были рассмотрены два варианта.

I.2.1.1. Поиск по каталогу. Составляется каталог, в котором хранятся адреса первых тетрад каждого ПТС. Затем при поиске нужной ПТС по номеру символа выбирается из каталога адрес первой тетрады кодировки этого символа.

Способ оптимален по времени поиска, однако требует значительную память для каталога (правда, не нужны разделители между ПТС соседних символов).

I.2.1.2. Поиск по счетчику. ПТС всех символов отделены друг от друга тетрадой-разделителем. Поиск нужной ПТС осуществляется проходом от начала кодировки до первой тетрады нужного символа. При этом подсчитываются разделители. Как только встретится разделитель, номер которого совпадает с номером нужного символа, ПТС будет найдена.

Этот способ не оптимален по времени, но требует меньше памяти для целей поиска, поэтому он и был принят для кодировки I.2. Разделителем является тетрада IIII. Нумерация символов соответствует кодировке устройства КУ-3 (всего 256 символов). ПТС для незадействованных символов состоит из одного разделителя.

Опишем структуру ПТС для некоторого символа. Первой тетрадой ПТС всегда является разделитель IIII. Вторая тетрада определяет растяжение символа по осям x и y. Первые два разряда, обозначим их k_x , определяют коэффициент растяжения по x (k_{0x}), вторые два разряда, обозначим их k_y , - по y (k_{0y}). Зависимость k_{0x} от k_x и k_{0y} от k_y определяется по таблицам:

k_x	k_{0x}	k_y	k_{0y}
00	MAX/5	00	MAU/7
01	3/5	01	5/7
10	4/5	10	I
II	I	II	6/7

Комбинация $k_x = II$ и $k_y = II$ запрещена, поскольку в этом случае масштабная тетрада будет совпадать с разделителем. Принятый способ масштабирования, к сожалению, требует второго прохода по ПТС. При первом (холостом) проходе определяются максимальные размеры символа (в условных единицах системы УКС) MAX и MAU, а при втором (рабочем) проходе они используются в формулах пересчета условных координат в листовые. Формулы пересчета имеют следующий вид:

$$x = k_{px} \times k_{0x} / MAX \times px,$$

$$y = k_{py} \times k_{0y} / MAU \times py.$$

Здесь k_{px} , k_{py} - условные координаты, а px , py - требуемые размеры символа в миллиметрах.

Для устранения второго прохода пришлось бы ввести еще две масштабные тетрады для MAX и MAU, что вступило бы в противоречие с принципом оптимальности по памяти для кодировки I.2.

Третья и четвертая тетрады ПТС суть соответственно условные координаты x и y точки, от которой начинает рисоваться

символ. (Считая от той точки, в которой находилось перо в момент обращения к СИМ).

Следующие тетрады ПТС могут быть указателями направления движения пера, признаками изменения состояния пера, повторителями или признаками поворота траектории пера ("графами").

Тетрада-указатель направления имеет следующий вид:

номер разряда	4р	3р	2р	1р
признак движения	-у	+у	-х	+х

Более наглядно, нумерация направлений такова:

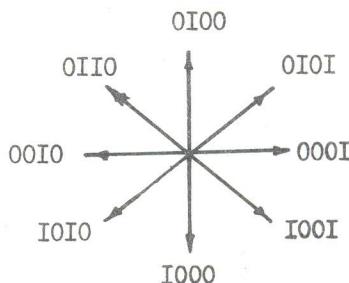


Рис.2

Тетрады II00, II01, III0, 00II, 0III, 10II, IIII не имеют смысла как указатели направлений. (В них указаны одновременно признаки +х и -х или +у и -у). Тетрада 0000 указывает, что перо должно остаться в той же точке. Естественно использовать ее для изменения состояния пера (поднять/опустить). Начальное положение пера – опущено. Тетрада 0000 изменяет положение пера: если было опущено – поднимает, и наоборот.

Тетрада IIII используется как разделитель. Тетрады 10II, II00, II01, III0 используются как признаки повторения предыдущего направления, первая – один раз, вторая – два, третья – три, четвертая – четыре раза.

Тетрады 00II и 0III используются как признаки поворота траектории движения пера ("графы"). Поворот траектории на

45° вправо (граф 00II) или влево (граф 0III) производится после отработки каждого очередного повторителя направления. Любой граф в ПТС должен представлять из себя следующую последовательность тетрад:

... НГПП ... ПН ...

Здесь Н - тетрада-направление, Г - тетрада-признак графа, П - тетрада-повторитель.

Первая тетрада типа Н задает начальное направление (которое отрабатывается). Тетрада Г задает признак левого или правого графа. Признаком конца графа служит любая тетрада-направления. Тело графа - несколько тетрад-повторителей. Первый, третий, пятый и так далее повторители после тетрады Г означают количество поворотов траектории движения пера. После каждого такого поворота производится элементарное движение по новому направлению. Количество шагов по нему определяется соответственно вторым, четвертым, шестым и так далее повторителями после признака начала графа.

Пример 1.

... 0I0I 00II II0I III0 III0 III0 ...



Рис.3

Пример 2.

... 000I 0III II00 II00 III0 III0 ...

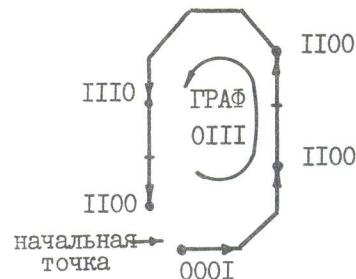


Рис. 4

I.2.2. Декодер I.2. Работа декодера в общих чертах ясна из описания кодировки, поэтому здесь мы ограничимся примером работы декодера для одного из символов кодировки. Для этого возьмем ПТС буквы α .

Обращение к СИМ: СИМ (103, 35, 50, 0, I.57);
 ПТС α : IIII I000 OIII OIOI IOIO III0 00II
 II0I IOII II0I IOOI II00 000I
 масштабная тетрада: I000.

$$k_x = 10 \quad k_{0x} = 4/5 \\ k_y = 00 \quad k_{0y} = M_{Ay}$$

точка начального подвода: (7, 5)

Рисунок символа на УКС:

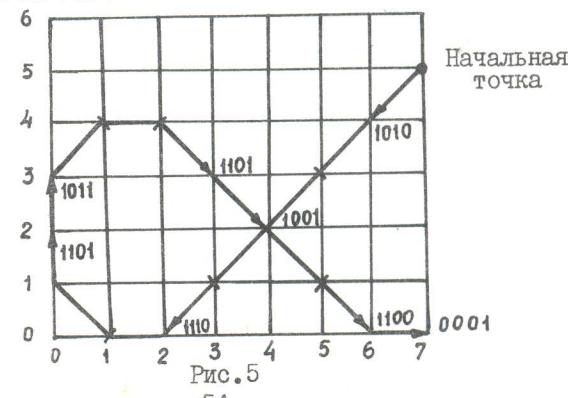


Рис. 5

После холостого прохода: $MAx = 7$, $MAy = 5$.

Отсюда $kOy = 5/7$.

После рабочего прохода (на миллиметровой сетке):

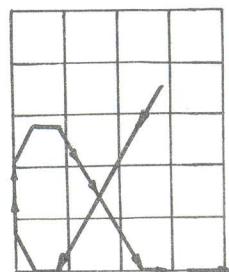


Рис.6

Здесь

$$I.y.e. \text{ по } x = kOx / MAx_{px} = 4/5/7 \times 35 = 4 \text{ мм}$$

$$I.y.e. \text{ по } y = kOy / MAy_{py} = 5/7/5 \times 50 = 7.1 \text{ мм}$$

стрелки на рис.6 показаны для лучшего соответствия с рис.5. Уголками показаны границы литер при данных px и py . После рабочего прохода делается подвод пера к началу следующего символа, в точку $(3/2 MAx, 0)$ условных координат (что удобно для отрисовки текстов).

I.2.3. Память для кодировки и декодера I.2. Кодировка занимает 370_8 ячеек, декодер – 560_8 ячеек (в вариантах Б1 и М1 почти одинаково). Как видно, декодер довольно сложен (он больше подпрограммы ТРА!). Общий объем памяти 1150_8 ячеек, из них почти 60% занимает декодер.

Кодировка I.3. В основе этой кодировки лежит кодировка I.1, описанная в работе [2]. Прочитать ее описание необходимо до чтения этого пункта. Здесь мы остановимся на тех добавлениях, которые нужно сделать, чтобы эффективно кодировать рисунки символов.

Элемент кодировки I.3 также имеет переменную длину. При сравнении с кодировкой I.2 сразу заметен "изъян" – минимальный размер элемента в кодировке I.3 восемь разрядов – вдвое больше, чем в I.2. Вероятно, что для задачи кодировки

символов, где достаточно часто встречаются перемещения на одну у.е., мы будем терять много памяти при переходе от I.2 к I.3. Для компенсации этого недостатка в I.3 вводится новый вид элемента: элемент-указатель подрисунка. Использование подрисунков позволяет кодировать часто встречающиеся части рисунков-символов только один раз. Это дает большую экономию памяти по сравнению с I.1, но все равно не доводит нас до уровня I.2 в вопросе оптимальности по памяти.

Важным достоинством I.3 является возможность более красиво закодировать символы, увеличив их размер на УКС, так как кодировка I.3 почти не увеличивает ПТС при увеличении рисунка. При этом можно отказаться от масштабной тетрады и тетрад-указателей начальной точки рисунка.

Еще более важно то, что можно от способа поиска ПТС по счетчику перейти к способу поиска по каталогу. В кодировке I.2 при общем объеме $370_8 = 248_{10}$ ячеек на разделители уходит 4×256 разрядов = 22 ячейки. Если под каталог и кодировку I.3 для 512 рисунков и подрисунков отвести 1024_{10} ячейки, то адрес некоторой тетрады будет записываться в I4 разрядов (I0 разрядов на номер ячейки +4 разряда на номер тетрады в ячейке), и в ячейку можно поместить три элемента каталога. Такой каталог будет занимать $512/3=173_{10}$ ячейки. (В четыре раза больше, чем в I.2 занимали бы разделители при 512 символах в кодировке). В целом кодировка будет примерно в два раза больше, чем I.2 для 512 символов.

Несмотря на это, кодировка I.3, по нашему мнению, заметно лучше. Во-первых, существенно упрощается декодер. Во-вторых, намного уменьшается время поиска ПТС. В-третьих, появляется возможность подключения личных библиотек рисунков (подрисунков), упрощается включение новых и изменение старых рисунков.

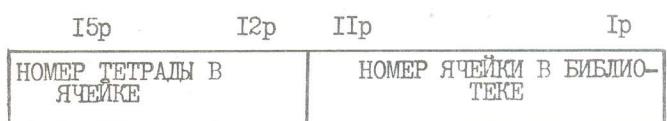
I.3.1. Описание кодировки I.3. Итак, в нашем распоряжении $2000_8 = 1024_{10}$ ячейки под кодировку (I лист БЭСМ-6). На рисунке 7 показан предлагаемый вариант размещения информации в листе памяти. В нулевой ячейке хранится имя библиотеки (6 символов в коде УПШ). Для общей библиотеки это СИСБМБ. В ячейках I- 257_8 хранится каталог библиотеки на 512 рисунков (термин подрисунки мы в дальнейшем, где это не

является существенным, будем опускать). В ячейках каталога упаковано по три элемента каталога (будем называть их ссылками). Каждая ссылка занимает 15 разрядов, поэтому как в БЭСМ-6, так и в М-220 в ячейку помещается три ссылки (в БЭСМ-6 разряды 48-46 не используются).

Структура ссылки для некоторого рисунка:



Рис. 7



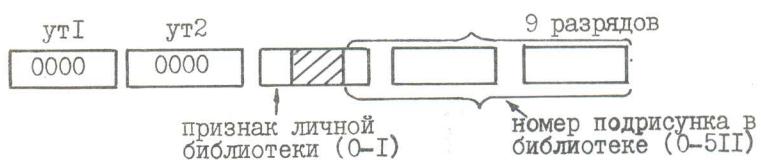
Под номер ячейки в библиотеке отведено II разрядов. Нумерация ячеек начинается с нуля. Общая библиотека занимает не более 1520_8 ячеек, личная может занимать до 4000_8 ячеек. Тетрады в ячейке нумеруются от 0 до 13_8 в БЭСМ-6 или от 0 до 12_8 в М-220. Ссылка для незадействованных символов (рисунков) содержит единицы во всех пятнадцати разрядах. В собственно библиотеке одна за другой, без разделителей, расположены ПТС для всех рисунков (в порядке возрастания номеров символов!). Конец ПТС определяется по каталогу: имеется первая следующая задействованная ссылка, после ссылки на раскодируемый рисунок. Таковая всегда будет найдена, если последняя задействованная ссылка в каталоге будет указывать на первую свободную тетраду библиотеки. Эта тетрада должна быть равна 0000, а следующая за ней — IIII. Такая комбинация кодирует приказ "Конец Графика" (см. [2]).

В описании кодировки I.2 (пункт I.2.I) мы не остановились на вопросе технологии кодирования МКС. В I.2 это делалось только один раз — для системной библиотеки рисунков. Для I.3 технология кодирования библиотек находится на более высоком

уровне. Опишем схему кодирования для I.3 в общих чертах.

Для кодирования запускается специальная задача, использующая один из вариантов СП СМОГ. На уровне подпрограммы СИМ происходит разрыв цепи входной язык \rightarrow СП СМОГ \rightarrow Графическое Устройство (в [2] такой разрыв может происходить на уровне подпрограммы БУФЕР). Полученные коды движений преобразуются в ПТС и записываются в библиотеку, при этом формируется соответствующая ссылка в каталоге библиотеки. После формирования библиотека может быть записана на внешний носитель (МЛ или МД). Добавление новых символов в библиотеку легко укладывается в эту схему.

По специальным обращениям к подпрограмме СИМ (вид которых мы здесь не уточняем) в ПТС вместо обычного элемента кодировки I.I могут быть записаны элементы-указатели подрисунков. Каждый такой элемент состоит из пяти тетрад:

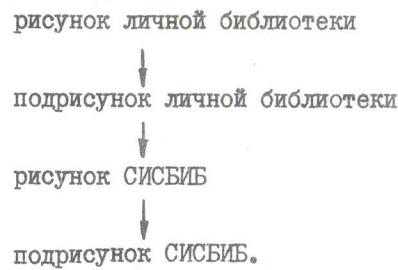


Здесь ут1, ут2 – управляющие тетрады (см. [2]). Левый разряд третьей тетрады – признак личной библиотеки ПРЛБ (I – личная библиотека, 0 – СИСБИБ). Средние 2 разряда третьей тетрады не используются. Правый разряд тетрады + четвертая + пятая тетрады составляют номер подпрограммы, от 0 до 5II, в личной или системной библиотеке.

В каждый момент времени в оперативной памяти находятся две библиотеки: СИСБИБ и одна из личных библиотек. Поэтому элементы-указатели в СИСБИБ должны иметь ПРЛБ = 0 (так как в памяти может находиться любая личная библиотека). При этом в любой личной библиотеке указатели могут ссылаться на рисунки как из системной библиотеки, так и из этой личной библиотеки. Назовем такие рисунки, на которые имеются указатели (в этой библиотеке), подрисунками. Принципиального различия между рисунками и подрисунками во время кодирования би-

лиотеки не делается. Соблюдается лишь одно ограничение: в подрисунках не должно быть элементов указателей (чтобы не появилась рекурсия). Это ограничение вызывает распечатку об ошибке при попытке раскодировать такой подрисунок. В личных библиотеках, однако, можно использовать указатели на рисунки из СИСБИБ, имеющие подрисунки (рекурсия при этом возникнуть не может). Обращения к декодеру I.3 для рисунков и подрисунков одинаковы, так как подрисунки и рисунки могут быть расположены в библиотеках вперемешку, имея общую нумерацию.

Из вышеизложенного ясно, что максимально длинная цепь указателей имеет следующий вид:



I.3.2. Д е к о д е р I.3. Задача декодера I.3 – найти по каталогу личной или системной библиотеки адрес первой тетрады ПТС нужного рисунка. Затем, просматривая одну за другой тетрады ПТС, выделяем элементы кодировки. Если это элемент–движение пера, то после перевода из условных координат в листовые, масштабирования, реализации поворота и наклона производится отрисовка через подпрограмму ТРА.

Если это элемент–указатель, то декодер временно переключается на выборку элементов подрисунка. Глубина вложенности подрисунков не более трех.

Интересно, что после того, как ТРА обратится к БУФЕРу для реализации отрисовки, движения пера будут переведены в элементы кодировки I.I. Фактически, декодер убирает из кодировки I.3 элементы–указатели, заменяя их на последовательности элементов I.I.

На рисунках I.8 показаны возможные схемы работ по кодированию и отрисовке символов. На этом рисунке 0, 1, 2, 3 и 4 – входы в схему 5, 6, 7 – разветвления схемы. К – конец работы.

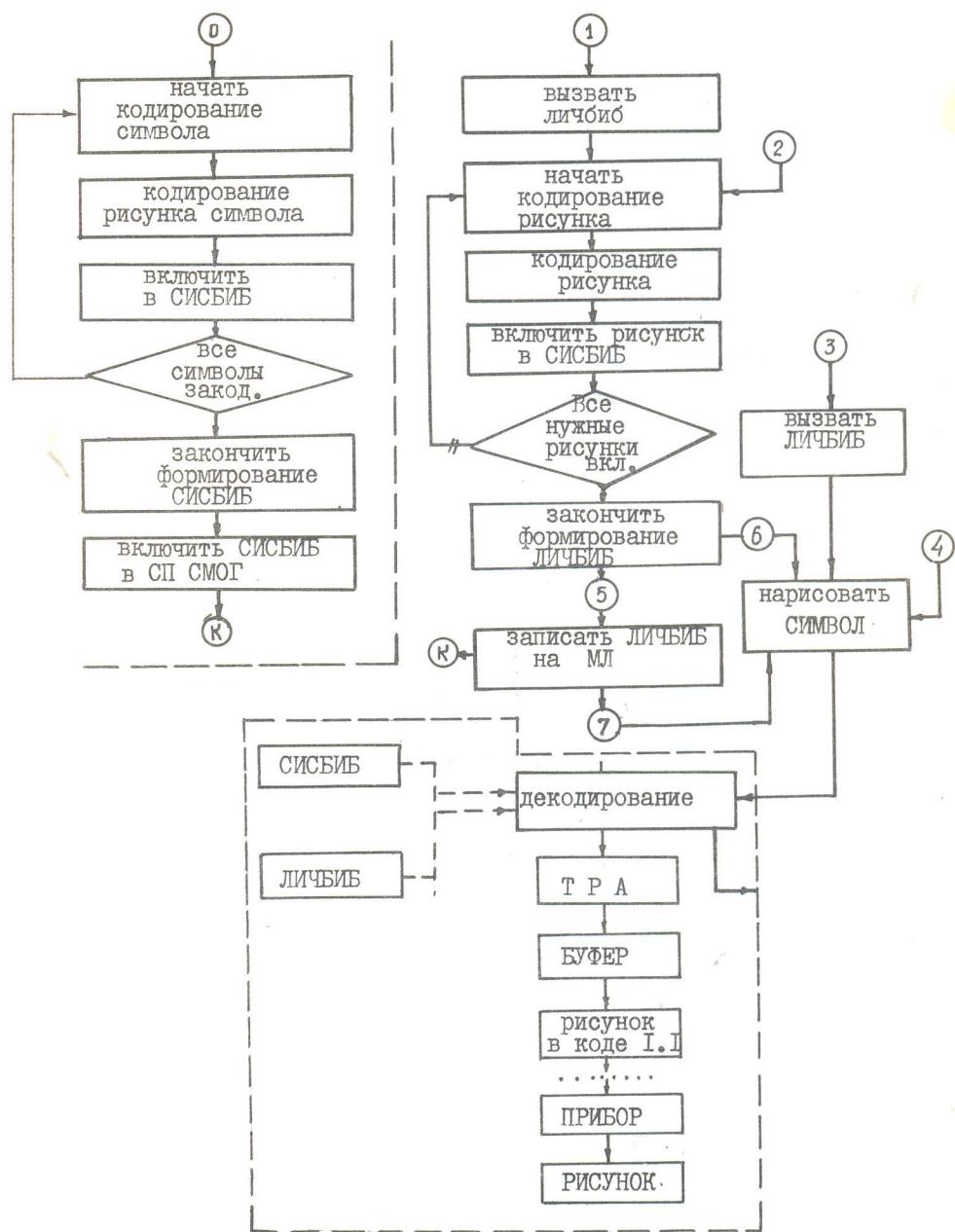
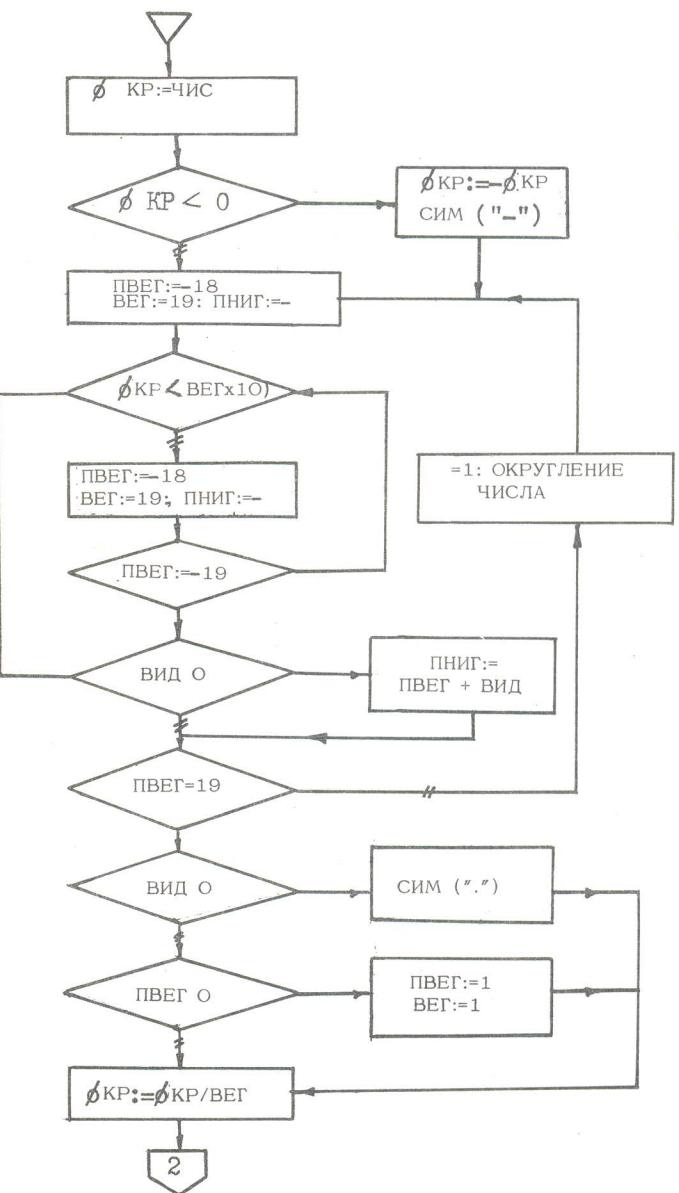
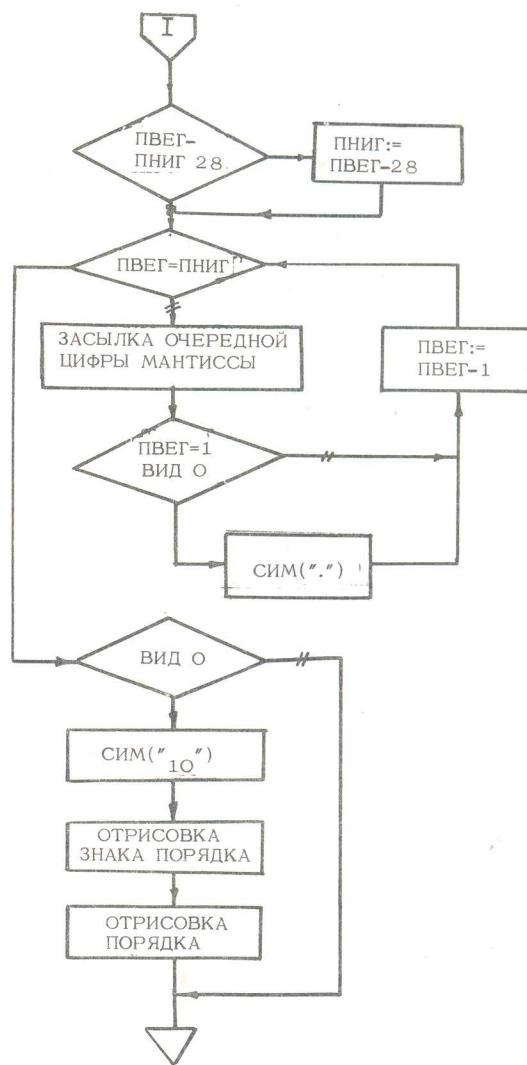


Рис. I.8

БЛОК-СХЕМА ПОДПРОГРАММЫ ЧИСЛО





В настоящее время проводится внедрение в СМОГ кодировки I.3. Ясно, что это потребует расширения входного языка СМОГ.

2. Генерация чисел

Эту работу выполняет подпрограмма ЧИСЛО. Напомним, что ее первые два параметра называются ЧИС и ВИД. ЧИС – нормализованное число, занимающее одну ячейку БЭСМ-6 или М-220. ВИД – параметр, указывающий, сколько цифр мантиссы (при $\text{ВИД} \geq 0$) или дробной части числа (при $\text{ВИД} < 0$) должно быть отрисовано.

В первых реализациях СП СМОГ работала довольно сложная и объемная версия этой подпрограммы, однако затем был разработан более простой и наглядный алгоритм, который и будет описан ниже. Этот алгоритм может найти применение и в областях, далеких от машинной графики, поскольку задача перевода двоичного числа в форматный десятичный вид встречается очень часто. Он уже был использован в одной из программ печати на АЛПУ.

Приступим к описанию алгоритма. Рекомендуем параллельно с чтением этого описания следить за изложением по блок-схеме (см.ниже).

Назовем порядком верхней границы заданного числа ЧИС такое целое число ПВЕГ, что

$$10^{\text{ПВЕГ}-1} \leq \text{abs}(\text{ЧИС}) \leq 10^{\text{ПВЕГ}}. \quad (1)$$

Пусть ОКР – округленное значение ЧИС, У – признак "Округление сделано", а ВЕГ = $10^{\text{ПВЕГ}-1}$.

Сначала ОКР := abs(ЧИС) и, если ЧИС < 0, рисуем знак "-" обращением к подпрограмме СИМ. Далее, начинаем поиск верхней границы числа. Засыпаем в ВЕГ число 10^{-19} , а в ПВЕГ – число -18. Затем умножим ВЕГ на 10 столько раз, сколько нужно, чтобы выполнялось неравенство (1). При каждом умножении ВЕГ на 10 к ПВЕГ прибавляется единица.

Определим целое число ПНИГ (порядок нижней границы) следующим образом:

$$\text{ПНИГ} := \underline{\text{ЕСЛИ}} \text{ ВИД} < 0 \text{ } \underline{\text{ТО}} \text{ ПВЕГ} + \text{ВИД} \text{ } \underline{\text{ИНАЧЕ}} - \text{ВИД}$$

Покажем, что количество рисуемых цифр числа равно ПВЕГ – ПНИГ. Действительно, при $\text{ВИД} < 0$ количество выводимых цифр

равно – ВИД, а при $\text{ВИД} \geq 0$ равно ПВЕГ + ВИД, так как при $\text{ОКР} \geq I (\text{ПВЕГ} \geq I)$ число цифр до десятичной точки, равно ПВЕГ, а после – равно ВИД. При $\text{ОКР} < I$ мы зашлем в ВЕГ и ПВЕГ число I (см. блок-схему), чтобы отрисовать значащие нули перед числом (для чисел типа 0.000542).

Округление производится обычным способом – можно показать, что для этого к ОКР достаточно прибавить число $0.5 \times 10^{\text{ПНИГ}}$.

После округления снова перевычисляем ВЕГ, ПВЕГ и ПНИГ (в предельном случае они могли измениться). Если $\text{ВИД} < 0$, то перед числом рисуется "..". Иначе проверяем неравенство $\text{ПВЕГ} \leq 0$, что эквивалентно $\text{ОКР} \geq I$. Если оно выполняется, то выполняем оператор $\text{ПВЕГ:} = \text{ВЕГ:} = I$.

Далее, ОКР переводится в диапазон

$$I \leq \text{ОКР} \leq 10$$

делением на ВЕГ. Затем проверяем неравенство $\text{ПВЕГ} - \text{ПНИГ} \geq 28$. Если оно выполняется, то $\text{ПНИГ:} = \text{ПВЕГ} - 28$. Это сделано для того, чтобы исключить ошибки пользователя, связанные с заданием слишком большого ВИДа. Число 28 выбрано потому, что до десятичной точки в машинах М-220 и БЭСМ-6 может быть не более 19-и цифр, а после десятичной точки имеют смысл не более 10 цифр.

Затем крушим цикл отрисовки цифр числа ОКР (слева направо). При каждом проходе цикла выделяем очередную цифру числа оператором $ц := \text{ENTIER}(\text{ОКР})$ и отбрасываем ее, умножая остаток на 10 [выполняем оператор $\text{ОКР:} = (\text{ОКР} - \text{ENTIER}(\text{ОКР})) \times 10$]. Потом проверяем равенство $\text{ПВЕГ} = I$ при $\text{ВИД} > 0$. Если оно выполняется, то необходимо вывести ".." (внутри числа, в форме с фиксированной запятой).

Действительно, если $\text{ПВЕГ} = I$, то из условия $I \leq \text{ОКР} \leq 10$ следует

$$10^{\text{ПВЕГ}-I} \leq \text{ОКР} \leq 10^{\text{ПВЕГ}},$$

то есть ОКР в этот момент имеет вид:

ц.ццц...ц

и после отрисованной цифры нужно вывести "..". Для отрисовки следующей цифры $\text{ПВЕГ:} = \text{ПВЕГ} - I$ и проверяется равенство $\text{ПВЕГ} = \text{ПНИГ}$. Если оно выполняется, выданы уже все цифры числа.

По окончании цикла отрисовки цифр, рисуется порядок числа (при ВИД < 0) и работа подпрограммы заканчивается.

Пример. Пусть ЧИС = 248.192, ВИД = +1.

ВЕГ = 100, ПВЕГ = 3, ПНИГ = -ВИД = -1.

OKP = ЧИС + 0.5 x 10^{-1} = 248. 242.

После приведения числа в диапазон 1 ÷ 10, OKP = 2.48242.

1. При отрисовке первой цифры ПВЕГ = 3, ENTIER (OKP)=2.

OKP: = (OKP - ENTIER (OKP)) x 10 = 4.8242.

2. Для второй цифры ПВЕГ = 2, ENTIER (OKP) = 4.

OKP: = (OKP - ENTIER (OKP)) x 10 = 8.242.

3. Для третьей цифры ПВЕГ = 1, ENTIER (OKP) = 8.

OKP: = (OKP - ENTIER (OKP)) x 10 = 2.42.

Поскольку ПВЕГ = 1 и ВИД > 0, то отрисовывается ".".

4. Для четвертой цифры ПВЕГ = 0, ENTIER (OKP) = 2.

OKP: = (OKP - ENTIER (OKP)) x 10 = 4.2.

5. ПВЕГ = -1 = ПНИГ. Цикл отрисовки окончен.

Л и т е р а т у р а

1. Куртуков А.Я. Математическое обеспечение для графопостроителей. I уровень.-ВЦ СО АН СССР, 1970, с.82.
2. Дебелов В.А. Буферная часть СМОГ.- Настоящий сборник.
3. Горин С.В., Дворжец В.И., Дебелов В.А., Куртуков А.Я. Структура СМОГ БЭСМ-6. - Настоящий сборник
4. Машинная графика и ее применение. Сборник статей под ред. Ю.А.Кузнецова. Новосибирск, ВЦ СО АН СССР, 1973.