

НИИ «НАУЧНЫЙ ЦЕНТР»

ПРОГРАММНОЕ
ОБЕСПЕЧЕНИЕ
ДВК

КНИГА 8

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДИСПЛЕЯ
НА БАЗЕ КЦГД

НИИ «НАУЧНЫЙ ЦЕНТР»

ПРОГРАММНОЕ
ОБЕСПЕЧЕНИЕ

ДВК

КНИГА 8

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДИСПЛЕЯ
НА БАЗЕ КЦГД

МОСКВА 1990 год



АННОТАЦИЯ

Данная книга представляет собой описание программного обеспечения контроллера цветного графического дисплея (КЦГД). Два варианта управляющей программы, размещенные в ПЗУ КР1801РЕ2-181 и КР1801РЕ2-182, реализуют функции дисплеев VT100 и VT200. Эти управляющие программы позволяют расширить свои функции с помощью дополнительного программного обеспечения, загружаемого из ЭВМ.

В первой главе изложены функции символьного дисплея. Описаны принципы кодирования входной информации, встроенные алфавиты, управляющие символы и командные последовательности, работа клавиатуры, установка цвета символов и фона. Описаны также дополнительные функции, обеспечиваемые загружаемой программой графического протокола KeyGP: дополнительные команды алфавитно-цифрового дисплея и работа с меню установки режимов. Во второй главе описаны команды и функции графического протокола KeyGP, выполняющие построение различных графических примитивов на экране дисплея. Третья глава книги — руководство программиста по использованию программы графического протокола. В ней описано, что входит в состав поставляемого комплекта программного обеспечения, как создать файл — загрузчик программы графического протокола на жестком диске, и как загрузить эту программу в КЦГД. Описана также программа GX, позволяющая пользователю непосредственно проверить исполнение команд графического протокола; и драйвер HC.SYS, выводящий копию экрана на печатающее устройство. В Приложении перечислены отличия ПЗУ КР1801РЕ2-182 от КР1801РЕ2-181.

Управляющие программы, размещенные в ПЗУ КР1801РЕ2-181 и КР1801РЕ2-182, разработаны В. А. Димовым, С. П. Савельевым и А. Г. Григорьевым. Программа графического протокола KeyGP разработана А. Г. Григорьевым, В. А. Димовым, В. М. Полетаевым, С. П. Савельевым при участии Ю. В. Панферова, К. С. Рыжонкова и А. В. Судакова. Исправление ошибок ПЗУ КР1801РЕ2-181 в программе графического протокола KeyGP выполнено В. М. Полетаевым.

ФУНКЦИИ СИМВОЛЬНОГО ДИСПЛЕЯ НА БАЗЕ КЦГД. РУКОВОДСТВО ОПЕРАТОРА

1. ВВЕДЕНИЕ

Контроллер цветного графического дисплея (КЦГД) представляет собой микро-ЭВМ на основе микропроцессора К1801ВМ2, связанную через порт ввода-вывода с шиной центрального процессора ПЭВМ Электроника МС0507.03—05, МС0504. Для центрального процессора КЦГД выглядит как стандартный системный терминал, имеющий регистры состояния и данных ввода и вывода. Процессор КЦГД принимает информацию, которую посылает центральный процессор, и согласно программе, записанной в ПЗУ, распознает управляющие и печатные символы и строит на экране изображение выводимых символов. Кроме того, процессор КЦГД принимает информацию с клавиатурного блока, преобразует принятые номера клавиш в стандартные коды символов и посылает эти символы центральному процессору. Вся работа КЦГД, его реакция на различные последовательности управляющих символов определяется программой, записанной в микросхеме КР1801РЕ2-181 (далее ПЗУ-181) или КР1801РЕ2-182 (далее ПЗУ-182). В дальнейшем под словом «дисплей» мы будем понимать КЦГД с управляющей программой ПЗУ-181 или ПЗУ-182.

В ПЗУ-182 по сравнению с ПЗУ-181 введены новые функции, а также исправлены имевшиеся ошибки. Загружаемая в КЦГД программа графического протокола KeyGP дополняет управляющую программу ПЗУ-182 возможностью эмуляции команд дисплея VT220, а вариант ее для ПЗУ-181 также исправляет некоторые ошибки, имеющиеся в управляющей программе. Отличия ПЗУ-182 от ПЗУ-181, а также ошибки ПЗУ-181, перечислены в ПРИЛОЖЕНИИ.

Управляющая программа позволяет заменить и/или расширить систему команд путем «подкачки» программного обес-

печения извне в ОЗУ КЦГД и подключения его к существующему в ПЗУ. За счет загруженного дополнительного программного обеспечения можно скорректировать систему команд дисплея, реализовать графические функции КЦГД, сменить кодировки и/или прорисовки символов, а также коды, передаваемые в ЭВМ при нажатии на клавиши клавиатуры. Для выполнения «подкачки» имеются специальные команды, описанные ниже. Правила написания дополнительного ПО в настоящем документе не рассматриваются.

Дисплей имеет два основных режима управления — режим совместимости с дисплеем VT52 и контроллером КСМ ЭВМ MC0507, далее называемый режимом совместимости, и режим ANSI (соответствует стандарту ANSI X3.64—1979), причем в режиме ANSI есть три уровня, соответствующих широко распространенным дисплеям VT100, режим VT200TM, ниже по тексту они называются режим VT100, режим VT200 7 бит и режим VT200 8 бит. Переход между этими режимами осуществляется по специальным последовательностям, которые будут описаны ниже.

Управляющая программа ПЗУ-181 не обеспечивает режим VT200.

ВНИМАНИЕ! Если на дисплее в течение 25 минут не была нажата ни одна клавиша или дисплей не принял ни одного символа из ЭВМ, то гасится изображение на экране для увеличения ресурса ЭЛТ монитора. Для его восстановления необходимо нажать любую клавишу, например, СУ.

Примечание. Управляющая программа, представленная содержимым ПЗУ КР1801РЕ2-182, а также данный документ, являются собственностью НИИ «Научный Центр» и разработчиков, и защищены знаком «Copyright», что запрещает тиражирование программы и данного документа без согласия НИИ НЦ или разработчиков.

2. КОДИРОВАНИЕ ВХОДНОЙ ИНФОРМАЦИИ

Дисплей и ЭВМ обмениваются информацией в 7- и 8-битных кодах (от 0 до 377). Здесь и далее коды приводятся в восьмеричной системе счисления. В зависимости от кода символ обрабатывается как управляющий или печатный. Символы с кодами от 0 до 37 являются управляющими символами

TM Названия серий дисплеев VT52, VT100, VT200 являются зарегистрированными торговыми знаками фирмы Digital Equipment Corp., США.

ми таблицы С0, хотя не все из них имеют свою функцию. Символы с кодами от 200 до 237 принадлежат к таблице управляющих символов С1. Они обрабатываются только в режиме VT200 8 бит. Управляющие символы служат для задания элементарных редакторских функций дисплея. Символы с кодами от 40 до 177 и от 240 до 377 являются печатными и выводятся на экран в виде соответствующей буквы, цифры или специального знака. Соответствие между кодом и выводимым символом зависит от текущего назначения алфавитов.

Из управляющих и печатных символов могут формироваться управляющие последовательности, служащие для задания сложных функций редактирования информации. Управляющие последовательности состоят из начального символа (или нескольких символов), которым обычно является ESC, затем идут один или более символов с кодами от 40 до 176. Печатные символы, входящие в состав последовательности, не отображаются на экране.

3. АЛФАВИТЫ

Дисплей имеет 4 встроенных алфавита — ASCII (стандарт ИСО 646—1977) — латинские буквы, цифры и специальные символы, нанесенные на клавиатуре, кириллица, дополнительные знаки европейских национальных алфавитов и псевдографика. Наборы ASCII и кириллица соответствуют ГОСТ 13052-74 (КОИ-7 Н0, Н1). В них символы имеют следующие коды:

Алфавит ASCII (КОИ-7 Н0)

040	пр	060	0	100	Q	120	P	140	'	160	р
041	!	061	1	101	A	121	Q	141	a	161	q
042	"	062	2	102	B	122	R	142	b	162	r
043	#	063	3	103	C	123	S	143	c	163	s
044	¤	064	4	104	D	124	T	144	d	164	t
045	¥	065	5	105	E	125	U	145	e	165	u
046	&	066	6	106	F	126	V	146	f	166	v
047	'	067	7	107	G	127	W	147	g	167	w
050	(070	8	110	H	130	X	150	h	170	x
051)	071	9	111	I	131	Y	151	i	171	y
052	*	072	:	112	J	132	Z	152	j	172	z
053	+	073	;	113	K	133	[153	k	173	{
054	,	074	<	114	L	134	\	154	l	174	
055	-	075	=	115	M	135]	155	m	175	}
056	.	076	>	116	N	136	^	156	n	176	~
057	/	077	?	117	O	137	_	157	o	177	■

пр — пробел

Алфавит КИРИЛЛИЦА (КОИ-7 Н1)

040	пр	060	0	100	Ю	120	П	140	Ю	160	П
041	!	061	1	101	Э	121	Я	141	А	161	Я
042	"	062	2	102	Б	122	Р	142	Б	162	Р
043	#	063	3	103	Ц	123	С	143	Ц	163	С
044	¤	064	4	104	Д	124	Т	144	Д	164	Т
045	¥	065	5	105	Е	125	У	145	Е	165	У
046	&	066	6	106	Ф	126	Ж	146	Ф	166	Ж
047	'	067	7	107	Ф	127	В	147	Г	167	В
050	(070	8	110	Х	130	Ь	150	Х	170	Ь
051)	071	9	111	И	131	Ы	151	И	171	Ы
052	*	072	:	112	Й	132	Э	152	Й	172	Ш
053	+	073	;	113	К	133	Ш	153	К	173	Ш
054	,	074	<	114	Л	134	Э	154	Л	174	Щ
055	-	075	=	115	М	135	Щ	155	М	175	Щ
056	.	076	>	116	Н	136	Ч	156	Н	176	Ч
057	/	077	?	117	О	137	Ь	157	О	177	Ь

пр — пробел

В алфавите КИРИЛЛИЦА символы с кодами 40-77 изображаются так же, как в ASCII.

ПСЕВДОГРАФИКА

040	пр	060	0	100	Q	120	P	140	◆	160	—
041	!	061	1	101	A	121	Q	141	■	161	—
042	"	062	2	102	B	122	R	142	■	162	—
043	#	063	3	103	C	123	S	143	■	163	—
044	№	064	4	104	D	124	T	144	■	164	—
045	%	065	5	105	E	125	U	145	■	165	—
046	&	066	6	106	F	126	V	146	■	166	—
047	'	067	7	107	G	127	W	147	■	167	—
050	(070	8	110	H	130	X	150	■	170	—
051)	071	9	111	I	131	Y	151	■	171	—
052	*	072	:	112	J	132	Z	152	■	172	—
053	+	073	;	113	K	133	[153	■	173	—
054	,	074	<	114	L	134	\	154	■	174	—
055	-	075	=	115	M	135]	155	■	175	—
056	.	076	>	116	N	136	^	156	■	176	—
057	/	077	?	117	O	137	пр	157	■	177	—

пр — пробел

Коды 40—136 изображаются так же, как и в алфавите ASCII. Символы с кодами от 157 до 163 могут использоваться для рисования горизонтальных линий на разных уровнях знакоместа. Коду 157 соответствует линия на уровне 1/10 (сверху), 160 — 3/10, 161 — 5/10, 162 — 7/10, 163 — 9/10. Для рисования сеток и таблиц с использованием символов «углов» и T-сопряжений следует применять горизонтальную линию на уровне 5/10 (код 161).

Дополнительные знаки европейских алфавитов

040	пр	060	°	100	À	120	?	140	à	160	?
041	í	061	±	101	Á	121	Ñ	141	á	161	ñ
042	¢	062	²	102	Â	122	Ò	142	â	162	ó
043	£	063	³	103	Ã	123	Ó	143	ã	163	ô
044	¥	064	¼	104	Ä	124	Ô	144	ä	164	õ
045	¥	065	½	105	Å	125	Õ	145	å	165	ö
046	¥	066	¾	106	Æ	126	Ö	146	æ	166	ø
047	§	067	·	107	Ç	127	È	147	ç	167	é
050	§	070	¸	110	È	130	Ù	150	è	170	ê
051	©	071	¹	111	É	131	Ú	151	é	171	ë
052	®	072	º	112	Ê	132	Û	152	ê	172	ü
053	€	073	»	113	Ë	133	Ü	153	ë	173	ÿ
054	¿	074	¼	114	Ì	134	Û	154	ì	174	ÿ
055	¿	075	½	115	Í	135	Ü	155	í	175	ÿ
056	¿	076	¾	116	Î	136	Ý	156	î	176	ÿ
057	¿	077	¿	117	Ï	137	ß	157	ï	177	■

4. СДВИГ ИНФОРМАЦИИ ПО ЭКРАНУ (РУЛОН)

Текст, находящийся на экране, может сдвигаться вверх или вниз в пределах всего экрана или некоторой его области. Такой сдвиг обычно называется рулоном или скроллингом. Сдвигаемая часть текста (область рулона) всегда имеет ширину экрана и может содержать от 2 до 24 строк. При сдвиге текста строки, выходящие за пределы области рулона, теряются, а в противоположной стороне области появляются пустые строки. Первоначально областью рулона является весь экран, однако в режиме ANSI может быть установлена произвольная область размером не менее двух строк.

Предусмотрен также специальный режим, в котором нумерация строк экрана идет относительно начала области рулона (режим относительной адресации), в отличие от обычного режима, когда строки нумеруются от начала экрана. В этом режиме курсор не может быть выведен из области рулона.

5. УПРАВЛЯЮЩИЕ СИМВОЛЫ КОДОВОЙ ТАБЛИЦЫ C0

Управляющие символы таблицы C0 обрабатываются одинаково как в режиме совместимости, так и в ANSI. Мнемоника управляющих символов дается в соответствии с ГОСТ 13052—74 и в скобках по стандарту ANSI X3.41—1974. Они выполняют следующие функции:

КТМ код 5

— (ENQ — enquiry) — дисплей посылает в ЭВМ запрограммированный ответ. После включения или сброса дисплея такого ответа нет, он вводится оператором в режиме меню, когда загружена программа графического протокола KeyGP.

ЗВ код 7

— (BEL — bell) — дисплей подает звуковой сигнал. Обычно этот символ используется для индикации ошибки в какой-либо прикладной программе, например, в редакторе текстов.

ВШ код 10

— (BS — backspace) — курсор передвигается на одну позицию назад, кроме случая, когда он находится на начале строки;

ГТ код 11

— (HT — horizontal tabulation) — горизонтальная табуляция. Курсор передвигается вправо на следующую позицию табуляции (табулятор). Обычно они располагаются через 8 символов, однако в режиме ANSI с помощью специальных команд можно установить табуляторы в произвольные позиции строки. Если следующего табулятора нет, курсор останавливается в последней позиции строки.

ПС код 12

— (LF — line feed) — перевод строки. Курсор передвигается на одну строку вниз, оставаясь в той же позиции строки. Если курсор находился в нижней строке установленной области рулона (обычно это нижняя строка экрана), то текст сдвигается вверх, появляется чистая строка, курсор остается на той же строке. Если установлен режим отработки ПС как «Новая строка», то одновременно курсор перемещается на начало строки. Режим новой строки включается в режиме ANSI при приеме последовательности ESC [20h, а выключается по ESC 20I].

ВТ код 13

— (VT — vertical tabulation) — вертикальная табуляция, обрабатывается аналогично ПС.

ПФ код 14

— (FF — form feed) — перевод формата, обрабатывается аналогично ПС.

ВК код 15

— (CR — carriage return) — возврат каретки. Курсор устанавливается на начало текущей строки.

РУС (П1) код 16

— (SO — shift out) — на таблицу GL назначается условный алфавит G1, обычно соответствующий кириллице.

ЛАТ (П0) код 17

— (SI — shift in) — на таблицу GL назначается условный алфавит G0, обычно соответствующий ASCII.

АН код 30

— (CAN — cancel) — отмена ввода любой управляющей последовательности (ESC—, CSI—, DCS—).

ЗМ код 32

— (SUB — substitute) — отмена ввода любой управляющей последовательности, на экран выводится символ ошибки — обращенный вопросительный знак.

АР2 код 33

— (ESC — escape) — символ, открывающий специальную управляющую последовательность (ESC — последовательность). За символом ESC должны следовать один или более символов, определяющих функцию дисплея. В режимах совместимости и ANSI дисплей воспринимает разные ESC-последовательности.

Прочие управляющие символы дисплей не обрабатывает и никак не реагирует на них. Загружаемые извне программы могут использовать эти символы по своему усмотрению.

6. УПРАВЛЯЮЩИЕ СИМВОЛЫ КОДОВОЙ ТАБЛИЦЫ C1

Управляющие символы, входящие в состав кодовой таблицы C1, имеют коды от 200 до 237 и обрабатываются только в режиме ANSI уровня VT200 8 бит. Если такой символ получен в ином режиме, то его старший бит (200) обнуляется, и он обрабатывается в соответствии с таблицей C0, то есть непредусмотренным образом. Все символы C1 имеют соответствующие эквиваленты в виде ESC-последовательностей,

состоящие из ESC и символа с кодом, на 100 меньшим кода соответствующего символа C1. Например, символ CSI с кодом 233 может быть представлен в виде последовательности ESC [(код 133)]. Полное их описание приведено в разделе «ESC-последовательности режима ANSI».

7. ESC-ПОСЛЕДОВАТЕЛЬНОСТИ

ESC-последовательности, как уже было сказано, состоят из начального символа ESC с кодом 33, и следующих за ним печатных символов с кодами от 40 до 176. Они, однако, не отображаются на экране. В режиме VT52 за ESC обычно следует один символ, определяющий функцию, а в режиме ANSI после ESC могут идти промежуточные символы и финальный символ. Промежуточные символы могут иметь коды от 40 до 57 (то есть от пробела до символа /), а финальный символ имеет код от 60 (цифра 0) до 176 (знак ~). Дисплей запоминает промежуточные символы до тех пор, пока не придет финальный, а затем выполняет заданную функцию.

Если во время приема последовательности приходят управляющие символы таблицы C0 или C1 (кроме ESC, CAN, SUB, CSI, DCS), то они незамедлительно обрабатываются, не влияя на принимаемую последовательность. Символы CAN и SUB прерывают текущую последовательность и дисплей снова переходит в обычный режим, когда печатные символы выводятся на экран. В случае приема SUB на экране выводится обратный вопросительный знак. По приему символов ESC, CSI, DCS начинается новая управляющая последовательность, а прежний ввод теряется.

8. РЕЖИМ ОТОБРАЖЕНИЯ УПРАВЛЯЮЩИХ СИМВОЛОВ

Загружаемая в КЦГД программа KeyGP дает возможность просмотреть поток управляющих символов и последовательностей, поступающий в дисплей. Режим отображения управляющих символов включается с помощью меню, вызываемого нажатием клавиш УСТ РЕЖИМА (поле находится в кадре УСТАНОВКА ДИСПЛЕЯ). В этом режиме дисплей вместо исполнения управляющих символов отображает их на экране в виде символов дополнительного цвета. Коду 1 соответствует буква А, коду 2 — буква В и так далее. Символ

ESC изображается в виде символа \$. Символы кодовой таблицы C1 изображаются подчеркнутыми буквами дополнительного цвета. Символы HT, CR, LF, FF, VT изображаются в виде псевдографических символов H_T , C_R , L_F , F_F , V_T дополнительного цвета, причем после приема LF, FF или VT происходит переход на новую строку. Печатные символы изображаются нормальным цветом.

Выйти из режима отображения можно также посредством меню.

9. РЕЖИМ СОВМЕСТИМОСТИ С VT52 И КСМ

После включения питания или выполнения полного сброса дисплей находится в режиме совместимости. В этом режиме обрабатываются следующие ESC-последовательности: ESC= (коды 33 75)

— дополнительная цифровая клавиатура переводится в специальный режим, в котором цифровые клавиши и клавиши [BOD], [.] , [—], [·], расположенные на ней, выдают специальные ESC-последовательности, что позволяет использовать их в качестве функциональных клавиш для некоторой прикладной программы, например, для редактора текстов. См. ниже описание работы клавиатуры.

ESC> (33 76)

— специальный режим дополнительной цифровой клавиатуры выключается.

ESC F (33 106)

— включается режим вывода псевдографики. После приема этой последовательности вместо символов с кодами 140—176 (малых латинских букв и знаков) дисплей будет отображать специальные псевдографические символы, позволяющие, в частности, рисовать на экране дисплея различные таблицы и диаграммы.

ESC G (33 107)

— режим вывода псевдографики выключается.

ESC Y <номер строки> <позиция в строке> (33 131 1+37 с+37)

— позиционирование курсора. Строки пронумерованы сверху вниз, начиная от 1, столбцы пронумерованы слева направо, также с единицы. Номера строки и позиции в строке, куда должен быть установлен курсор, задаются двумя символами, следующими за ESC Y.

Коды этих символов получают прибавлением к номеру строки или позиции в строке кода 37. Например, для указания верхней строки экрана или первой позиции строки необходимо послать код 40, то есть пробел, вторая строка задается символом с кодом 41 (восклицательный знак) и т. п. Если закодирован номер строки более 24. (30восьм) или номер столбца более 80 (120восьм), то курсор в соответствующем направлении не перемещается.

ESC H (33 110)

— установка курсора на начало экрана, то есть на начало его верхней строки.

ESC K (33 113)

— стирание до конца строки. Все символы справа от курсора в текущей строке, включая символ, на котором он находится, стираются, то есть заменяются пробелами.

ESC J (33 112)

— стирание до конца экрана. В текущей строке стираются все символы справа от курсора, включая символ, на котором он находится, а все строки ниже текущей стираются полностью. Чтобы полностью стереть экран, необходимо установить курсор на начало экрана, послать ESC H, затем послать ESC J.

ESC I (33 111)

— курсор вверх со сдвигом текста (рулоном). Курсор передвигается на одну строку вверх. Если курсор находился в верхней строке экрана, то текст на экране сдвигается вниз, нижняя строка теряется, а сверху появляется одна пробельная строка. Позиция курсора в строке не изменяется.

ESC A (33 101)

— курсор вверх. Курсор передвигается на одну строку вверх. Если курсор уже находится в верхней строке экрана, то он остается на месте.

ESC B (33 102)

— курсор вниз. Курсор передвигается на одну строку вниз. Если курсор уже находится в нижней строке экрана, то он остается на месте.

ESC C (33 103)

— курсор вправо. Курсор передвигается на одну позицию вправо. Если он уже находится в конце строки, то остается на месте.

ESC D (33 104)

— курсор влево. Курсор передвигается на одну позицию влево. Если он находится в начале строки, то остается на месте.

ESC V (33 126)

— сброс дисплея. Дисплей переходит в начальное состояние, как после включения питания или после нажатия кнопки УСТ на передней панели ЭВМ. Экран полностью стирается, вся память очищается, все загруженные программы сбрасываются. На программу, работающую в памяти центрального процессора, это никакого влияния не оказывает.

ESC Z (33 132)

— выдача автоответа, идентифицирующего дисплей. Дисплей посылает в ЭВМ символы ESC / Z (33 57 132).

ESC < (33 74)

— переход в режим ANSI. В режиме ANSI все перечисленные выше управляющие последовательности не обрабатываются дисплеем или обрабатываются иначе, чем описано. Так как после сброса дисплей находится в режиме совместимости, программы, использующие его в режиме ANSI, должны в начале работы посылать ESC <.

В режиме совместимости доступны три алфавита — русский, латинский и псевдографика. Коды 40—176 могут отображаться в виде любого из этих алфавитов. Для отображения латинских букв надо послать символ ЛАТ (код 17, другое название — SI), а для отображения русских букв надо послать символ РУС (код 16, другое название — SO). Независимо от этих символов, коды 300—377 всегда отображаются в виде русских букв. Когда дисплей принимает последовательность ESC F, то вместо малых латинских букв и знаков {, }, ~, |, — он начинает выводить специальные графические символы. Для выключения этого режима необходимо послать ESC G. На вывод русских букв это не влияет.

Для входа в режим совместимости из режима ANSI необходимо послать ESC [? 2 1 (последний символ — малая буква L). Если дисплей уже был в режиме совместимости, то на экране появится строка «?21».

10. РЕЖИМ ANSI

В режиме ANSI обеспечивается намного более широкий набор функций, нежели в режиме совместимости. Так, например, доступны 4 алфавита, причем с более развитой сис-

темой выбора, символы могут выводиться негативными, подчеркнутыми, мигающими, дополнительного цвета, строка может быть выведена символами двойной ширины, из двух строк может быть сделана строка с символами двойной высоты и т. д. Для перехода в режим ANSI из режима совместимости, в котором дисплей первоначально находился, необходимо послать ESC < (коды 33 74). Если дисплей уже в режиме ANSI, то эта посылка будет проигнорирована и ни на что не повлияет.

10.1. Управляющие последовательности режима ANSI

В режиме ANSI дисплей распознает три типа управляющих последовательностей, в соответствии с их начальными символами они называются ESC-, CSI- и DCS-последовательностями.

ESC-последовательность начинается с символа ESC (код 33), за которым может следовать до трех промежуточных символов (или ни одного) с кодами от 40 до 57, далее следует финальный символ с кодом от 60 до 176, только после его получения дисплей выполняет функцию, определяемую принятыми символами.

CSI-последовательность начинается с символа CSI, имеющего код 233, или эквивалентной ему последовательности ESC [(33 133). Символ CSI можно использовать только в режиме VT200 8 бит. Последовательность имеет такой формат:

CSI P1 ; P2 ; ... Pn ps fc

Здесь CSI — либо символ CSI (код 233), либо ESC [(коды 33 133);

P1; P2; ... Pn — строки десятичных цифр (коды от 60 до 71), каждая строка является числом, задающим значение параметра, например, число 12 представляется цифрами 1 и 2 с кодами 61 и 62. Параметры разделяются символами «;» (код 73). Функции CSI-последовательностей могут выполняться различным образом в зависимости от наличия модификатора параметров. Модификатором является символ «>» (код 76) или «?» (77), предшествующий хотя бы одному из параметров последовательности, при этом все остальные параметры также считаются модифицированными.

ps — промежуточный символ последовательности с кодом от 40 до 57 (от пробела до знака /). Он может отсутствовать.

fc — финальный символ последовательности с кодом от 100 до 176 (от @ до ~), он вместе с промежуточным символом задает требуемую функцию.

В одной CSI-последовательности может быть задано до

21 параметра. Для некоторых функций используются только 1—2 параметра, лишние игнорируются. Если внутри параметра встречается недопустимый символ, то все символы, следующие до знака «;» или до промежуточного или финального символа, игнорируются. Если параметр опущен, то в функциях используются определенные значения, обычно 0 или 1.

DCS-последовательность служит для посылки в дисплей большого объема информации для выполнения специальных управляющих функций, таких, как загрузка прорисовок символов, загрузка последовательностей для определяемых пользователем клавиш верхнего ряда клавиатуры, загрузка графической информации в виде так называемых сикселей (sixels), а также команды построения графических образов. Подробно формат DCS-последовательности будет описан ниже. В ПЗУ программы для функций, выполняющихся через DCS-последовательности, не занесены, поэтому вся информация, содержащаяся в них, игнорируется. Эти последовательности, однако, могут обрабатываться дополнительным программным обеспечением, загруженным в КЦГД.

DCS-последовательность имеет следующий формат:

DCS P1 ; P2 ; ... Pn ps fc данные ST

Здесь DCS — либо символ DCS (код 220), либо последовательность символов ESC P (коды 33 120);

P1; P2; ... Pn — десятичные числа, разделенные символами «;», являющиеся параметрами команды, подобно как в CSI-последовательности.

ps — промежуточный символ последовательности с кодом от 40 до 57 (от пробела до знака /). Он может отсутствовать;

fc — финальный символ последовательности с кодом от 100 до 176 (от @ до ~), вместе с промежуточным символом определяющий выполняемую функцию;

данные — текст, задающий информацию для выполняемой функции, например коды загружаемых прорисовок символов в специальной закодированной форме. Управляющие символы типа CR, LF, HT, которые могут встречаться в этом тексте, не обрабатываются обычным образом.

ST (string terminator) — либо символ ST (код 234), либо последовательность ESC \ (коды 33 134).

Данные, содержащиеся в DCS-последовательности, являются своего рода программой. Например, это могут быть команды графического протокола KeyGP, предписывающие построение различных графических изображений.

Выполнение DCS-последовательности прекращается при

приеме символов ST, CAN, SUB, ESC, CSI, DCS. При этом символы ST, CAN, SUB просто завершают ее, а ESC, CSI, DCS еще и начинают выполнение новой управляющей последовательности.

10.2. ESC-последовательности

Ниже приведены ESC-последовательности, которые выполняются только в режиме ANSI. В режиме ANSI уровня VT200 8 бит вместо некоторых последовательностей можно использовать их однобайтовые эквиваленты из таблицы C1. Их код получается прибавлением 100 к коду второго символа последовательности. Это допускается только для символов с кодами от 100 до 137, например, для ESC O (код 117) есть эквивалент 217, а для ESC п (код 156) его нет.

ESC D (33 104) или 204

— IND (index) — курсор вниз. Если курсор находился на нижней границе области рулона, то текст сдвигается вверх на одну строку, курсор остается на прежнем месте.

ESC E (33 105) или 205

— NEL (next line) — курсор на начало следующей строки. Отрабатывается как BK+ПС.

ESC H (33 110) или 210

— HTS (horizontal tab set) — установка табулятора в текущей позиции строки.

ESC M (33 115) или 215

— RI (reverse index) — курсор вверх. Если курсор находится в первой строке области рулона, выполняется сдвиг информации (рулон) вниз, сверху появляется чистая строка, а нижняя вытесняется и теряется.

ESC п (33 156)

— LS2 (lock shift G2) — на таблицу GL назначается алфавит G2 (постоянно).

ESC N (33 116) или 216

— SS2 (single shift G2) — переход в GL на алфавит G2 для одного символа.

ESC o (33 157)

— LS3 (lock shift G3) — на таблицу GL назначается алфавит G3.

ESC O (33 117) или 217

— SS3 (single shift G3) — переход на алфавит G3 для одного символа.

ESC ~ (33 176)

— LS1R (lock shift G1 right) — на таблицу GR назначается алфавит G1. Выполняется только в режиме VT200.

ESC } (33 175)

— LS2R (lock shift G2 right) — на таблицу GR назначается алфавит G2. Выполняется только в режиме VT200.

ESC | (33 174)

— LS3R (lock shift G3 right) — на таблицу GR назначается алфавит G3. Выполняется только в режиме VT200.

ESC с (33 143)

— RIS (reset to initial state) — сброс дисплея в исходное состояние, как при включении питания. Рекомендуется использовать частичный сброс, описанный ниже.

ESC 7 (33 67)

— сохранение состояния курсора. Сохраняются текущие назначения для GL, GR, G0, G1, G2, G3, переназначение алфавита для одного символа (single shift), текущие координаты курсора (номер строки и колонки), атрибуты выводимых символов, признак выборочного стирания, признак относительной адресации курсора, признак включения автоперевода на новую строку.

ESC 8 (33 70)

— восстановление состояния курсора, запомненного по ESC 7. Если ESC 7 не было, то курсор устанавливается в начало экрана, для алфавитов устанавливаются начальные значения, автоперевод включен, относительная адресация выключена, алфавит для вывода одного символа не назначен, все атрибуты символа, в том числе защита при выборочном стирании, выключены.

ESC Z (33 132) или 232

— запрос атрибутов дисплея. Дисплей посылает последовательность ESC [?1;2с (коды 33 133 77 61 73 62 143).

ESC= (коды 33 75)

— дополнительная цифровая клавиатура переводится в специальный режим, в котором цифровые клавиши и клавиши ВВОД, [], [—], [·], расположенные на ней, выдают специальные ESC-последовательности, что позволяет использовать их в качестве функциональных клавиш для некоторой прикладной программы, например, для редактора текстов.

ESC > (33 76)

— специальный режим дополнительной цифровой клавиатуры выключается.

Прочие ESC-последовательности и их частные случаи будут рассмотрены в соответствующих разделах.

10.3. Переключение алфавитов

Дисплей имеет 4 встроенных алфавита, приведенных вы-

ше — ASCII, кириллицу, дополнительные знаки европейских алфавитов, псевдографика. Кроме них, еще один алфавит может быть загружен в дисплей из прикладной программы (down-reloadable character set — DRCS) и еще один может быть определен из имеющихся прорисовок (national replacement character set — NRCS) в соответствии с некоторым национальным алфавитом. Эти функции выполняются дополнительным программным обеспечением, загружаемым в КЦГД.

Символы с кодами от 40 до 177 выводятся на дисплей в соответствии с кодовой таблицей GL (Graphics Left), а с кодами от 240 до 377 — в соответствии с таблицей GR (Graphics Right). Связь между конкретным алфавитом и таблицами GL и GR устанавливается с помощью двухуровневой системы ссылок. На таблицы GR и GL назначаются условные алфавиты, называемые G0, G1, G2, G3, каждому из которых может быть приписан конкретный алфавит. На GL могут быть постоянно назначены G0 или G1 и временно (только для одного символа) — G2 или G3. На GR могут быть постоянно назначены G1, G2, G3.

Первоначально установлены следующие соответствия:

GL <— G0
 GR <— G1
 G0 <— ASCII
 G1 <— кириллица
 G2 <— дополнительные знаки европейских алфавитов
 G3 <— псевдографика

Такие же соотношения устанавливаются при выполнении полного или частичного сброса дисплея.

Назначение реального алфавита на условный алфавит G0—G3 выполняется функциями под общим названием SCS (select character set). Это ESC-последовательности из символа ESC, за которым следует символ, определяющий алфавит (G0—G3), затем идут от одного до трех символов, определяющих имя реального алфавита. Имя встроенного алфавита всегда кодируется одним символом.

Условный алфавит задается символами:

G0 — «(» (50) — символ левой скобки;
 G1 — «)» (51) — символ правой скобки;
 G2 — «*» (52) — звездочка;
 G3 — «+» (53) — плюс.

Соответствующий ему реальный алфавит задается такими символами:

ASCII — «B» (102);
 Кириллица — «u» (165);

Дополнительные знаки — «<» (74);

Псевдографика — «0» (60).

Имя загруженного набора (DRCS) определяется при его загрузке, и оно же должно указываться в соответствующих SCS-командах.

Для удобства можно привести таблицу команд установки встроенных алфавитов:

	G0	G1	G2	G3
ASCII	ESC (B	ESC) B	ESC * B	ESC + B
Кириллица	ESC (u	ESC) u	ESC * u	ESC + u
Доп. знаки	ESC (<	ESC) <	ESC * <	ESC + <
Псевдографика	ESC (0	ESC) 0	ESC * 0	ESC + 0

Назначение на G2 и G3, а также переназначение алфавита дополнительных знаков на любой алфавит выполняется только в режиме VT200.

Соответствие между GR и GL с одной стороны, и G0, G1, G2, G3 с другой стороны, может быть установлено как постоянно (lock shift), так и только для одного символа (single shift). Ниже приведены управляющие символы и последовательности, с помощью которых выполняется такое назначение различных алфавитов на GL и GR.

	GL постоянно	GL временно	GR постоянно
G0	ЛАТ (17)	нет	нет
G1	РУС (16)	нет	ESC ~ (33 176)
G2	ESC n (33 156)	ESC N (33 116) или SS2 (216)	ESC } (33 175)
G3	ESC o (33 157)	ESC O (33 117) или SS3 (217)	ESC (33 174)

Следует отметить, что назначение алфавитов на GR выполняется только в режиме VT200.

Текущее назначение условных алфавитов и таблиц GR, GL запоминается в числе прочих параметров при выполнении команды ESC 7, и восстанавливается при выполнении ESC 8.

10.4. Изменение атрибутов строк

Некоторая строка экрана может быть использована как строка с символами одинарной или двойной ширины, одинарной высоты или как нижняя или верхняя половина строки двойной высоты. Чистые строки, появляющиеся при сдвиге текста, имеют атрибуты одинарной высоты и ширины. Когда

строка полностью стирается в результате операции «стирание в пределах экрана», она теряет все атрибуты, и символы, выводимые на нее впоследствии, будут иметь обычные (одинарные) размеры.

Атрибуты строк изменяются с помощью следующих ESC-последовательностей:

ESC # 6 (33 43 66) — символы в строке становятся двойной ширины одинарной высоты. Все бывшие на ней символы расширяются в два раза, а те что не вошли — теряются. Курсор на такой строке выводится также удвоенным по ширине. Символы, выводимые впоследствии в эту строку, будут иметь двойную ширину.

ESC # 5 (33 43 65) — символы в строке снова приобретают одинарную ширину и высоту. Если они были двойной ширины, то сжимаются в два раза, а освободившаяся правая половина строки заполняется пробелами.

ESC # 3 (33 43 63) — строка превращается в верхнюю половину строки двойной высоты двойной ширины.

ESC # 4 (33 43 64) — строка превращается в нижнюю половину строки двойной высоты двойной ширины.

Полная строка двойной высоты получается из двух строк, в которые необходимо вывести одну и ту же информацию. При изменении высоты строки информация, содержащаяся в ней, не теряется. Так, строка, являющаяся верхней половиной, может быть превращена в нижнюю половину или в строку с символами нормальной высоты.

10.5. Уровни управления режима ANSI

В режиме ANSI дисплей обеспечивает три уровня протокола: VT100, VT200 7 бит, VT200 8 бит. В режиме VT100 и VT200 7 бит дисплей не принимает управляющие символы таблицы C1. Кроме того, в режиме VT100 не выполняются некоторые функции.

Когда дисплей был в режиме совместимости (например, после включения питания ЭВМ), после приема ESC < он переходит в режим ANSI уровня VT100. Приведенные ниже CSI-последовательности позволяют установить любой уровень работы дисплея:

CSI 61"p (коды 33 133 66 61 42 160)
или CSI 61;0"p (33 133 66 61 73 60 42 160) — режим VT100;

CSI 62;1"p (33 133 66 62 73 61 42 160) — режим VT200 7 бит;

CSI 62;2"p (33 133 66 62 73 62 42 160)
или CSI 62;0"p (33 133 66 62 73 60 42 160)

или CSI 62"p (33 133 66 62 42 160) — режим VT200 8 бит.

В режиме VT200 8 бит, кроме того, дисплей посылает в ЭВМ 8-битовые коды русских символов с клавиатуры и может посылать 8-битовые эквиваленты ESC-последовательностей, таких, как CSI, SS3.

Режим VT200 обеспечивается только ПЗУ-182. ПЗУ-181 не имеет такого режима, однако в режиме VT100 обрабатывает все SCS-последовательности и команды переключения алфавитов.

10.6. 7- или 8-битовая передача управляющих символов

Дисплей посылает в ЭВМ различную информацию. Прежде всего, это коды символов, соответствующих нажатию различных клавиш алфавитно-цифровой клавиатуры, затем это последовательности, посылаемые при нажатии функциональных клавиш, кроме того, дисплей посылает ответы на некоторые запросы, поступающие из ЭВМ. Посылаемые последовательности начинаются обычно или с ESC [(коды 33 133), или с ESC O (коды 33 117). Первая последовательность называется CSI, вторая носит название SS3. Может быть включен режим 8-битовой передачи управляющих символов, когда вместо этих последовательностей посылается один символ таблицы C1, являющийся их 8-битовым эквивалентом. Так, в качестве CSI может быть послан символ с кодом 233, а в качестве SS3 — код 217. Режим посылки CSI и SS3 8-битовыми эквивалентами включается командой:

ESC пробел G (коды 33 40 106)
а выключается командой:

ESC пробел F (коды 33 40 107)

Эти команды выполняются только ПЗУ-182 в режиме VT200. В режиме VT100 дисплей всегда посылает только 7-битовые последовательности символов.

Прикладные программы, использующие возможности уровня VT200, должны уметь распознавать оба вида посылок.

10.7. Установка и сброс режимов работы дисплея

Дисплей способен отображать в строке до 80 или 132 символов обычной ширины, производить плавный рулон в области или скачкообразный, реверсировать экран, запрещать или разрешать переход курсора из крайней правой позиции в строке в следующую при приеме символа. Эти и другие режимы устанавливаются и сбрасываются с помощью CSI-последовательностей, описанных ниже.

Включение производится последовательностью:
CSI P1; P2; Pn h (последний код 150)

Для выключения посылается последовательность:
CSI P1; P2; ... Pn I (последний код 154 — буква I)
Здесь CSI — либо символ CSI (код 233), либо ESC
[(коды 33 133);

P1; P2; ... Pn — десятичные числа, разделенные символами «;», задающие включаемые или выключаемые режимы. Параметру может предшествовать символ «?», в этом случае дисплей предполагает, что такой символ стоит перед всеми параметрами, например, последовательность CSI 3 ; ? 4 I эквивалентна CSI ? 3 I CSI ? 4 I.

В одной последовательности может быть задано до 21 параметра.

Многие из параметров дисплея могут быть установлены не только ESC-последовательностями, но и посредством меню установки режимов, вызываемого нажатием клавиши УСТ РЕЖИМА. Эта возможность обеспечивается лишь тогда, когда в КЦГД загружена программа графического протокола KeyGP.

10.7.1. Блокирование клавиатуры

CSI 2 h (коды 33 133 62 150) — блокирует клавиатуру, при этом дисплей при нажатии клавиш не посылает символы в ЭВМ. Когда клавиатура заблокирована, на ней загорается индикатор ОЖИД.

CSI 2 l (коды 33 133 62 154) — разблокирует клавиатуру, при этом информация с нее вновь начинает поступать в ЭВМ. При разблокировании клавиатуры индикатор ОЖИД гаснет. Клавиатура разблокируется также при выполнении частичного сброса.

Блокирование клавиатуры выполняется программой, входящей в состав графического протокола KeyGP для ПЗУ-182.

10.7.2. Режим вставки/замещения

CSI 4 h (коды 33 133 64 150) — включает режим раздвижки в строке. При этом все вводимые символы не замещают прежние, а вставляются в строку, раздвигая ее вправо. Вытесняемые символы теряются. После включения или сброса этот режим выключен.

CSI 4 l (коды 33 133 64 154) — выключает режим раздвижки. Вводимые символы, как обычно, будут замещать прежние без раздвижки строки. Режим выключается также при выполнении частичного сброса дисплея.

Режим вставки обеспечивается только ПЗУ-182.

10.7.3. Локальное эхо

CSI 12 l (коды 33 133 61 62 154) — включает режим локального эха. При этом информация, посылаемая с клавиатуры в ЭВМ, одновременно будет обрабатываться дисплеем так, как будто бы она пришла из ЭВМ. Например, при нажатии буквенных клавиш на клавиатуре буквы будут также появляться на экране, как если бы они были посланы в него программой. Первоначально этот режим выключен.

CSI 12 h (коды 33 133 61 62 150) — режим локального эха выключается. При этом информация, идущая с клавиатуры, будет посылаться в ЭВМ, не отображаясь на экране. Прикладная программа или операционная система должна послать принятый символ обратно в дисплей, чтобы отобразить вводимую информацию. Все операционные системы именно так и поступают.

Режим локального эха обеспечивается только ПЗУ-182.

10.7.4. Перевод строки — новая строка

CSI 20 h (коды 33 133 62 60 150) — включение режима новой строки. При этом дисплей при обработке символа LF, а также FF и VT, наряду с передвижением курсора на одну строку вниз, будет устанавливать его на начало следующей строки. Кроме того, в этом режиме клавиша BK, а когда специальный режим дополнительной клавиатуры выключен, то и ВВОД, посылает в ЭВМ при нажатии не один символ CR (код 15), а два символа — CR+LF (коды 15, 12). Обычно этот режим выключен.

CSI 20 l (коды 33 133 62 60 154) — выключение режима. Дисплей обрабатывает LF, а также FF и VT, передвигая курсор на одну строку вниз, позиция же его в строке остается неизменной. Клавиши BK и ВВОД посылают в ЭВМ при нажатии только один символ CR (код 15).

10.7.5. Режим клавиш управления курсором

CSI ? 1 l (коды 33 133 77 61 154) — после приема этой последовательности включается так называемый режим клавиш управления курсором. В этом режиме клавиши со стрелками при нажатии посылают в ЭВМ последовательности, начинающиеся с пары ESC [(коды 33 133) или символа CSI (код 233), если включен режим 8-битовой передачи. Если послать такую последовательность обратно в дисплей, то курсор передвинется в соответствующем направлении. Первоначально этот режим включен и включается также при выполнении частичного сброса дисплея.

CSI ? 1 h (коды 33 133 77 61 150) — после приема этих символов режим управления курсором выключается. Клавиши со стрелками будут посылать последовательности, начинающиеся с ESC O (коды 33 117) или символа SS3 (код 217). Посланные обратно в дисплей, эти последовательности

не выполняют перемещение курсора. Прикладные программы следует строить таким образом, чтобы они обрабатывали оба вида последовательностей.

10.7.6. Переход в режим совместимости

CSI ? 2 1 (коды 33 133 77 62 154) — переводит дисплей в режим совместимости с VT52 и КСМ ДВК-3. Если программа использует режим совместимости, то при запуске, а также при восстановлении экрана, она должна посылать такую последовательность, чтобы установить нужный ей режим. При переходе из режима ANSI в режим совместимости, установки, произведенные в режиме ANSI, не изменяются, за исключением того, что в G0 засылается алфавит ASCII, в G1 — кириллица, в G3 — псевдографика.

10.7.7. 80 или 132 символа в строке

Дисплей может выводить в строке 80 или 132 символа обычной ширины в матрице знакоместа 10×10 и 6×10 соответственно, причем символ занимает в знакоместе 8×10 или 5×10 (кроме символов псевдографики и забоя). Выбор количества символов в строке производится следующими последовательностями:

CSI ? 3 h (коды 33 133 77 63 150) — переводит дисплей в режим отображения 132-х символов в строке.

CSI ? 3 l (коды 33 133 77 63 154) — переводит дисплей в режим отображения 80 символов в строке.

При приеме этих последовательностей (независимо от того, изменяется ли текущий режим) экран очищается, курсор устанавливается в начало экрана, область рулона назначается на полный экран, режим относительной адресации выключается.

10.7.8. Управление рулоном

Обычно при рулоне строки перемещаются по экрану «скачком», однако в режиме ANSI можно включить плавный рулон. При этом сдвиг информации по экрану, например при выводе длинного текста, будет происходить плавно, со скоростью 6 строк в секунду. Выводимые строки будут как бы всплывать из-под нижней границы экрана.

CSI ? 4 h (коды 33 133 77 64 150) — включает плавный рулон. Первоначально этот режим выключен.

CSI ? 4 l (коды 33 133 77 64 154) — выключает плавный рулон. Сдвиг текста снова будет выполняться скачком.

10.7.9. Установка негативного/положительного экрана

CSI ? 5 h (коды 33 133 77 65 150) — инверсия экрана. Фон экрана становится светлым, а символы — темными.

CSI ? 5 l (коды 33 133 77 65 154) — выключение инверсии

экрана. Фон становится темным, а символы — светлыми.

В ПЗУ-181 обе этих функции выполняются как изменение состояния экрана на обратное.

10.7.10. Режим относительной адресации курсора в области рулона

CSI ? 6 h (коды 33 133 77 66 150) — включение режима относительной адресации курсора. В этом режиме номер строки курсора отсчитывается относительно начала области рулона и курсор никакими командами не может быть выведен из нее. После приема последовательности курсор устанавливается в начале области. Первоначально этот режим выключен.

CSI ? 6 l (коды 33 133 77 66 154) — выключение режима относительной адресации курсора. Курсор далее адресуется относительно начала экрана. После приема такой последовательности курсор устанавливается в начало экрана. При выполнении частичного сброса дисплея режим не выключается.

Текущее состояние режима относительной адресации запоминается в числе прочих параметров при выполнении команды ESC 7, и восстанавливается при выполнении ESC 8.

10.7.11. Автоперевод курсора в новую строку

Когда при выводе текста очередной символ выводится в последнюю позицию строки, то следующий выводимый символ заставит курсор перейти на следующую строку, возможно, с рулоном. Это называется автопереводом. После вывода символа в последнюю позицию строки курсор все еще остается на этой позиции.

CSI ? 7 l (коды 33 133 77 67 154) — выключение режима автоматического начала новой строки при выводе текста. Это значит, что после вывода символа в последнюю позицию строки курсор останется в этой позиции и все последующие символы до CR или LF будут выводиться туда же.

CSI ? 7 h (коды 33 133 77 67 150) — включение режима автоматического начала новой строки при выводе текста. Это значит, что при попытке вывода второго символа в последнюю позицию строки курсор перейдет на новую строку и вывод будет продолжен с начала следующей строки. Первоначально этот режим включен, он включается также при частичном сбросе.

10.7.12. Автоповтор клавиатуры

CSI ? 8 h (коды 33 133 77 70 150) — включение режима автоповтора клавиатуры. Это значит, что при длительном нажатии одной клавиши дисплей будет повторять посылку

в ЭВМ ее кода с частотой примерно 30 раз в секунду. Первоначально этот режим включен.

CSI ? 8 1 (коды 33 133 77 70 154) — выключение автоповтора клавиш. При нажатии некоторой клавиши дисплей будет посылать ее код только один раз. Чтобы послать символ еще раз, требуется повторное нажатие.

Выключение автоповтора выполняется только ПЗУ-182.

10.7.13. Изменение режима вывода алфавита

Для работы с программным обеспечением, использующим сокращенный набор кода КОИ-7 (только большие русские и латинские буквы), в управляющей программе КЦГД предусмотрен режим, когда выводятся только большие русские и большие латинские буквы. Он включается последовательностью:

CSI ? 9 1 (коды 33 133 77 71 154) — большие русские, большие латинские буквы. Для выключения его необходимо послать:

CSI ? 9 h (коды 33 133 77 71 150) — обычный вывод (большие латинские, маленькие латинские или большие русские, маленькие русские буквы).

При работе с сокращенным набором символов клавиша РУС/ЛАТ на клавиатуре не посылает кодов в ЭВМ, а используется для внутренних переключений. Коды РУС, ЛАТ, поступающие от ЭВМ, не влияют на выбор алфавита GL. Коды символов от 40 до 137 выводятся согласно GL, а коды 140—177 — согласно GR, в котором по включению питания установлен алфавит кириллицы.

Для ПЗУ-181 этот режим обеспечивается загружаемой программой, входящей в состав графического протокола KeyGP.

10.7.14. Включение/выключение индикации курсора

CSI ? 25 1 (коды 33 133 77 62 65 154) — выключение индикации курсора. Мигающий курсор, отображающий место, куда будет выводиться текстовая информация, перестает быть виден на экране.

CSI ? 25 h (коды 33 133 77 62 65 150) — включение курсора. Курсор снова становится виден. Он включается также при выполнении частичного сброса дисплея.

10.7.15. Специальный режим клавиш верхнего ряда

В специальном режиме 15 клавиш верхнего ряда клавиатуры (от ПЕРЫВ до Ф20) выдают последовательности, содержащие номер клавиши. Этот режим обеспечивается программой графического протокола KeyGP для совместимости с программами, разработанными для ЭВМ «Электроника-85».

Включение режима производится командой:

CSI ? 3 9 h (коды 33 133 77 63 71 154).

Для выключения его необходимо послать:

CSI ? 3 9 1 (коды 33 133 77 63 71 150).

При работе с ПЗУ-182 клавиатура выдает такие последовательности также и в режиме VT200.

Дополнительные загруженные программы могут обрабатывать сброс и установку дополнительных режимов, не описанных выше.

10.8. Установка области рулона

Область рулона определяется номером начальной и конечной строки. Первоначально областью рулона является весь экран. Для установки другой следует послать последовательность:

CSI Pt ; Pb r (последний код 162)

Здесь CSI — либо символ CSI (код 233), либо ESC [(коды 33 133);

Pt — номер верхней строки области (не больше 23). Если не задан, или задан неправильно, то принимается равным 1.

Pb — номер нижней строки области (не меньше 2). Если не задан, или задан неправильно, то принимается равным 24.

Pt и Pb представлены в виде десятичных чисел (см. описание формата CSI-последовательности).

Номер верхней строки должен быть меньше номера нижней, то есть область должна состоять не менее чем из двух строк. Напомним, что строки пронумерованы сверху вниз, начиная с единицы.

После приема этой команды курсор устанавливается на начало экрана, а если включен режим относительной адресации, то на начало области рулона.

10.9. Вставка и удаление строк

Дисплей может вставить в текст заданное число пустых строк, при этом строки от текущей, включая ее, до нижней границы области рулона, сдвигаются вниз, вытесняемые строки теряются.

По другой команде дисплей удаляет из текста заданное число строк под курсором, включая ту, на которой он находится. Текст от текущей строки до нижней границы области рулона сдвигается вверх, с образованием соответствующего числа пустых строк.

Вставка строк выполняется командой:

CSI Pn L (последний код 114)

а удаление — по команде:

CSI Pn M (последний код 115)

Здесь P_n — число строк (в виде десятичного числа). Если P_n равно нулю или отсутствует, то вставляется или удаляется одна строка. Перед выполнением указанных команд курсор должен находиться в области рулона, иначе команда игнорируется.

После выполнения команд курсор устанавливается на начало текущей строки.

Эти функции выполняются только ПЗУ-182.

10.10. Задание атрибутов символов

Дисплей может выводить символы с различными атрибутами — подчеркнутые, негативные, дополнительного цвета, и мигающие в такт с курсором. Включаются и выключаются эти атрибуты командной последовательностью вида:

$CSI P_1; P_2; \dots P_n m$ (последний код 155)

где CSI — либо символ CSI (код 233), либо $ESC [$ (коды 33 133);

$P_1; P_2; \dots P_n$ — десятичные числа, разделенные символами «;», задающие включаемые или выключаемые атрибуты:

0 (код 60) (или параметр отсутствует) — выключаются все атрибуты;

1 (код 61) — включается вывод символов дополнительного цвета;

22 (62 62) — вывод символов нормального цвета;

4 (64) — включается вывод подчеркнутых символов;

24 (62 64) — подчеркивание выключается;

5 (65) — включается вывод мигающих символов, цвет которых попеременно меняется с нормального на дополнительный;

25 (62 65) — вывод мигающих символов отменяется;

7 (67) — включается вывод негативных символов;

27 (62 67) — вывод негативных символов выключается.

Атрибуты могут включаться и выключаться в произвольных сочетаниях, однако, когда включено мигание, признак основного/дополнительного цвета игнорируется.

Текущие атрибуты выводимых символов запоминаются в числе прочих параметров при выполнении команды $ESC 7$, и восстанавливаются при выполнении $ESC 8$.

При выполнении полного или частичного сброса дисплея все атрибуты символов выключаются.

10.11. Управление курсором

Различные команды дисплея позволяют передвигать курсор на определенное число позиций вверх или вниз, вправо или влево, устанавливать на произвольную позицию экрана, включать и выключать его индикацию.

Текущие координаты курсора можно запомнить в числе прочих параметров при выполнении команды $ESC 7$, и восстановить командой $ESC 8$.

Позиция курсора изменяется, во первых, управляющими символами CR, LF, FF, VT, BS, HT , действие которых было описано в главе «Управляющие символы таблицы $C0$ ». Другие команды таковы:

$ESC E$ (коды 33 105) или NEL (код 205) — новая строка. Курсор переходит на начало следующей строки. Если курсор находился на последней строке области рулона, то текст в этой области передвигается на строку вверх, появляется новая чистая строка, курсор остается на ее начале.

$ESC D$ (33 104) или IND (204) — курсор вниз на строку с сохранением позиции в строке. Если курсор находился на последней строке области рулона, то текст в этой области передвигается на строку вверх, появляется новая чистая строка, курсор остается на ней в прежней позиции.

$ESC M$ (33 115) или RI (215) — курсор вверх на строку с сохранением позиции в строке. Если курсор находился на первой строке области рулона, то текст в этой области передвигается на строку вниз, появляется новая чистая строка, курсор остается на ней в прежней позиции.

$CSI P_n A$ (последний код 101) — курсор вверх на заданное число строк. Здесь P_n — десятичное число, указывающее, на сколько строк он должен переместиться. Если оно не задано, или равно нулю, то курсор перемещается на одну строку. При движении курсор не может перейти через верхнюю границу области рулона, и если достигает ее, то там и остается.

$CSI P_n B$ (последний код 102) — курсор вниз на P_n строк. Если P_n не задано, или равно нулю, то курсор перемещается на одну строку. При движении курсор не может перейти через нижнюю границу области рулона, и если достигает ее, то там и остается.

$CSI P_n C$ (последний код 103) — курсор вправо на P_n позиций. Если P_n не задано или равно нулю, то курсор перемещается на одну позицию. Курсор не может перейти через правую границу экрана, при такой попытке он остается в последней позиции строки.

$CSI P_n D$ (последний код 104) — курсор влево на P_n позиций. Если P_n не задано или равно нулю, то курсор перемещается на одну позицию. Если курсор достигает начала строки, то там и остается.

$CSI P_l ; P_c H$ (последний код 110) — установка курсора

на строку P1 и позицию в строке Pс. Позиции в строке нумеруются, начиная с 1. Если режим относительной адресации выключен, то указывается номер строки от начала экрана (верхняя строка имеет номер 1), и курсор может быть установлен на любую строку. Если же включена относительная адресация, то номер строки отсчитывается от начала области рулона (верхняя строка области имеет номер 1) и курсор может быть установлен только на строки в пределах этой области. Если параметр P1 или Pс опущен или имеет нулевое значение, то предполагается, что он равен единице. Так, последовательность CSI H устанавливает курсор на начало экрана.

CSI P1 ; Pс f (последний код 146) — устанавливает курсор таким же образом, что и предыдущая описанная функция. Стандарты не рекомендуют использовать эту последовательность.

Позиция курсора изменяется также в результате команд, описанных в предыдущих разделах. Так, команды установки области рулона (CSI Pt ; Pb r) и включения/выключения режима относительной адресации (CSI ? 6 h и CSI ? 6 l) устанавливают курсор на начало экрана, если режим относительной адресации выключен, или на начало области рулона, если этот режим включен.

10.12. Установка и сброс позиций табуляции

Символ HT (код 11) заставляет курсор двигаться вправо до следующей позиции табуляции (табулятор). Первоначально они назначены на 1, 9, 17 (и далее через 8) позиции строки.

Последовательность ESC H (коды 33 110) или символ HTS (код 210) устанавливает новый табулятор на месте, где расположен курсор. Последовательность CSI 3 g (33 133 63 147) отменяет табулятор в текущей позиции курсора, а последовательность CSI g (33 133 147) или CSI 0 g (33 133 60 147) отменяет все табуляторы.

Установить или отменить табуляторы можно также посредством меню установки режимов, которое описано в данном документе.

10.13. Стирание текста в пределах строки и экрана

В пределах строки дисплей может выполнить стирание от ее начала до текущей позиции курсора, включая ее, от текущей позиции до конца строки, а также строки целиком.

CSI 0 K (33 133 60 113) или CSI K (33 133 113) — стирает символы от текущей позиции, включая ее, до конца строки.

CSI 1 K (33 133 61 113) — стирает символы от начала строки до текущей позиции курсора, включая ее.

CSI 2 K (33 133 62 113) — стирает символы во всей строке, на которой стоит курсор. Позиция курсора не изменяется.

В режиме VT200 в строке можно стереть произвольное число символов вправо от текущей позиции курсора, включая ее. Это делает последовательность:

CSI Pn X (последний код 130)

Здесь Pn — число стираемых символов. Если оно не задано или равно нулю, то берется равным 1. Эта функция выполняется только ПЗУ-182.

В пределах экрана также можно стереть от начала до курсора, от курсора до конца, или весь экран целиком.

CSI 0 J (33 133 60 112) или CSI J (33 133 112) — стирает от текущей позиции курсора, включая ее, до конца экрана.

CSI 1 J (33 133 61 112) — стирает от начала экрана до текущей позиции курсора, включая ее.

CSI 2 J (33 133 62 112) — стирает экран целиком.

При стирании в пределах экрана строки, стерты полностью, теряют атрибуты двойной ширины и двойной высоты, устанавливаемые через ESC #6, ESC #3, ESC #4. Символы, выводимые впоследствии в эти строки, будут иметь нормальные размеры.

10.14. Выборочное стирание

Выборочное стирание в пределах строки и экрана выполняется только ПЗУ KP1801PE2—182 в режиме VT200. Суть его заключается в том, что символы, защищенные от такого стирания, остаются, прочие же стираются. При выборочном стирании атрибуты стираемых символов не изменяются, то есть если, например, стираются подчеркнутые символы, то линия подчеркивания остается, а если стирается негативный символ, то вместо него остается негативное знакоместо. При выполнении «обычного» стирания признак защиты не принимается во внимание.

После приема определенной управляющей последовательности все символы, выводимые на экран, становятся защищенными от выборочного стирания. Другая последовательность выключает это состояние, после нее вновь выводятся незащищенные символы.

CSI 1"q (коды 33 133 61 42 161) — включает вывод защищенных символов.

CSI 0"q (33 133 60 42 161)

или CSI "q (33 133 42 161)

или CSI 2"q (33 133 62 42 161) — выключает защиту выводимых символов.

Текущее состояние защиты выводимых символов запоминается в числе прочих параметров при выполнении команды ESC 7, и восстанавливается при выполнении ESC 8.

Стирание выполняется следующими управляющими последовательностями:

CSI ? 0 K (33 133 77 60 113)

или CSI ? K (33 133 77 113) — стирает незащищенные символы от текущей позиции, включая ее, до конца строки.

CSI ? 1 K (33 133 77 61 113) — стирает незащищенные символы от начала строки до текущей позиции курсора, включая ее.

CSI ? 2 K (33 133 77 62 113) — стирает незащищенные символы во всей строке, на которой стоит курсор. Позиция курсора не изменяется.

CSI ? 0 J (33 133 77 60 112)

или CSI ? J (33 133 77 112) — стирает незащищенные символы от текущей позиции, включая ее, до конца экрана.

CSI ? 1 J (33 133 77 61 112) — стирает незащищенные символы от начала экрана до текущей позиции курсора, включая ее.

CSI ? 2 J (33 133 77 62 112) — стирает незащищенные символы на всем экране.

При выборочном стирании атрибуты строк не изменяются.

10.15. Раздвижка и сдвижка символов в строке

Дисплей может сдвинуть символы от текущей позиции курсора, включая ее, и до конца строки, на заданное число позиций вправо, при этом в строку вставляется соответствующее число пробелов. Вытесняемые символы теряются, а позиция курсора не изменяется. Для этого подается команда:

CSI Rn @ (последний код 100)

Здесь Rn — число позиций, на которое должен быть выполнен сдвиг. Если Rn не задано или равно нулю, то выполняется сдвиг на одну позицию. Команда выполняется только в режиме VT200.

По другой команде дисплей сдвигает символы от текущей позиции курсора, включая ее, и до конца строки, влево на заданное число позиций, причем символы, уходящие левее курсора, теряются так, что часть строки левее его не изменяется. В конце строки появляется соответствующее число пробелов. Эта функция выполняется как в режиме VT200, так и в режиме VT100 командой:

CSI Rn P (последний код 120)

Здесь Rn — число позиций, на которое должен быть выполнен сдвиг. Если Rn не задано или равно нулю, то выполняется сдвиг на одну позицию. Все эти функции выполняются только ПЗУ-182.

10.16. Сброс дисплея (начальная установка)

Дисплей выполняет две разновидности начальной установки — полный сброс (RIS) и частичный сброс (Soft Reset). Полный сброс в режиме ANSI выполняется командой ESC с (коды 33 143) и переводит дисплей в такое же состояние, как после включения питания. В режиме VT52 то же выполняется командой ESC V (33 126). Частичный сброс выполняется только в режиме ANSI командой CSI ! p (коды 33 133 41 160) и заключается в следующем:

- для алфавитов G0—G3 и таблиц GL, GR устанавливаются исходные значения, как после включения питания;
- алфавит для одного символа отменен;
- автоматический переход на новую строку, включен;
- все атрибуты символа отменяются и защита выводимых символов от выборочного стирания также выключается;
- устанавливается режим клавиш управления курсором;
- специальный режим дополнительной цифровой клавиатуры выключается;
- клавиатура разблокируется;
- режим вывода с раздвижкой строки отменяется;
- индикация курсора включается;
- программирование верхнего ряда клавиш в режиме VT200 разрешено;
- в буфере состояния курсора (который используется функциями ESC 7 и ESC 8) записываются те же значения, что и при включении питания;
- текущее положение курсора и режим относительной адресации не изменяются.
- режимы работы графического протокола устанавливаются в состояние, описанное в главе «ГРАФИЧЕСКИЙ ПРОТОКОЛ KeyGP ДЛЯ КЦГД».

Функция частичного сброса отсутствует в ПЗУ и выполняется программой, входящей в состав графического протокола KeyGP, загружаемого в КЦГД, или второй ПЗУ. Частичный сброс может быть также выполнен через меню установки режимов. В этом меню имеется и операция «Начальная установка», при выполнении которой все режимы

устанавливаются в состояние, соответствующее полному сбросу.

10.17. Сообщения от дисплея в ЭВМ

Дисплей обрабатывает специальные команды, называемые запросами, в ответ на которые он посылает сообщения в ЭВМ.

При приеме символа ENQ (код 5) дисплей посылает запрограммированный ответ. Этот ответ задается оператором в режиме «меню», когда загружена программа графического протокола KeyGP.

По приему ESC Z (33 132) дисплей в режиме VT52 посылает ESC/Z (33 57 132). Если же дисплей находится в режиме ANSI, то он посылает строку первичных атрибутов. Строка первичных атрибутов посылается также по команде CSI с (33 133 143) или CSI 0 с (33 133 60 143). Если дополнительный пакет программ не загружен в КЦГД, то дисплей посылает последовательность CSI ?1;2с (коды 33 133 77 61 73 62 143). Когда в КЦГД загружена программа протокола KeyGP, то выдается ответ:

CSI ?61;1с — ПЗУ-181;

CSI ?62;1;4;6;7;8с — ПЗУ-182.

По команде CSI > с (33 133 76 143)

или CSI > 0 с (33 133 76 60 143) дисплей выдает строку вторичных атрибутов — CSI >7;501с — ПЗУ-181; CSI >1;182;0с — ПЗУ-182.

Команда CSI 5 п (33 133 65 156) запрашивает у дисплея, исправлен ли он. В ответ всегда посылается CSI 0 п (33 133 60 156).

По команде CSI 6 п (33 133 66 156) дисплей посылает текущие координаты курсора — номер строки и его позицию в строке в виде:

CSI P1; Pс R (последний код 122)

Здесь P1 — номер строки, а Pс — номер столбца в виде десятичных чисел, разделенных точкой с запятой. Номер строки отсчитывается от начала экрана, причем верхняя строка имеет номер 1, но если включен режим относительной адресации, то номер строки отсчитывается от начала области рулона. Номер столбца отсчитывается от начала строки, начиная с 1.

Если в КЦГД с ПЗУ-182 загружена программа графического протокола KeyGP, то выдаются также ответы на следующие запросы:

CSI ? 1 5 п (33 133 77 61 65 156) — запрос состояния принтера. Так как КЦГД не имеет порта печатающего уст-

ройства, он посылает ответ CSI ? 1 3 п (33 133 77 61 63 156).

CSI ? 2 5 п (33 133 77 62 65 156) — запрос разрешения программирования функциональных клавиш верхнего ряда клавиатуры. Если клавиши можно программировать, посылается:

CSI ? 2 0 п (33 133 77 62 60 156), если же они заблокированы, посылается:

CSI ? 2 1 п (33 133 77 62 61 156).

CSI ? 2 6 п (33 133 77 62 66 156) — запрос текущего языка клавиатуры. Посылается ответ:

CSI ? 2 7 ; 1 п (33 133 77 62 67 73 61 156).

10.18. Заполнение экрана для настройки ЭЛТ

По команде ESC # 8 (коды 33 43 70) весь экран заполняется буквами «Е», курсор устанавливается на начало экрана, режим относительной адресации выключается, границы области рулона устанавливаются на полный экран. Текущие установки алфавитов не изменяются, атрибуты выводимых символов также остаются неизменными.

Эта функция выполняется программой графического протокола KeyGP, только совместно с ПЗУ-182.

11. РАБОТА КЛАВИАТУРЫ

Блок клавиатуры «Электроника МС 7004» содержит основную (алфавитно-цифровую) клавиатуру, клавиши-стрелки и клавиши НТ, ВСТ, УДАЛ, ВЫБР, ПРЕД КАДР, СЛЕД КАДР, а также дополнительную цифровую клавиатуру, на которой кроме цифр имеются клавиши ВВОД, ПФ1, ПФ2, ПФ3, ПФ4, [,], [—], [·] (запятая, минус и точка). Над основной клавиатурой имеются 20 клавиш верхнего ряда.

При нажатии клавиш в ЭВМ посылаются соответствующие им символы или управляющие последовательности символов. Дисплей имеет буфер на 32 символа, который хранит информацию до тех пор, пока ЭВМ не сможет принять ее. В обычном состоянии (дуплексный режим), чтобы посылаемые символы отобразились на экране, прикладная программа или операционная система должна послать принятые символы обратно в дисплей.

Дисплей может также работать в полудуплексном режиме (режим локального эха), когда вся информация, посылаемая в ЭВМ, принимается также самим дисплеем, в частности, символы сразу же отображаются на экране, не требуя

специального эха прикладной программы или операционной системы. Режим локального эха включается при приеме последовательности CSI 4 l, а выключается последовательностью CSI 4 h. Эти команды выполняются только в режиме ANSI.

Дисплей может работать в автономном режиме, когда выход дисплея отключается от ЭВМ и направляется на вход дисплея, при этом он не посылает информации в ЭВМ и принимает ее только с собственной клавиатуры. Этот режим включается при нажатии клавиши УСТ РЕЖИМА на верхнем ряду клавиатуры и выключается повторным нажатием этой же клавиши. В автономном режиме с клавиатуры можно подать дисплею любые управляющие символы и последовательности.

Прикладная программа может выключить клавиатуру (заблокировать), послав в дисплей CSI 2 h. При этом на клавиатуре загорается индикатор ОЖИД. Когда клавиатура заблокирована, то при нажатии любых клавиш информация в ЭВМ не посылается. Для разблокирования программа должна послать дисплею CSI 2 l. Эти команды выполняются только в режиме ANSI. В автономном режиме клавиатура функционирует всегда, даже если была заблокирована.

При длительном нажатии большинства клавиш посылка соответствующего им кода повторяется 30 раз в секунду. Это называется автоповтором. При работе в режиме ANSI прикладная программа может запретить автоповтор, послав в дисплей команду CSI ? 8 l. Включается автоповтор посылкой CSI ? 8 h. Клавиши верхнего ряда, а также ВК, СУ, ВР, ФКС, РУС/ЛАТ, КМП, ВК, НТ, ВСТ, УДАЛ, ВЫБР, ПРЕД КАДР, СЛЕД КАДР не имеют автоповтора.

Алфавитно-цифровая (основная) клавиатура содержит клавиши для ввода букв, цифр, а также специальных символов. Кроме того, на ней находятся клавиши ВК, ЗБ, ТАБ, а также клавиша ввода управляющих символов СУ и клавиши переключения регистров ВР, ФКС, РУС/ЛАТ. Они выполняют следующие функции:

- ВК** — посылает символ CR (код 15). Если включен режим перевода строки новой строкой, то посылается пара символов CR LF (коды 15, 12).
- ЗБ** — посылает символ DEL (код 177). Прикладные программы и операционная система обычно интерпретируют этот код как стирание (забой) последнего введенного символа.

ТАБ — посылает символ НТ (код 11) — горизонтальная табуляция, который для различных прикладных программ имеет разное значение.

СУ — пока она нажата, алфавитно-цифровые клавиши будут посылать коды управляющих символов таблицы СО. Действие СУ заключается в том, что старшие 3 бита символов обнуляются, в результате чего посылаются коды от 1 до 37. Так, например, при нажатии СУ/[(код 133) или СУ/{ (код 173) посылается символ ESC (33).

ВР — сдвиг регистра на время нажатия. Пока она не нажата, клавиши с цифрами и специальными символами будут выдавать символы, нанесенные на их верхней половине. Если она нажата, то будут выдаваться символы, нанесенные на нижней половине клавиш. Код выдаваемый буквенными клавишами, кроме нажатия клавиши ВР зависит также от фиксации регистра.

ФКС — фиксация регистра для буквенных клавиш. Срабатывает только тогда, когда нажата клавиша ВР. В обычном состоянии клавиатура выдает большие латинские или малые русские буквы. После нажатия ВР/ФКС загорается индикатор ФКС, и клавиатура будет постоянно выдавать малые латинские и большие русские буквы. Повторное нажатие этих клавиш возвращает прежнее состояние.

РУС/ЛАТ

— переключение русского и латинского регистров. В 7-битовом режиме выдает при нажатии код SO или SI, в зависимости от текущего состояния клавиатуры. Операционная система должна послать этот код обратно в дисплей, при этом переключится алфавит, назначенный на кодовую таблицу GL, и изменится состояние индикатора ЛАТ на клавиатуре. В 8-битовом режиме символ в ЭВМ не посылается, дисплей сам обрабатывает состояние клавиатуры и формирует 8-битовый код русских и латинских символов по КОИ-8.

КМП

— ввод составных символов национальных европейских алфавитов. Работает только в режиме VT200, если загружена специальная программа. Для ввода составных букв сначала нажимается КМП, затем до двух клавиш с буквами и специальными символами. Когда была нажата КМП, светится индикатор КОМПОЗ.

Клавиатура может обмениваться с ЭВМ в режимах 8 или 7 бит, то есть в коде КОИ-8 или КОИ-7. В режиме 8 бит дисплей сам учитывает, какой регистр установлен — русский или латинский, и посылает с учетом этого коды букв. Русские буквы имеют коды от 300 до 377, а латинские буквы — в диапазоне от 100 до 176. При нажатии клавиши РУС/ЛАТ этот регистр меняется на обратный, что отображается индикатором ЛАТ. Коды символов @, ^, [,], \, нанесенных на клавишах [Ю], [Ч], [Ш], [Щ], [Э] соответственно, посылаются на латинском регистре.

В режиме 7 бит регистр клавиатуры (русский или латинский) должна обрабатывать операционная система или прикладная программа. Дисплей посылает 7-битовые коды русских и латинских букв, при этом коды прописных латинских и строчных русских букв совпадают, строчные латинские и прописные русские тоже имеют одинаковые коды. Русский и латинский регистры клавиатуры переключаются при нажатии клавиши РУС/ЛАТ. Если на GL был назначен русский алфавит, то клавиша РУС/ЛАТ посылает символ ЛАТ (SI), а если был назначен латинский алфавит, то посылается РУС (SO). Индикатор ЛАТ на верхнем ряду клавиатуры отображает текущее состояние кодовой таблицы GL. Прикладная программа или операционная система должна посылать принятый символ РУС или ЛАТ обратно на дисплей, чтобы индикатор отображал установленный регистр. Если программа обрабатывает тексты в коде КОИ-8, то принятые 7-битовые коды должны быть преобразованы в 8-битовые в соответствии с регистром.

В зависимости от текущего регистра и от нажатия клавиши ВР большинство клавиш основной клавиатуры могут посылать в ЭВМ коды различных символов. На клавишах четвертого ряда, а также на клавишах [:], [.), [.), [/) нанесено по два символа. Для ввода символов различного регистра используются клавиши ВР (временный сдвиг регистра) и ФКС (фиксация сдвига регистра). Когда индикатор «ФКС» над клавиатурой не горит, то на латинском регистре вводятся прописные (большие) буквы (верхний регистр), на русском регистре вводятся строчные (малые) буквы. Для буквенных клавиш можно зафиксировать состояние ввода противоположного регистра. Для этого нажмите одновременно ВР и ФКС, при этом загорится индикатор «ФКС» справа над клавиатурой. Повторное нажатие ВР и ФКС выключает этот режим. Для переключения регистра на обратный нажмите клавишу ВР (две такие клавиши имеются слева и справа от

клавиатуры). В то время, когда ВР нажата, буквенные клавиши будут вводить символы противоположного регистра. Клавиша ВР управляет также выбором символов небуквенных клавиш. Когда она не нажата, то клавиатура выдает символы, нанесенные на верхней половине клавиш (цифры, запятую, точку, минус, фигурные скобки). Когда эта клавиша нажата, выдаются коды символов, нанесенных на нижней половине клавиш.

Некоторые клавиши выдают символы, которые на них не нанесены. Это такие:

- ъ — твердый знак, вводится на нижнем русском регистре при нажатии клавиши «подчеркивание» (справа в нижнем ряду клавиатуры). Эта буква посылается также клавишей «ъ». В 8-битовом режиме она может выдавать код прописного твердого знака (377).
- ~ — верхняя черта (тильда), вводится на нижнем латинском регистре при нажатии клавиши [Ч]. Дисплей отображает ее волнистой чертой, однако в зависимости от типа печатающего устройства она может печататься и прямой чертой. Тильда вводится также клавишей слева от ВК, независимо от регистра.
- {и} — левая и правая фигурные скобки (используются в Паскале и Си), вводятся на нижнем латинском регистре при нажатии клавиш [Ш] и [Щ] соответственно. Фигурные скобки можно ввести также клавишами четвертого ряда клавиатуры, когда ВР не нажата.
- | — вертикальная черта, вводится клавишей [Э] на нижнем латинском регистре. Ее можно также ввести клавишей [{} четвертого ряда, когда нажата ВР.
- ' — обратный апостроф, вводится клавишей [Ю] на нижнем латинском регистре. Может быть также введен клавишей [}] (справа в четвертом ряду клавиатуры) с нажатой клавишей ВР.

11.1. Дополнительная цифровая клавиатура

Дополнительная цифровая клавиатура содержит 10 цифровых клавиш, а также клавиши ПФ1, ПФ2, ПФ3, ПФ4, ВВОД, [.), [—), [·). Коды, выдаваемые при нажатии ее клавиш, зависят от режима — совместимости или ANSI, а также от включенного специального режима дополнительной клавиатуры. В таблице приведены коды, выдаваемые клавишами в режиме совместимости.

КЛАВИША	Обычный режим		Специальный режим	
	(символы)	(коды)	(символы)	(коды)
0	0	60	ESC ? p	33 77 160
1	1	61	ESC ? q	33 77 161
2	2	62	ESC ? r	33 77 162
3	3	63	ESC ? s	33 77 163
4	4	64	ESC ? t	33 77 164
5	5	65	ESC ? u	33 77 165
6	6	66	ESC ? v	33 77 166
7	7	67	ESC ? w	33 77 167
8	8	70	ESC ? x	33 77 170
9	9	71	ESC ? y	33 77 171
, (запятая)	,	54	ESC ? l	33 77 154
- (минус)	-	55	ESC ? m	33 77 155
. (точка)	.	56	ESC ? n	33 77 156
ПФ1	ESC P	33 120	ESC P	33 120
ПФ2	ESC Q	33 121	ESC Q	33 121
ПФ3	ESC R	33 122	ESC R	33 122
ПФ4	ESC S	33 123	ESC S	33 123
ВВОД	CR	15	ESC ? M	33 77 115

Специальный режим включается командой ESC =, а выключается командой ESC >.

В режиме ANSI цифровая клавиатура выдает следующие коды:

КЛАВИША	Обычный режим		Специальный режим	
	(символы)	(коды)	(символы)	(коды)
0	0	60	SS3 p	33 117 160
1	1	61	SS3 q	33 117 161
2	2	62	SS3 r	33 117 162
3	3	63	SS3 s	33 117 163
4	4	64	SS3 t	33 117 164
5	5	65	SS3 u	33 117 165
6	6	66	SS3 v	33 117 166
7	7	67	SS3 w	33 117 167
8	8	70	SS3 x	33 117 170
9	9	71	SS3 y	33 117 171
, (запятая)	,	54	SS3 l	33 117 154
- (минус)	-	55	SS3 m	33 117 155
. (точка)	.	56	SS3 n	33 117 156
ПФ1	SS3 P	33 117 120	SS3 P	33 117 120
ПФ2	SS3 Q	33 117 121	SS3 Q	33 117 121
ПФ3	SS3 R	33 117 122	SS3 R	33 117 122
ПФ4	SS3 S	33 117 123	SS3 S	33 117 123
ВВОД	CR	15	SS3 M	33 117 115

Как и в режиме совместимости, специальный режим включается командой ESC =, а выключается командой ESC >.

Примечание 1. Когда включен режим перевода строки с новой строкой, то клавиша ВВОД в нормальном режиме дополнительной клавиатуры посылает пару символов CR+LF

(коды 15, 12) вместо одиночного символа CR.

Примечание 2. Последовательность SS3 — это символы ESC O (коды 33 117). Если включен режим передачи 8-битовых эквивалентов (только в режиме VT200 8 бит), то вместо этих двух символов передается код 217. Прикладные программы должны уметь распознавать оба вида посылок.

11.2. Клавиши управления курсором

Четыре клавиши со стрелками используются прикладными программами для управления курсором. В режиме совместимости они посылают такие последовательности кодов:

	Символы	Коды
Стрелка вверх	ESC A	33 101
Стрелка вниз	ESC B	33 102
Стрелка вправо	ESC C	33 103
Стрелка влево	ESC D	33 104

В режиме ANSI эти клавиши могут выдавать два вида последовательностей. Обычно включен так называемый режим управления курсором, когда вырабатываемые последовательности могут использоваться для передвижения курсора. Эти последовательности начинаются с CSI (ESC [). Режим управления курсором может быть выключен командой CSI ? 1 h. После этого передаваемые последовательности будут начинаться с SS3 (ESC O). Режим управления курсором можно снова включить, подав дисплею команду CSI ? 1 l. Передаваемые коды приведены в следующей таблице:

	Режим курсора включен		Режим курсора выключен	
	Символы	Коды	Символы	Коды
Стрелка вверх	CSI A	33 133 101	SS3 A	33 117 101
Стрелка вниз	CSI B	33 133 102	SS3 B	33 117 102
Стрелка вправо	CSI C	33 133 103	SS3 C	33 117 103
Стрелка влево	CSI D	33 133 104	SS3 D	33 117 104

Примечание. Последовательность CSI — это символы ESC [(коды 33 133), а SS3 — это ESC O (33 117). Если включен режим передачи 8-битовых эквивалентов (только в режиме VT200 8 бит), то CSI передается одним символом 233, а SS3 — символом 217. Прикладные программы должны уметь распознавать оба вида посылок. Желательно также, чтобы воспринимались как последовательности с CSI, так и с SS3.

11.3. Прочие функциональные клавиши

Ниже описаны клавиши верхнего ряда клавиатуры, а так-

же клавиши НТ, ВСТ, УДАЛ, ВЫБР, ПРЕД КАДР, СЛЕД КАДР.

Слева в верхнем ряду клавиатуры находятся 5 клавиш — СТОП КАДР, ПЕЧАТЬ КАДРА, ПАУЗА, УСТ РЕЖИМА, Ф5. Клавиша СТОП КАДР служит для остановки и продолжения вывода на экран. При первом нажатии она посылает в ЭВМ код 23 (X—OFF), который заставляет операционную систему останавливать вывод информации в дисплей. Одновременно с этим загорается индикатор СТОП КАДР. При втором нажатии на клавишу СТОП КАДР дисплей посылает код 21 (X—ON), который разрешает операционной системе продолжить вывод информации, при этом соответствующий индикатор на клавиатуре гаснет.

Клавиша ПЕЧАТЬ КАДРА функционирует только тогда, когда загружена программа графического протокола KeyGP. При нажатии ее совместно с СУ или ВР дисплей посылает в ЭВМ код 200. Обычно этот код отсеивается операционной системой, однако если включен драйвер печати копии экрана HC.SYS, то он перехватывает этот символ, запрашивает у дисплея содержимое экрана и выводит его на печать. Если было нажато СУ/ПЕЧАТЬ КАДРА, то содержимое экрана печатается поперек листа на 4 градации яркости. Если было нажато ВР/ПЕЧАТЬ КАДРА, то экран печатается вдоль листа в большем масштабе на 16 градаций яркости. Для остановки печати следует нажать эту клавишу без нажатия ВР или СУ.

Клавиши ПАУЗА и Ф5 не выполняют никаких функций.

Клавиша УСТ РЕЖИМА переводит дисплей в автономный режим. В этом режиме информация, вводимая с клавиатуры, не передается в ЭВМ, а подается самому дисплею, как будто она была принята из ЭВМ. Если в КЦГД загружена программа графического протокола KeyGP, то нажатие клавиши УСТ РЕЖИМА вызывает на экран меню установки режимов, с помощью которого можно установить многие из режимов, устанавливаемых программно, выполнить частичный сброс, очистить экран и т. п. Описание меню режимов содержится в данном документе.

При повторном нажатии этой клавиши дисплей вновь переходит в обычный режим.

Остальные 15 клавиш верхнего ряда и клавиши НТ, ВСТ, УДАЛ, ВЫБР, ПРЕД КАДР, СЛЕД КАДР работают в зависимости от режима. Их функции приведены в следующей таблице.

КЛАВИШИ	Режим VT52		Режим VT200	
	символы	коды	символы	коды
ПРЕРЫВ	—	—	CSI 17~	33 133 61 67 176
ПРОДОЛЖ	—	—	CSI 18~	33 133 61 70 176
ОТМЕН	—	—	CSI 19~	33 133 61 71 176
ОСНОВН КАДР	—	—	CSI 20~	33 133 62 60 176
ВЫХОД	—	—	CSI 21~	33 133 62 61 176
Ф11 AP2	ESC	33	CSI 23~	33 133 62 63 176
Ф12 ВШ	BS	10	CSI 24~	33 133 62 64 176
Ф13 ПС ДОП	LF	12	CSI 25~	33 133 62 65 176
ВАРИАНТ	ESC =	33 75	CSI 26~	33 133 62 66 176
ПМ	ESC a	33 141	CSI 28~	33 133 62 70 176
ИСП	ESC b	33 142	CSI 29~	33 133 62 71 176
Ф17	ESC H	33 110	CSI 31~	33 133 63 61 176
Ф18	ESC J	33 112	CSI 32~	33 133 63 62 176
Ф19	ESC K	33 113	CSI 33~	33 133 63 63 176
Ф20	ESC L	33 114	CSI 34~	33 133 63 64 176
НТ	ESC c	33 143	CSI 1~	33 133 61 176
ВСТ	ESC d	33 144	CSI 2~	33 133 62 176
УДАЛ	ESC e	33 145	CSI 3~	33 133 63 176
ВЫБР	ESC f	33 146	CSI 4~	33 133 64 176
ПРЕД КАДР	ESC g	33 147	CSI 5~	33 133 65 176
СЛЕД КАДР	ESC h	33 150	CSI 6~	33 133 66 176

Примечание 1. В режиме VT100 из перечисленных клавиш функционируют только Ф11 (AP2), которая выдает символ ESC (код 33), Ф12 (ВШ), которая выдает символ BS (10), и Ф13 (ПС), посылающая LF (12).

Примечание 2. В режиме совместимости клавиша ПРЕРЫВ изменяет цвет фона, перебирая при последовательных нажатиях 64 возможных цвета. Клавиша ПРОДОЛЖ изменяет таким же образом цвет символов. На ПЗУ-181 эти функции выполняются только совместно с программой графического протокола KeyGP.

Примечание 3. В автономном режиме перечисленные клавиши всегда функционируют так же, как в режиме совместимости. В частности, нажав клавишу НТ (ESC c), можно выполнить полный сброс дисплея в режиме ANSI.

Примечание 4. В режиме VT200 клавиши верхнего ряда при нажатии их совместно с ВР выдают предварительно запрограммированные последовательности символов (UDK —

User Defined Keys). Программирование их возможно только с ПЗУ-182 и только тогда, когда загружена программа графического протокола KeyGP.

Примечание 5. Последовательность CSI — это символы ESC [(коды 33 133). Если включен режим передачи 8-битовых эквивалентов (только в режиме VT200 8 бит), то вместо этих двух символов передается код 233. Прикладные программы должны уметь распознавать оба вида посылок.

Примечание 6. Программа графического протокола KeyGP обеспечивает выдачу клавишами верхнего ряда клавиатуры в режиме ANSI таких же последовательностей, какие выдает ПЗУ-182 в режиме VT200. Включение такой выдачи производится командой ESC [? 39 h, а выключение — командой ESC [? 39 l.

12. УСТАНОВКА ЦВЕТА ЭКРАНА И СИМВОЛОВ

Нажатием определенных клавиш можно установить произвольный цвет экрана и символов. Когда дисплей работает в режиме совместимости, то при последовательных нажатиях клавиши ПЕРЫВ перебираются 64 возможных цвета фона экрана, а при нажатии ПРОДОЛЖ — 64 возможных цвета символов. В режиме ANSI эти клавиши выполняют другие функции, однако, когда дисплей работает в автономном режиме, клавиши верхнего ряда всегда работают как в режиме совместимости, и клавишами ПЕРЫВ, ПРОДОЛЖ можно пользоваться для установки цвета. Для включения и выключения автономного режима используется клавиша УСТ РЕЖИМА.

В программе графического протокола KeyGP имеется более удобная процедура установки цветов. Она позволяет быстро установить для цвета экрана, символов и дополнительного цвета символов, по каждому из основных цветов, любую из 4-х интенсивностей. Это производится клавишами дополнительной цифровой клавиатуры при одновременно нажатых клавишах СУ и ВР. Клавиши [1], [2], [3] устанавливают соответственно интенсивность красного, зеленого и синего цветов для фона экрана, [4], [5], [6] — интенсивности основных цветов для цвета символов, и [7], [8], [9] — для дополнительного цвета символов. При последовательных нажатиях этих клавиш перебираются 4 степени яркости каждого из основных цветов (красного, зеленого, синего).

Посредством меню программы графического протокола

KeyGP можно также установить значение любого из 16 цветов графического режима.

13. ЗАГРУЗКА ДОПОЛНИТЕЛЬНОГО ПО В ОЗУ КЦГД

Управляющая программа КЦГД во время разбора принятых из ЭВМ кодов производит в некоторых точках переходы на подпрограммы через указатели, размещенные в ОЗУ КЦГД. Во время инициализации дисплея соответствующие ячейки указывают на пустые подпрограммы. После дозагрузки ПО в ОЗУ КЦГД можно расширить или изменить порядок отработки управляющих кодов и т. д. Для реализации расширения обрабатываемой системы команд или смены ее вообще имеется команда дозагрузки ПО, которая обрабатывается при установленном режиме совместимости.

Формат команды:

ESC	X	X	KM KC	AM AC
033	130	130	0..377	0..377

KM и KC — соответственно младший и старший байты 16-разрядного числа, которое определяет, сколько байтов дополнительного ПО будет передано в ОЗУ КЦГД.

AM и AC — соответственно младший и старший байты 16-разрядного числа, которое определяет адрес в ОЗУ КЦГД, начиная с которого будет производиться загрузка. На этот адрес будет передано управление по завершению приема (если он четный). Если предполагается совместное использование ПО из ПЗУ и ОЗУ, то рекомендуется адрес загрузки 40000(8), при этом, естественно, нужно знать структуру ПО в ПЗУ и распределение ОЗУ. Независимое ПО может загружаться с произвольного адреса, однако в этом случае необходимо помнить, что 0 банк ОЗУ КЦГД используется для обеспечения работы КЦГД, а 1-ый — для работы управляющей программы, размещенной в БИС ПЗУ КР1801РЕ2-181, 182.

14. ДОПОЛНИТЕЛЬНЫЕ ФУНКЦИИ ТЕКСТОВОГО РЕЖИМА, ОБЕСПЕЧИВАЕМЫЕ ПРОГРАММОЙ KeyGP

Программа графического протокола KeyGP кроме выполнения функций вывода графических изображений расширяет набор команд ПЗУ КР1801РЕ2-182 до полного протокола

VT220. Реализуются функции программирования клавиш верхнего ряда клавиатуры, загрузка прорисовок символов и загрузка графического изображения в сиксельном формате.

14.1. Клавиши, программируемые пользователем

В режиме VT200 клавиши верхнего ряда клавиатуры (кроме левых пяти клавиш) могут посылать наряду с выше описанными кодами произвольные кодовые последовательности, запрограммированные прикладной программой. Такая посылка происходит при нажатии клавиши совместно с ВР, если же ВР не нажато, то посылается обычная последовательность. Эта функция обрабатывается загружаемыми программами, например, входящими в состав графического протокола. В режимах VT100 и совместимости программно установленные управляющие клавиши не обрабатываются. Загрузка последовательностей кодов клавиш выполняется через следующую DCS-последовательность:

DCS Pc; P1 | ky1/st1; ky2/st2; ... ; kyn/stn ST

где

DCS — либо последовательность ESC P (33 120), либо символ с кодом 220 (только в режиме VT200 8 бит).

Pc — признак очистки клавиш;

P1 — признак блокировки назначений;

| — символ с кодом 174;

ky_n — номер клавиши;

st_n — код клавиши;

ST — последовательность ESC \ (коды 33 134) или символ ST (код 234, только в режиме VT200 8 бит).

Параметр Pc (признак очистки клавиш) может иметь следующие значения:

0 (код 60) или опущено

— очищает все клавиши перед загрузкой новых значений;

1 (код 61)

— очищает перед загрузкой клавиши, для которых значения были заданы ранее.

Параметр P1 (признак блокировки загрузки клавиш) может иметь такие значения:

0 (код 60) или опущено

— запрещает на будущее загрузку и переопределение клавиш (текущая операция все же выполняется);

1 (код 61)

— не запрещает в будущем загрузку и переопределение клавиш.

После выполнения частичного сброса блокировка загруз-

ки снимается. Установить или снять блокировку можно также посредством меню установки режимов клавиатуры.

После символа | (код 174), указывающего на то, что данная управляющая последовательность выполняет загрузку клавиш, следуют определения клавиш в формате kyp/stn; (они отделяются друг от друга символом «;» (код 73)), где kyp — десятичный номер клавиши, за ним следует символ «/» (код 57), а stn — строка символов, назначаемая на данную клавишу, закодированная шестнадцатиричными цифрами.

Клавиша	Значение kyp — последовательности	Восьмеричный эквивалент
ПРЕРЫВ	17	61 67
ПРОДОЛЖ	18	61 68
ОТМЕН	19	61 69
ОСНОВН КАДР	20	62 60
ВЫХОД	21	62 61
Ф11 (AP2)	23	62 63
Ф12 (ВШ)	24	62 64
Ф13 (ПС)	25	62 65
ДОП ВАРИАНТ	26	62 66
ПМ	28	62 68
ИСП	29	62 69
Ф17	31	63 61
Ф18	32	63 62
Ф19	33	63 63
Ф20	34	63 64

Назначаемая на клавишу последовательность записывается через символ «/» после ее номера в виде последовательности шестнадцатиричных цифр. Две шестнадцатиричные цифры (символы, лежащие в диапазонах 0 (60) — 9 (71), A (101) — F (106), a (141) — f (146)) кодируют один байт последовательности.

Буфер программируемых клавиш имеет размер 256 байт, таким образом, суммарный размер запрограммированной информации не может превысить этот объем. Когда переопределяется уже запрограммированная клавиша, то прежнее назначение очищается перед началом загрузки нового, и эта память становится свободной. После использования всех 256 байт никакие установки невозможны до тех пор, пока не будет очищена достаточная для них область памяти. Это можно сделать таким образом:

1. Переопределить функции каких-либо клавиш.
2. Очистить какие-либо клавиши.

3. Очистить предыдущие установки сбросом питания или программно.

Для того, чтобы гарантировать правильную обработку клавишами новых функций, желательно установить режим блокировки загрузки. Если установлен режим блокировки, то изменить установленные функции нельзя. В противном случае клавишам можно присвоить новые функции. Отменить блокировку можно только через полный или частичный сброс.

14.2. Загрузка новых прорисовок символов пользователем

В режиме VT200 в дисплей может быть загружен дополнительный набор прорисовок символов. Эта возможность обеспечивается программами, входящими в состав графического протокола KeyGR.

Для прорисовки символа используется матрица размером 10×10 пикселей при 80 колонках и 6×10 пикселей при 132 колонках в строке. Размер ячейки символа дисплея 8×10 и 5×10 пикселей соответственно, она называется фонтом символа. Пиксел (pixel=picture element) — это минимальный элемент изображения (точка).

Символ задается в виде последовательности сикселей (sixel=six pixels) — это вертикальная матрица пикселей размером 1×6 . Порядок ввода последовательности сикселей при этом — слева направо, сверху вниз. Старший значимый бит соответствует нижнему пикселу, младший — верхнему. Так как в нижних колонках содержится 4 пиксела, то 5-ый и 6-ой биты сикселей, задающих нижнюю часть символа, игнорируются. К коду сиксела, определяющему пиксели колонки (код от 0 до 77), прибавляется код 77, таким образом получают коды символов из диапазона 77—176 (от знака «?» до знака «~»). Эти символы и посылаются в дисплей в составе специальной DCS-последовательности.

Для загрузки прорисовок произвольного числа символов (до 94) необходимо ввести следующую DCS-последовательность:

DCS Pfn;Pcn;Pe;Pcms;Pw;Pt { Dscs Sxbp1;Sxbp2;...;Sxbpn ST

где Pfn

— номер фонта. Имеет значение 0 или 1. Дисплей игнорирует его.

Pcp — номер символа начала последовательности. Определяет номер символа начала последовательности фонтов символов, которые будут загружаться (от 1 — для символа «!» (41) до 94 для «~» (176)). Представля-

ется десятичными цифрами, подобно параметру CSI-последовательности.

Pe — управление стиранием фонта. Указывает, должны ли стираться прорисовки символов перед загрузкой:

0 или 2 — стираются все прорисовки, загруженные ранее;

1 — стирает только определяемые вновь.

Pcms — размер матрицы символа :

0 — по умолчанию (7×10);

1 — не используется;

2 — 5×10 ;

3 — 6×10 ;

4 — 7×10 ;

10 — 10×10 .

Pw — атрибут ширины:

0 или 1 — 80 колонок;

2 — 132 колонки.

Pt — устанавливает матрицу как фонтов текстового символа или как фонтов полной ячейки:

0 или 1 — текстовый символ;

2 — полное знакоместо.

Возможны следующие комбинации Pcms, Pw, Pt:

Pcms	Pw	Pt	Действие
0, 2, 3, 4 10	0, 1 0, 1	0, 1 2	Задает фонтов, который может быть использован только при 80 символах в строке
2	2	0, 1	Задает фонтов, который может быть использован только при 132 символах в строке
3	2	2	Задает фонтов, который может быть использован только при 132 символах в строке и режиме «полная ячейка»

Остальные комбинации игнорируются. Следует также заметить, что если прорисовка символа была загружена в расчете на 132 символа в строке, то при работе дисплея в режиме 80 символов вместо данного символа будет выводиться обращенный вопросительный знак. То же происходит и в обратном случае. Разные символы могут иметь разный признак ширины.

За Pt следует символ { (код 173), который указывает на то, что данная DCS-последовательность загружает набор символов. После него идет Dscs-комбинация символов, кото-

рая определяет имя набора фонтов и используется в ESC-последовательностях выбора алфавита. Она может содержать до двух (или ни одного) начальных символов с кодами от 40 до 57, затем финальный символ с кодом от 60 до 176. Рекомендуем использовать имя, состоящее из пробела (код 40) и знака @ (код 100).

Sxbr — битовый образец сикселя. Каждый образец имеет форму:

S...S/S...S, где первая последовательность S...S представляет верхние колонки (сиксели) символа слева направо, слэш («/» с кодом 57) переводит на нижние колонки, и вторая последовательность определяет эти колонки.

В ходе одной DCS-последовательности может быть запрограммировано произвольное число символов (до 94). Определения символов (S...S/S...S) отделяются друг от друга символом «;» (код 73).

При выключении питания и выполнении полного сброса загруженные символы теряются.

Чтобы очистить символы, введенные пользователем, достаточно ввести следующую последовательность:

DCS 1;1;2 { пробел @ ST

Они также очищаются, если осуществить одну из следующих операций:

- сброс питания;
- использование последовательности ESC с.
- выполнение операции «Начальные установки» в режиме меню.

14.3. Сиксельный формат для графики

На экран дисплея может быть загружено произвольное графическое содержимое (функция дополнительно загруженной программы, входящей, например, в состав графического протокола). Для этого в него следует послать DCS-последовательность такого вида:

DCS P1 ; P2 q s ... s ST

где

P1 — устанавливает плотность пикселя, которые принимает дисплей. Этот параметр необязательный и должен иметь значение 1 (код 61).

P2 — признак записи фона. Этот параметр имеет три различных значения:

0 или 2 — нулевые биты сикселя устанавливают на изображении пиксел с текущим цветом фона;

1 — нулевые биты сикселя не изменяют предыдущее значение пикселя экрана.

Символ q (код 161), следующий за P1 и P2, указывает на то, что это команда загрузки изображения.

s...s

— строка данных, закодированная в виде сикселей. Она содержит символы, в каждом из которых закодированы 6 пикселей (вертикальная матрица). Они кодируются в виде 6-разрядного двоичного числа, причем младший бит соответствует верхнему пикселу. Чтобы послать такое число, к нему следует прибавить код 77, в результате получаются символы с кодами от 77 («?») до 176 («~»). Такие символы и входят в состав строки данных.

Кроме данных, представленных сикселями, строка содержит управляющие коды, лежащие вне диапазона 77—176:

! (041)

— признак повторителя, за которым следует десятичное число. Символ, следующий за этим числом, повторяется столько раз, сколько определено.

(043)

— открывает последовательность выбора цвета (см. ниже). За ним следует номер цвета и дополнительные параметры.

\$ (044)

— указывает конец строки сикселей. Он перемещает текущую позицию вывода в начало строки (графический ВК). Последующие данные будут накладываться на уже выведенные в эту строку.

— (055)

— указывает конец строки сикселей. Он перемещает текущую позицию вывода в начало следующей строки сикселей (графическая «новая строка»).

; (073)

— разделяет числовые параметры в строке. Если перед или после разделителя нет параметра, то он полагается равным 0.

Символ ввода цвета «#» может выполнять две различных функции. Во-первых, он может установить текущий цвет выводимых пикселей (0—3), во-вторых, он может установить на каждый из этих четырех (условных) цветов конкретный цвет в цветовых координатах RGB (red, green, blue) или HLS (hue, lightness, saturation).

Если за символом «#» следует только одно число — но-

мер цвета, то задается цвет вывода последующих пикселей (0—3).

Если за символом «#» следуют несколько чисел, разделенных символом «;», то такая последовательность устанавливает действительный цвет для одного из условных цветов.

Числа идут в следующем порядке:

#Pc; Pц; Pх; Pу; Pz, где

Pц — выбор системы цветных координат;

Pс — номер цвета (0—3);

1 — HLS (оттенок, яркость, насыщенность);

2 — RGB (красный, зеленый, синий).

Смысл следующих параметров зависит от выбранной системы цветных координат (HLS или RGB):

Система HLS

Pх 0—360 (градусы) цветовой оттенок

Pу 0—100 (проценты) яркость

Pz 0—100 (проценты) насыщенность

Система RGB

Pх 0—100 (проценты) интенсивность красного

Pу 0—100 (проценты) интенсивность зеленого

Pz 0—100 (проценты) интенсивность синего

Реально по каждому цвету дисплей имеет 4 градации яркости, при этом от 0 до 24% — нулевая яркость, от 25 до 49% — первая градация, от 50 до 74% — вторая градация, 75% и выше — максимальная яркость.

Перед началом вывода экран очищается, вывод начинается с левого верхнего угла. Когда вывод доходит до конца строки, то последующая информация игнорируется, пока не придет символ \$ (графический ВК) или минус (новая графическая строка). Когда вывод доходит до конца экрана, то вся последующая информация игнорируется.

15. УСТАНОВКА ПАРАМЕТРОВ ДИСПЛЕЯ В РЕЖИМЕ МЕНЮ

В режиме меню оператор может устанавливать по своему усмотрению различные параметры дисплея, такие, как установки табуляторов, тип курсора, характер движения информации в рулоне и т. д.

Меню включает в себя шесть таблиц. Каждая таблица состоит из заголовка (названия) и полей с наименованиями функций. При использовании ПЗУ-182 в нижней строке ин-

дицируется режим вывода символов: вставка или наложение; поскольку в ПЗУ-181 реализован только режим наложения, то в этом случае индикация не осуществляется.

Вход в меню осуществляется по нажатию клавиши УСТ РЕЖИМА. При этом на экране дисплея выводится таблица с заголовком «Установка режимов». В правом нижнем углу таблицы индицируется время суток в часах, минутах и секундах, которое устанавливается при загрузке графического протокола равным системному времени, оно также может быть изменено в режиме меню оператором.

Поля таблиц выполняют определенные функции, такие, как «Очистка экрана» или «Сброс дисплея», или отражают текущий режим работы, как, например, «Нет автоповтора» — «Автоповтор есть». Движение по полям производится с помощью клавиш управления курсором, а выполнение функций или изменение режимов — нажатием клавиши ВВОД. Нажатие любой другой клавиши вызывает вывод сообщения в нижней строке экрана.

На экране может выводиться только одна таблица. Для перехода в любую другую необходимо в таблице «Установ-

ДИАГРАММА ПЕРЕХОДОВ МЕЖДУ ТАБЛИЦАМИ МЕНЮ



ка режимов» выбрать поле с названием нужной таблицы и нажать клавишу ВВОД. Из любой таблицы, нажав клавишу ВВОД при выбранном поле «В главную таблицу» или «К следующей таблице», можно перейти либо в таблицу «Установка режимов», либо в следующую, согласно приведенной ниже диаграмме.

Для выхода из режима меню необходимо перейти на поле «Выход» и нажать ВВОД или повторно нажать клавишу УСТ РЕЖИМА.

15.1. Главная таблица установки режимов

ПЗУ-181:

УСТАНОВКА РЕЖИМОВ			ДВК-3,4 KeyGP V2.2	
Дисплей	Общие функции	Клавиатура	Табуляция	Цвет
На ЭВМ	Очистка экрана	Сброс дисплея	Графика 400 точек	
Восстановить	Сохранить	Начальные установки	Выход	

ПЗУ-182:

УСТАНОВКА РЕЖИМОВ			ДВК-3,4 KeyGP V2.2	
Дисплей	Общие функции	Клавиатура	Табуляция	Цвет
На ЭВМ	Очистка экрана	Сброс дисплея	Графика 400 точек	
Восстановить	Сохранить	Начальные установки	Выход	ЧЧ:ММ:СС

Режим наложения

Дисплей

— переход в таблицу «Установка дисплея»;

Общие функции

— переход в таблицу «Общие функции»;

Клавиатура

— переход в таблицу «Режимы клавиатуры»;

Табуляция

— переход в таблицу «Установка табуляций»;

Цвет

— переход в таблицу «Установка цветов»;

На ЭВМ

— режим связи с ЭВМ (нормальный режим работы);

Автономно

— работа в автономном режиме, при этом дисплей не посылает информацию в ЭВМ и принимает ее только с собственной клавиатуры;

Очистка экрана

— при нажатии ВВОД выполняется очистка экрана дисплея;

Сброс дисплея

— при нажатии ВВОД выполняется частичный сброс;

Графика 400 точек

— установлен режим для графики 400×240 16 цветов (начальное значение);

Графика 800 точек

— установлен режим для графики 800×240 4 цвета;

Эмуляция LOAD

— установлен режим эмуляции графических функций программы LOAD;

Восстановить

— при нажатии ВВОД все значения, изменяемые посредством меню, ранее сохраненные по операции «Сохранить», восстанавливаются из памяти КЦГД;

Сохранить

— при нажатии ВВОД запоминаются все режимы, которые устанавливаются посредством меню, а также таблица цветов, автоответ и позиции табуляции. Эти значения хранятся в памяти КЦГД и при выключении машины теряются. Если требуется сохранить эти значения для последующей работы, то после выхода из меню следует подать команду монитора R KEYGP, по которой эти установки будут сохранены в файле KEYGP.SAV. Значения режимов, сохраненные в файле, автоматически восстанавливаются при загрузке программы графического протокола.

Начальные установки

— при нажатии ВВОД выполняется полный сброс всех установок дисплея в начальное состояние;

Выход

— при нажатии ВВОД — выход из меню с сохранением

установленных параметров. То же происходит, если при любом выбранном поле нажать УСТ РЕЖИМА; ЧЧ:ММ:СС

— текущее время. Оно корректируется нажатием клавиши ВВОД. Для перехода на поля ЧЧ (часы), ММ (минуты), СС (секунды) нажимайте клавиши «стрелка вправо» или «стрелка влево».

15.2. Таблица установки режимов дисплея

УСТАНОВКА ДИСПЛЕЯ				ДВК-3,4 KeyGP V2.2
К следующей таблице	В главную таблицу	80 колонок	Символ \$	
Есть автоперевод	Скачкообразный рулон	Темный экран, светлый текст		
Курсор	Курсор — блок	Отработка управляющих символов		

Режим наложения

К следующей таблице

— при нажатии ВВОД — переход в таблицу «Общие функции»;

В главную таблицу

— при нажатии ВВОД — переход в таблицу «Установка режимов»;

80 колонок

— установлен режим 80 символов в строке (начальное значение). При нажатии ВВОД изменяется на 132 символа в строке;

132 колонки

— установлен режим 132 символа в строке. При нажатии ВВОД изменяется на 80 символов в строке;

Символ ☒ или Символ \$

— отображение символа с кодом 44 (восьм.) как ☒ (начальное значение) или как \$;

Есть автоперевод

— установлен режим автоперевода на новую строку (стандартный режим);

Нет автоперевода

— режим автоперевода курсора из последней колонки в начало следующей строки при приеме символа вы-

ключен. Принимаемые символы будут выводиться в последнюю позицию в строке, заменяя последний выведенный символ;

Скачкообразный рулон

— установлен режим скачкообразного движения информации в зоне рулона (начальное значение);

Плавный рулон

— установлен режим плавного движения информации в зоне рулона;

Темный экран, светлый текст

— установлен нормальный экран: светлый текст, темный фон (стандартный режим);

Светлый экран, темный текст

— установлен инверсный экран: светлый фон, темный текст;

Курсор

— индикация курсора на экране включена (стандартный режим);

Нет курсора

— индикация курсора на экране выключена;

Курсор — блок

— курсор отображается в виде полного мигающего знака (начальное значение);

Курсор — подчерк

— курсор отображается в виде подчеркика;

Отработка управляющих символов

— дисплей обрабатывает управляющие символы (стандартный режим);

Отображение управляющих символов

— в этом режиме дисплей управляющие символы не обрабатывает, а отображает их на экране в виде символов дополнительного цвета. Коду 1 соответствует буква А, коду 2 буква В и так далее. Символ ESC изображается в виде символа \$. Символы кодовой таблицы С1 изображаются подчеркнутыми буквами дополнительного цвета. Символы HT, CR, LF, FF, VT изображаются в виде псевдографических символов HT, CR, LF, FF, VT дополнительного цвета, причем после приема LF, FF или VT происходит переход на новую строку. Печатные символы изображаются нормальным цветом.

15.3. Таблица установки общих функций

ОБЩИЕ ФУНКЦИИ		ДВК-3,4 KeyGP V2.2	
К следующей таблице	В главную таблицу	VT52	LF, VT, FF=LF
Ответ=		Автоответ виден	

Режим наложения

К следующей таблице

— при нажатии ВВОД — переход в таблицу «Установка клавиатуры»;

В главную таблицу

— при нажатии ВВОД — переход в таблицу «Установка режимов»;

VT52

— дисплей работает в режиме совместимости с VT52 и КСМ (начальное значение);

VT100

— дисплей работает в режиме VT100;

VT200, 7-бит

— дисплей работает в режиме VT200 с семибитной передачей данных;

VT200, 8-бит

— дисплей работает в режиме VT200 с восьмибитной передачей данных;

Примечание. В версии, работающей на ПЗУ-181, режим VT200 не поддерживается.

LF,VT,FF=LF

— по приему кода LF, VT, FF дисплей отрабатывает операцию перевод строки (стандартный режим);

LF,VT,FF=CR+LF

— по приему кода LF, VT, FF дисплей отрабатывает операцию новая строка — курсор на начало следующей строки, а клавиши BK и ВВОД посылают не один символ CR, а два символа CR+LF.

Ответ=

— ввод сообщения автоответа, который по принятию дисплеем кода ENQ (5) будет посылаться в ЭВМ. Для ввода автоответа необходимо нажать ВВОД, затем ввести текст, и снова ВВОД;

Автоответ виден

— этот параметр устанавливается при вводе автоответа с клавиатуры (начальное значение);

Автоответ не виден

— при нажатии ВВОД автоответ становится невидимым в поле Ответ=, но сохраняется в памяти.

15.4. Таблица установки режимов клавиатуры

ПЗУ-181:

РЕЖИМЫ КЛАВИАТУРЫ		ДВК-3,4 KeyGP V2.2	
К следующей таблице	В главную таблицу	Символы всех регистров	
Функциональная клавиатура		Коды для управления курсором	
Обычный режим клавиш верхнего ряда			

ПЗУ-182:

РЕЖИМЫ КЛАВИАТУРЫ		ДВК-3,4 KeyGP V2.2	
К следующей таблице	В главную таблицу	Автоповтор	
Нет эха	Функциональная клавиатура	Коды для управления курсором	
Символы всех регистров		Программирование клавиш разрешено	

Режим наложения

К следующей таблице

— при нажатии ВВОД происходит переход в таблицу «Установка табуляций»;

В главную таблицу

— при нажатии ВВОД происходит переход в таблицу «Установка режимов»;

Автоповтор

— режим автоповтора клавиатуры включен. При длительном нажатии клавиши дисплей будет повторять посылку ее кода с частотой 30 раз в секунду (начальное значение);

Нет автоповтора

— режим автоповтора клавиатуры выключен. При длительном нажатии клавиши дисплей будет посылать ее код только один раз;

Нет эха

— режим локального эха выключен. Информация, идущая с клавиатуры, будет посылаться в ЭВМ, не отображаясь на экране (стандартный режим);

Эхо

— режим локального эха включен. Информация, посылаемая с клавиатуры, будет также отображаться на экране, как если бы она была принята из ЭВМ. Обычно прикладные программы и операционная система сами выполняют эхо-печать;

Цифровая клавиатура

— дополнительная клавиатура выдает цифры (начальное значение);

Функциональная клавиатура

— дополнительная клавиатура выдает специальные ESC-последовательности;

Коды для управления курсором

— клавиши управления курсором выдают последовательности, являющиеся командами перемещения курсора (начальное значение);

Коды для прикладного ПО

— клавиши управления курсором выдают последовательности, не являющиеся командами перемещения курсора (только для режима ANSI);

Символы всех регистров

— вывод символов всех регистров — большие латинские, маленькие латинские или большие русские, маленькие русские (стандартный режим);

Буквы только верхнего регистра

— включен режим сокращенного набора кода КОИ-7 (только большие латинские и большие русские буквы);

Программирование клавиш разрешено

— программирование клавиш верхнего ряда клавиатуры в режиме VT200 разрешено (начальное значение);

Программирование клавиш запрещено

— загрузка и переопределение клавиш запрещена.

Обычный режим клавиш верхнего ряда

— в режиме ANSI работают только клавиши AP2, ВШ, ПС.

Особый режим клавиш верхнего ряда

— в режиме ANSI клавиши верхнего ряда выдают специальные последовательности;

15.5. Таблица установки позиций табуляторов

УСТАНОВКА ТАБУЛЯЦИИ				ДВК-3,4 KeyGP V2.2			
К следующей таблице	В главную таблицу	Очистить все	Через 8 колонок	Т	Т	Т	Т
Т	Т	Т	Т	Т	Т	Т	Т

123456789012345678901234567890 123456789012345678901234567890 1234567

Режим наложения

К следующей таблице

— при нажатии ВВОД происходит переход в таблицу «Установка цветов»;

В главную таблицу

— при нажатии ВВОД происходит переход в таблицу «Установка режимов»;

Очистить все

— при нажатии ВВОД происходит сброс всех табуляторов;

Через 8 колонок

— при нажатии ВВОД происходит установка табуляторов на 8 колонок (начальное значение);

Пользователь может устанавливать табуляторы по своему усмотрению, перейдя на поле табуляторов при помощи клавиш управления курсором, далее двигаясь по нему и устанавливая или отменяя табуляторы нажатием клавиши ВВОД.

15.6. Таблица установки цветов

УСТАНОВКА ЦВЕТОВ				ДВК-3,4 KeyGP V2.2			
К следующей таблице	В главную таблицу	Начальные установки	Мерцание	0000	1300	2030	3003
4111	5330	6303	7033	8222	9022	101201	10021220013202

Режим наложения

К следующей таблице

- при нажатии ВВОД происходит переход в таблицу «Установка дисплея»;

В главную таблицу

- при нажатии ВВОД происходит переход в таблицу «Установка режимов»;

Начальные установки

- при нажатии ВВОД происходит установка начальных значений цветов, таких же, какие устанавливаются по включению питания или полному сбросу дисплея;

Мерцание

- пиксели с индексом цвета 15 выводятся мерцающими
- их цвет попеременно изменяется между пятым и десятым цветом;

Нет мерцания

- пиксели с индексом цвета 15 изображаются неизменным цветом.

Под этими полями отображаются две строки, в первой приведены номера и значения цветов (в терминах градаций красной, зеленой и синей составляющих цветов), во второй — их изображения для визуального контроля. Для установки цвета необходимо нажать стрелку вниз, клавишами ← и → перейти на значение одной из составляющих нужного цвета и нажатиями клавиши ВВОД установить требуемое значение. При изменении значения цветное поле под ним сразу же принимает соответствующий цвет, что позволяет проконтролировать установку. Установить значение цвета с индексом 15 можно лишь тогда, когда мерцание его выключено.

ГРАФИЧЕСКИЙ ПРОТОКОЛ KeyGP ДЛЯ КЦГД. РУКОВОДСТВО ПРОГРАММИСТА

1. ВВЕДЕНИЕ

Контроллер цветного графического дисплея КЦГД ЭВМ «Электроника МС0507, МС0504», может быть использован, в зависимости от его управляющей программы, в качестве алфавитно-цифрового дисплея, а также как графический дисплей. Управляющая программа, реализующая алфавитно-цифровые функции, размещается в БИС ПЗУ типа КР1801РЕ2-181 или -182.

Графические функции реализуются либо путем установки дополнительных ПЗУ, либо загрузкой дополнительного программного обеспечения в ОЗУ КЦГД из ЭВМ. Дополнительное программное обеспечение, совместно с программами, записанными в ПЗУ, обеспечивает расширение алфавитно-цифровых функций и вводит новые, графические. По своим алфавитно-цифровым функциям КЦГД соответствует широко распространенным дисплеям фирмы Digital Equipment Corp., США, и реализует системы команд, совместимые с системами команд дисплеев VT52, VT100, VT200. Характерной чертой этих систем команд является использование так называемых ESC-последовательностей для выполнения управляющих или редакторских функций. Эти последовательности начинаются со специального символа ESC, за которым следуют символы команды, не отображаемые дисплеем как обычные. Совокупность обрабатываемых дисплеем управляющих символов и управляющих последовательностей называется системой команд или протоколом. Алфавитно-цифровой протокол управляющей программы КЦГД описан в первой главе данной книги.

В данной главе описан набор графических команд — графический протокол KeyGP. Команды графического протокола, как и команды алфавитно-цифрового протокола, имеют вид специальных ESC-последовательностей. Для реализации

набора графических команд, в КЦГД загружается специальная программа, которая подключается к программе, записанной в ПЗУ, и при разборе управляющих ESC- и DCS-последовательностей распознает и выполняет графические команды, команды же алфавитно-цифрового протокола выполняются программой, записанной в ПЗУ.

Программа графического протокола реализует следующие функции:

- построение отрезков и точек;
- построение окружностей, эллипсов и их дуг, секторов и сегментов, в том числе с заполнением заданным узором или штриховкой;
- построение ломаных линий и замкнутых многоугольников, в том числе заполненных узором или штриховкой;
- построение «плавных» разомкнутых и замкнутых кривых, в том числе заполненных узором или штриховкой;
- «быстрое» построение прямоугольников, в том числе заполненных;
- рисование символов различного размера и ориентации, а также вывод текста из таких символов;
- определение пользовательской прорисовки символов;
- вывод массивов точек из ЭВМ в ЗУ КЦГД;
- вывод массивов точек из ЗУ КЦГД в ЭВМ;
- масштабирование изображения;
- установка области вывода на экран;
- отключение масштабирования изображений и отсечения векторов по границам области вывода;
- установка атрибутов для графических примитивов — цвет, тип линий, представление границы и т. д.;
- управление режимами вывода;
- управление представлением цвета (яркости) на экране ЭЛТ;
- заполнение внутренне-определенных и гранично-определенных областей узором или штриховкой;
- построение заполненных сложных фигур, контур которых формируется из приведенных выше примитивов;
- построение маркеров и множества маркеров («полимаркеров»);
- вывод графического курсора в текущей позиции;
- выдача из дисплея в ЭВМ мировых или экранных координат текущей позиции;

— указание точки с помощью графического локатора — графического курсора, перемещаемого с помощью клавиш-стрелок или манипулятора «мышь», и передача в ЭВМ мировых или экранных координат графического локатора.

По набору выводимых графических примитивов и их параметров графический протокол КЦГД соответствует стандарту CGI (Computer Graphics Interface) для двумерной машинной графики, который может быть реализован со стороны ЭВМ, так же как и ядро графической системы GKS (Graphical Kernel System).

Для связи прикладных программ с загруженной в КЦГД программой, реализующей графический протокол, разработаны интерфейсные объективные модули PGPACK, FGPACK и CGPACK, с помощью которых обеспечивается возможность графического вывода из языков программирования соответственно Паскаль, Фортран, Си. Интерфейсные модули содержат точки входа, к которым прикладная программа обращается, как к подпрограммам, передавая требуемые аргументы. Интерфейсный модуль преобразует эти аргументы в последовательности символов по правилам графического протокола и посылает их в КЦГД, обеспечивая таким образом выполнение графических команд. Передача осуществляется через системный запрос вывода на терминал — TTYOUT. Различия в кодах модулей связаны только с особенностями передачи параметров в различных языках программирования, функционально же они идентичны. Программы интерфейсных модулей написаны на макроассемблере и занимают небольшой объем (около 2 Кбайт) в ОЗУ ЭВМ. Для программирования на макроассемблере имеется макробibliothek GMACRO.MLB, которая позволяет обращаться к подпрограммам интерфейсного модуля PGPACK.

Вышеприведенные модули представляют собой единый объектный модуль (не библиотеку), в котором содержатся ВСЕ доступные подпрограммы пакета. Небольшой объем модулей, и, соответственно, файлов, в которых они находятся, оправдывает такой подход. Однако, если программа использует небольшое подмножество подпрограмм пакета, удобнее использовать библиотеки, что дает уменьшение объема прикладной программы. Файлы библиотек имеют имена PGLIB, FGLIB, CGLIB. Минимально необходимое «ядро» библиотеки, которое всегда подключается к программе, занимает около 0.5 Кбайт.

2. ОБЩИЕ ПОНЯТИЯ И ТЕРМИНЫ

Для выполнения графических построений протокол содержит команды, обеспечивающие вывод на экран отрезков, точек, окружностей, дуг, символов. Любое сложное изображение можно представить в виде некоторого набора этих простых элементов изображения, которые обычно называются примитивами.

КЦГД является растровым графическим дисплеем, то есть в нем графические изображения и текстовые символы строятся на экране путем задания цвета отдельных точек раstra или пикселей (picture element) в графической памяти. На экране высвечивается 240 строк, в которых произвольным образом могут чередоваться группы из 8 пикселей высокого разрешения, либо из 4 пикселей низкого разрешения. Всего в строке выводится 100 таких групп, то есть либо 800 пикселей высокого разрешения, либо 400 пикселей низкого разрешения. Режим повышенного разрешения позволяет строить изображения из точек меньшего размера, однако пиксели, выведенные в этом режиме, отображаются только четырьмя цветами. В режиме низкого разрешения каждый пиксел выводится одним из 16 цветов, но точки имеют больший размер. Посредством аппаратно реализованной таблицы цветности каждый пиксел, в зависимости от его значения, отображается определенным цветом, который может быть изменен программно.

Изображение может записываться в графическую память различным образом — с наложением, очисткой разрядов, замещением их или инверсией. В режиме замещения код цвета, которым производится построение, замещает прежний код пикселя (цвет, которым он отображается на экране). В режиме очистки разрядов единичные биты нового кода обнуляют соответствующие биты старого кода. В режиме наложения выполняется операция ИЛИ над новым и старым цветом. В режиме инверсии над старым и новым цветом выполняется операция ИСКЛЮЧАЮЩЕЕ ИЛИ.

Программа графического протокола, загружаемая в память КЦГД, обеспечивает два режима работы — высокого и низкого разрешения. Режим устанавливается посредством меню, в котором также могут быть изменены другие режимы дисплея. В режиме высокого разрешения можно произвольным образом выводить на экран алфавитно-цифровую и графическую информацию в четырех цветах. В режиме низкого разрешения информацию можно выводить в 16 цветах, одна-

ко необходимо следить за тем, чтобы текстовая и графическая информация не выводилась в одно и то же место. В противном случае в месте наложения возникают искажения информации, связанные с тем, что 4 или 8 пикселей группы в месте наложения могут получать новый признак режима разрешения и цвета будут формироваться по другим правилам. Так как в режиме высокого разрешения изображения содержат больше пикселей, то они строятся несколько медленнее.

Для указания расположения примитивов на экране используется прямоугольная система координат.

Прикладная программа строит графические изображения в так называемых мировых координатах, которые могут принимать целочисленные значения от -32768 до $+32767$. На экране же дисплея изображение строится только в пределах экрана в терминах экранных координат, где единица масштаба соответствует одному пикселу. Отсчет экранных координат идет от левого нижнего угла экрана, координата X изменяется по горизонтали от 0 до 399 в режиме низкого разрешения и от 0 до 799 в режиме высокого разрешения. Координата Y изменяется по вертикали вверх от 0 до 239. Для установления соответствия между мировыми и экранными координатами используются понятия окна и области вывода. Окно (window) — это некоторый прямоугольник в мировых координатах, который определяет диапазон значений координат прикладной программы, а область вывода (viewport) — это прямоугольник в экранных координатах, на который отображается окно, то есть место фактического вывода графических построений. Из этого следует, что при выводе происходит смещение координат и масштабирование, причем масштаб по осям X и Y может различаться. Область вывода не может выходить за границы экрана, а границы окна могут иметь любые целочисленные значения, с тем ограничением, что размер окна не должен превышать 32767.

В процессе построения на экране графических примитивов происходит изменение местоположения текущего вывода. С этим местоположением обычно связана так называемая текущая позиция. Имеющиеся графические команды позволяют учитывать или не учитывать эту текущую позицию. Для большинства примитивов предусмотрено два формата — с заданными абсолютными координатами и с заданными смещениями от текущей позиции. Смещения передаются в виде 9-разрядных чисел со знаком, таким образом, возможная величина смещения лежит в диапазоне от -256 до $+255$. Например, при использовании абсолютных координат для по-

строения отрезка необходимо указать его начало и конец, а при использовании смещений началом отрезка считается текущая позиция и достаточно указать смещение по X и по Y до конечной точки. После выполнения каждой команды текущая позиция устанавливается в конечную. Таким образом можно легко строить ломаные линии, последовательно задавая координаты вершин. Заметим, что протокол включает и специальную операцию построения ломаных.

Для удобства изложения далее по тексту будут использованы мнемонические названия команд, отражающие выполняемую функцию и совпадающие с именами точек входа интерфейсных модулей. Имена команд, в которых передаются не абсолютные мировые координаты, а значения смещений относительно текущей позиции, начинаются обычно с буквы R (relative), например RDOT, RLINE, в отличие от DOT, LINE и т. п. При описании команд будет даваться краткое объяснение и взаимосвязь с другими командами и действующими атрибутами. Кроме этого, будет даваться формат команды графического протокола и объявление соответствующей подпрограммы для языков программирования с описанием параметров подпрограммы.

3. КОМАНДЫ ГРАФИЧЕСКОГО ПРОТОКОЛА

Расширения управляющей программы КЦГД, реализующие графический протокол, загружаются в КЦГД из специальной программы KeyGP. Загруженная программа графического протокола воспринимает два вида графических команд. В режиме VT52 (в котором КЦГД по алфавитно-цифровым функциям совместим с КСМ ДВК-3 и VT52) графическая команда начинается с символов ESC M (коды 33 115), за которыми следует символ команды и аргументы, а в режиме ANSI (или режим VT100) команды передаются в составе DCS-последовательности. DCS-последовательность (Device Control String) — это последовательность кодов, начинающаяся с символов ESC P (коды 33 120) или с символа DCS (код 220, допустим только в режиме VT200 8 бит), за которыми идет символ M, указывающий на то, что далее идет графическая команда, и затем следует произвольное число графических команд со своими аргументами. Закачивается DCS-последовательность символами ESC \ (коды 33 134) или символом ST (String Terminator) с кодом 234, допустимым только в режиме VT 200 8 бит. DCS-последовательность может быть прервана также любой последова-

тельностью, начинающейся с ESC, и символами SUB и CAN с кодами 32 и 30. Подробное описание формата управляющих последовательностей и особенностей режимов VT52 и ANSI содержится в документации на управляющую программу алфавитно-цифрового дисплея.

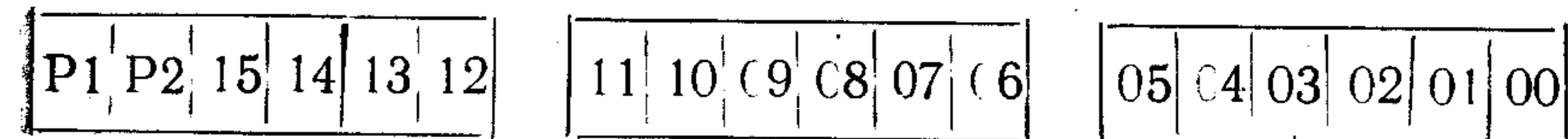
В режиме VT52 для выполнения команд достаточно выполнять обращения к соответствующим подпрограммам. В режиме ANSI для инициирования последовательности графических команд необходимо вызвать подпрограмму DCSOPEN, имеющуюся в интерфейсных модулях, которая посылает ESC P M, а для завершения последовательности надо обратиться к подпрограмме DCSCLOSE, которая посылает ESC \. Только после вызова подпрограммы DCSCLOSE или приема любой управляющей последовательности, начинающейся с символа ESC, дисплей будет воспринимать печатные символы. Для уменьшения количества обменов между ЭВМ и КЦГД рекомендуется использовать режим ANSI, так как при этом не происходит посылки кодов ESC M на каждую графическую команду, что несколько повышает скорость вывода графической информации, однако при этом необходимо быть уверенным в том, что в DCS-последовательность, внутри которой производится вывод графической информации, не попадет алфавитно-цифровой вывод, например, сообщения об ошибках переполнения или открытия файлов.

Примечание 1. Для установки нужного режима работы дисплея в интерфейсном пакете имеются две процедуры — MODE52 (режим VT52) и MODE100 (режим ANSI), которые будут описаны в разделе «Подпрограммы установки режимов дисплея».

Примечание 2. Если включен режим ANSI, то программы интерфейсного пакета будут автоматически вызывать процедуру DCSOPEN (начало DCS-последовательности графических команд), если до того она не была вызвана. Таким образом, обычно вызов DCSOPEN не требуется, кроме того случая, когда DCS-последовательность была прервана управляющей ESC-последовательностью, посланной из прикладной программы, а не из процедур интерфейсных пакетов.

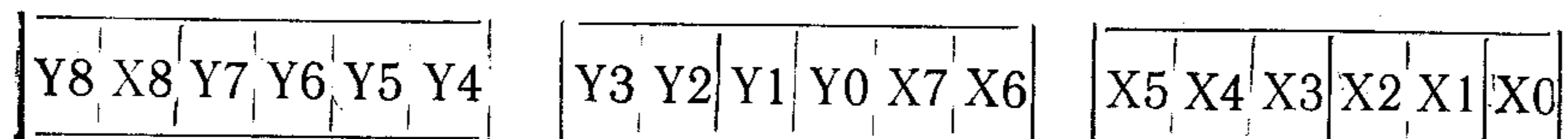
Графическая команда состоит из символа, определяющего команду, и некоторого количества символов, задающих значения параметров команды — координаты, цвет, и т. д. Параметры графических команд в ограниченном диапазоне значений (номер цвета и т. п.) передаются в виде одного символа, код которого является значением параметра плюс

40восьм. Другие аргументы, например, координаты, заданные как 16-разрядные числа, кодируются тремя символами. 16 разрядов разбиваются на три группы таким образом, что первая группа содержит 4 старших бита, вторая — следующие 6 бит, третья — младшие 6 бит. К первой группе могут быть добавлены еще два бита о дополнительной информации, имеющей специфическое значение для каждой конкретной команды графического протокола. Биты слова передаются в следующем порядке (слева направо):



P1, P2 — биты с дополнительной информацией. Полученные двоичные числа складываются с числом 40 и передаются в виде трех печатных символов с кодами от 40 до 137.

Координаты, передаваемые в виде смещений от текущей позиции, передаются в виде трех символов. Биты 9-разрядных величин смещений разбиваются на три группы в следующем порядке (слева направо):



Полученные двоичные числа складываются с числом 40 и передаются в виде печатных символов с кодами от 40 до 137.

В дальнейшем изложении будет принят следующий способ описания формата команд. Первым дается символ команды и его восьмеричный эквивалент в скобках, за которым следуют параметры. Параметр, передаваемый в виде одного байта, записывается с индексом 1, например, COLOR₁. Параметр, передаваемый в виде трех символов, записывается с индексом 3, например, X₃. Двухбитовая информация, передаваемая через дополнительные биты трехсимвольного представления параметра, записывается через знак + и с индексом 2, например, N₃+POLYSET₂. Если для передачи четырехбитового параметра (обычно это код цвета) объединяются двухбитовые поля двух следующих друг за другом трехсимвольных параметров, то они записываются в скобках, а четырехбитовый параметр — с индексом 4, например, (X₃, Y₃)+COLOR₄. Два 9-битовых смещения, передаваемые в виде группы из трех символов, записываются через запятую в скобках с добавлением индекса 3, например (DX, DY)₃.

4. АТТРИБУТЫ ГРАФИЧЕСКИХ ПРИМИТИВОВ

При построении графических примитивов получающиеся изображения могут видоизменяться в зависимости от установленных атрибутов.

Для выделения графических примитивов самым существенным атрибутом является цвет, которым они рисуются на экране.

Отрезки, окружности, эллипсы и их дуги могут выводиться различными типами линий: сплошными, штриховыми и т. д., в том числе и определенными пользователем.

Любые замкнутые фигуры — прямоугольники, окружности, эллипсы, их секторы и сегменты, замкнутые многоугольники — во время или после построения могут быть заполнены одним из имеющихся видов штриховки или узором, заданным пользователем. При построении заполненных принимается во внимание еще один атрибут — представление границы. Эти фигуры могут рисоваться как закрытые — с границей, нарисованной замкнутой линией, так и открытые — без рисования границы.

Выводимые на экран символы графического текста могут быть нарисованы с различной ориентацией, наклоном, разных размеров. Текстовые строки могут иметь различный просвет между буквами и произвольное направление написания.

Маркеры, а также графические курсор и локатор, выводимые соответствующими графическими командами, могут иметь различный вид, размер (определяется коэффициентом увеличения) и цвет.

При построении графических примитивов используется также текущий режим вывода, который определяет, каким образом будет взаимодействовать записываемый цвет с фоном, на который производится вывод.

5. УСТАНОВКА РЕЖИМА РАЗРЕШЕНИЯ

Как уже отмечалось, построение изображения может производиться в двух режимах: высокого и низкого разрешения.

Необходимый режим (высокой или низкой плотности) может быть установлен посредством меню установки режимов дисплея или непосредственно из прикладной программы командой SETDENSITY. Если ее аргумент DENSITY=0, то,

устанавливается низкая плотность, в противном случае устанавливается высокая плотность.

Формат команды:

SETDENSITY : e (145) DENSITY₁

Для использования этой команды в программы на языке Паскаль должно быть такое объявление:

```
PROCEDURE SETDENSITY(DENSITY:INTEGER); EXTERNAL;
```

Из программы на языке Паскаль процедура вызывается оператором:

```
SETDENSITY(DENSITY);
```

На языке Фортран:

```
CALL SETDEN(IDENS)
```

На языке Си:

```
int density;
setdensity(density);
```

6. КОМАНДЫ ЗАДАНИЯ ОКНА И ОБЛАСТИ ВЫВОДА, УПРАВЛЕНИЯ МАСШТАБИРОВАНИЕМ И ОТСЕЧЕНИЕМ

Расположение примитивов задается координатами X, Y в двумерной декартовой системе координат. Координата X задает положение по горизонтали, координата Y — по вертикали. Начало координат расположено в левом нижнем углу экрана, ему соответствуют координаты (0, 0), верхняя правая точка имеет координаты (399, 239) в режиме низкого разрешения и (799, 239) в режиме высокого разрешения. Графический вывод ограничивается пределами так называемой области вывода, которая первоначально назначается на полный экран. Все элементы изображения, выходящие за ее пределы, «отсекаются», то есть либо не видны вовсе, либо видны только те их части, которые попадают в границы области вывода, которые можно установить произвольным образом в пределах экрана, но размер области по осям X и Y должен быть больше одной точки.

Границы области вывода XVLEFT (левая), YVBOTTOM (нижняя), XVRIGHT (правая), YVTOP (верхняя) задаются командой графического протокола VIEWPORT, с целочисленными аргументами из диапазона 0..399 (0..799) и 0..239 по X и Y соответственно. Если заданы некорректные значения границ, то устанавливаются значение 0 для левой или нижней границы, и 399(799) или 239 для правой или верхней границы соответственно.

Как уже было сказано, прикладная программа строит изображение не в экранных координатах, а в мировых. Команда WINDOW устанавливает границы окна XWLEFT (левая), YWBOTTOM (нижняя), XWRIGHT (правая), YWTOP (верхняя) в пространстве мировых координат. Это окно будет отображаться путем соответствующего пересчета координат на установленную область вывода. Координаты границ окна могут иметь любые значения в диапазоне -32768..+32767 по X и Y, однако размер окна (разность между соответствующими границами) не должен превышать 32767.

При задании поля вывода и окна определяются коэффициенты преобразования и пределы. Кроме того, при задании поля вывода текущая точка устанавливается в его левый нижний угол. Никаких других действий не производится.

На рисунке показано отсечение рисуемых примитивов границами установленной области вывода:



Получение экранных координат производится по формулам:

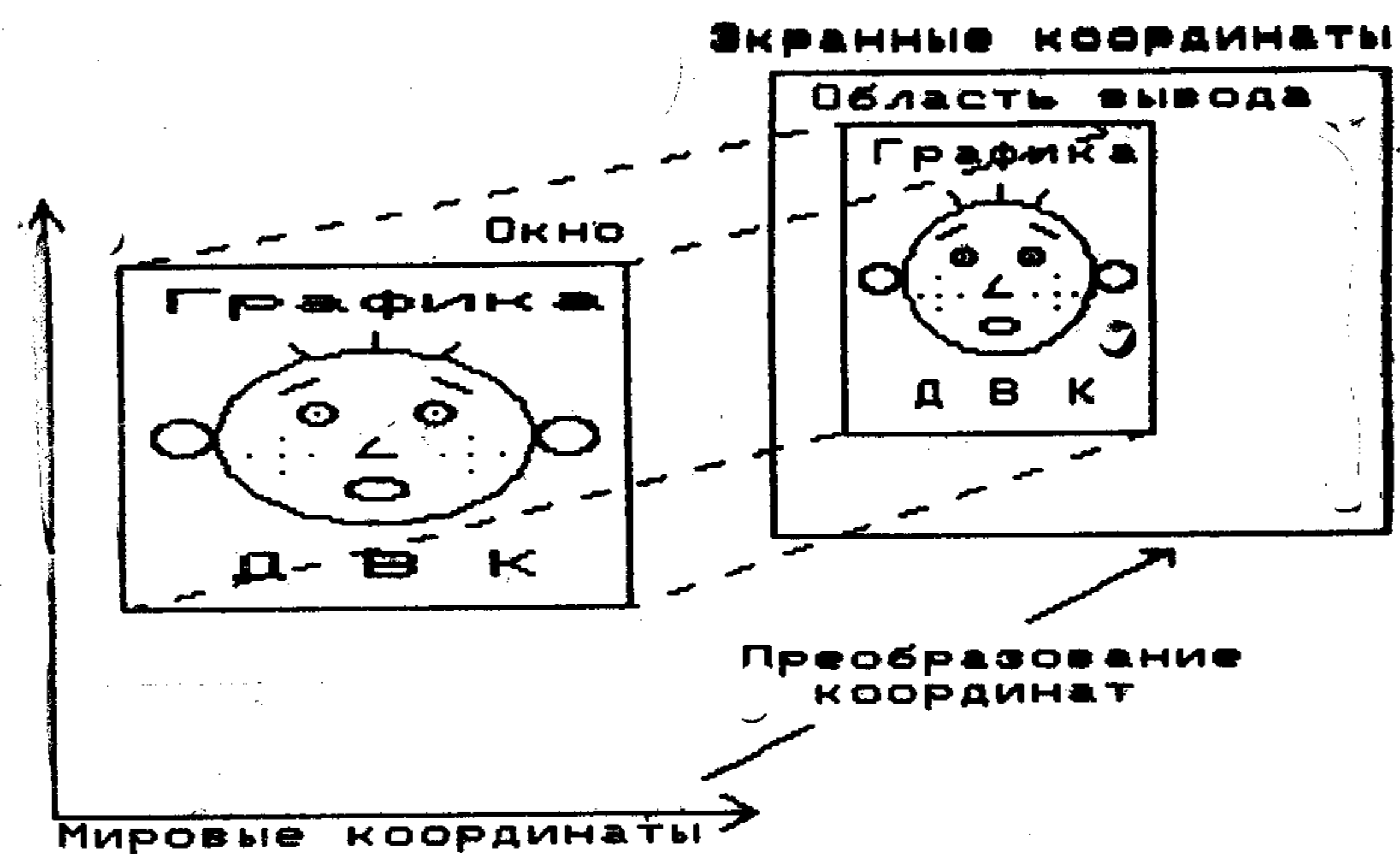
$$XV = XVLEFT + \frac{(XW - XWLEFT) (XVRIGHT - XVLEFT + 1)}{XWRIGHT - XWLEFT + 1} \quad (XWRIGHT > XWLEFT)$$

$$XV = XVLEFT + \frac{(XW - XWLEFT) (XVRIGHT - XVLEFT + 1)}{XWRIGHT - XWLEFT - 1} \quad (XWRIGHT < XWLEFT)$$

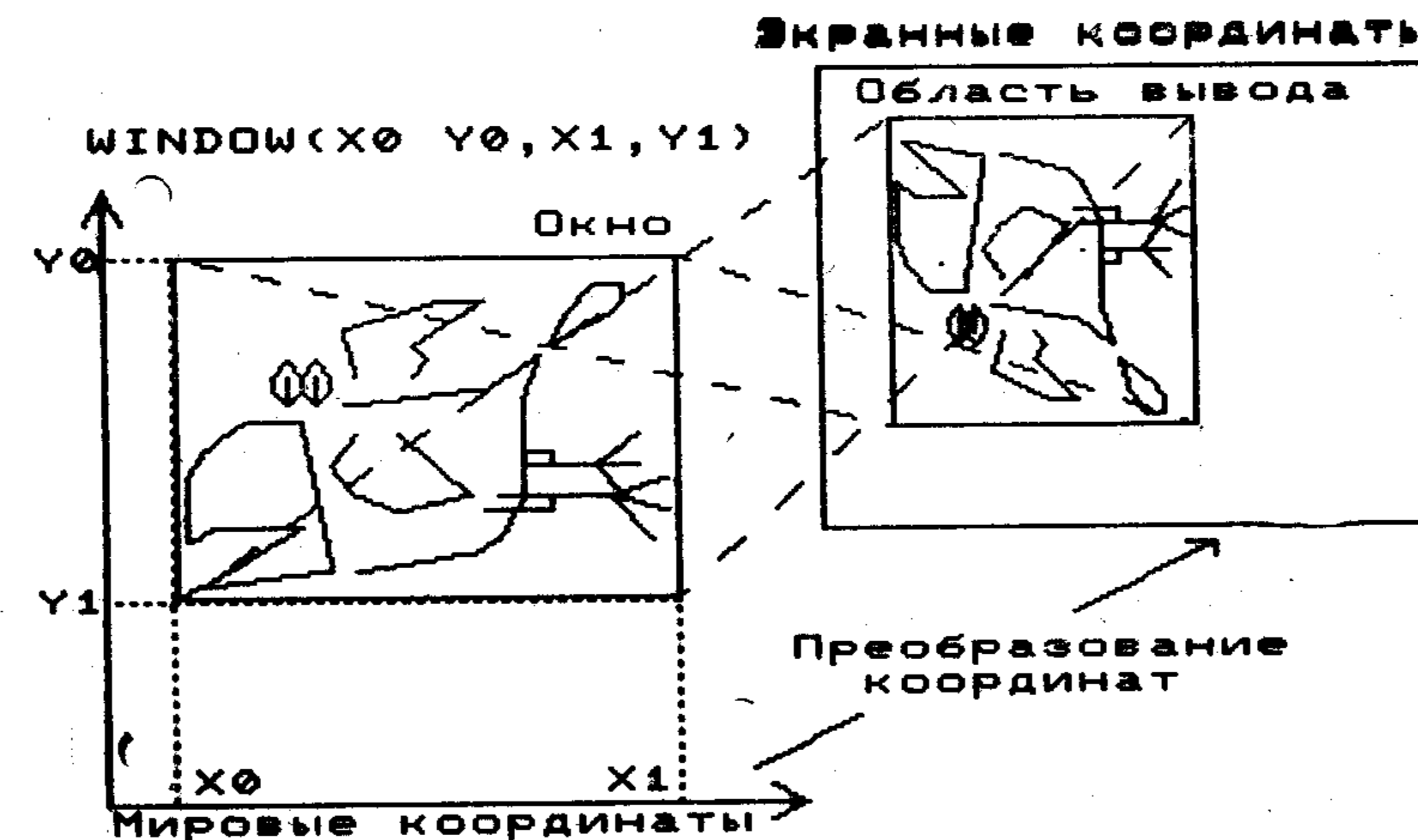
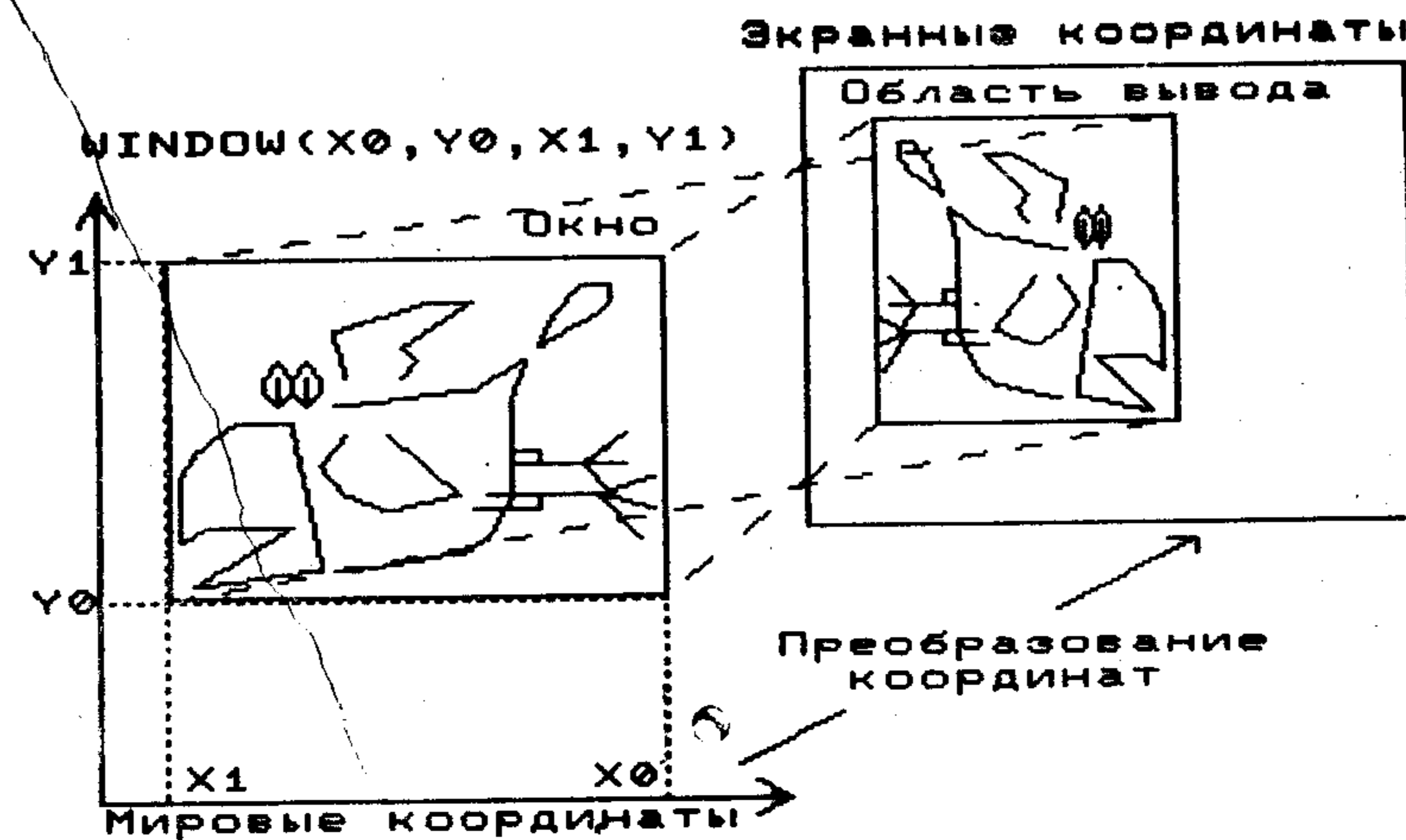
$$YV = YVBOTTOM + \frac{(YW - YWBOTTOM) (YVTOP - YVBOTTOM + 1)}{YWTOP - YWBOTTOM + 1} \quad (YWTOP > YWBOTTOM)$$

$$YV = YVBOTTOM + \frac{(YW - YWBOTTOM) (YVTOP - YVBOTTOM + 1)}{YWTOP - YWBOTTOM - 1} \quad (YWTOP < YWBOTTOM)$$

Пример отображения мировых координат в экранные приведен на рисунке:



При $XWLEFT > XWRIGHT$ изображение зеркально переворачивается относительно оси Y , при этом отсчет мировых координат будет идти справа налево. Аналогично, если $YWBOTTOM > YWTOP$, то отсчет мировых координат будет идти сверху вниз, а изображение переворачивается относительно оси X . Это свойство можно использовать, например, в тех случаях, когда адаптируются программы, первоначально разработанные для дисплея, у которого координата Y отсчитывается сверху вниз. При этом указывается $YWTOP = 0$, $YWBOTTOM = n$ (n — размер окна по вертикали). На следующих иллюстрациях показаны примеры изменения направления координат:



Некоторые прикладные программы могут сами выполнять преобразование координат и отсечение отрезков или никогда не строить изображение за пределами области вывода. В этом случае для увеличения скорости построения примитивов преобразование координат и отсечение может включаться и выключаться. Команда `MAPPING` управляет включением и выключением преобразования координат. Когда преобразование выключено, принимаемые протоколом координа-

ты интерпретируются как экранные. Команда CLIPPING включает и выключает отсечение отрезков по границам области вывода. Если отсечение выключено, а отрезки выходят за границы области вывода, то они рисуются полностью.

Примечание. Если координаты отрезков выходят за пределы экрана при выключенном отсечении, возможна порча содержимого ОЗУ КЦГД и программ графического протокола.

Иногда требуется изменить границы области вывода, не устанавливая новые значения коэффициентов пересчета координат. Для этого в команде VIEWPORT имеется специальный признак, а в интерфейсных пакетах имеются две процедуры — VIEWPORT и NVIEWPORT, последняя не изменяет имеющиеся коэффициенты пересчета координат.

Формат команд:

```
WINDOW :  
2 (62) XWLEFT3YWBOTTOM3XWRIGHT3YWTOP3  
VIEWPORT :  
1 (61) XVLEFT3+CHANGE2YVBOTTOM3XVRIGHT3YVTOP3
```

Параметр CHANGE, передаваемый вместе с XVLEFT, указывает, будут ли устанавливаться новые коэффициенты пересчета. Если CHANGE=1, то устанавливаются новые коэффициенты пересчета координат, так, что окно в мировых координатах будет отображаться на поле вывода. Если CHANGE=0, то пересчет координат не изменяется. При этом устанавливаются только новые границы поля вывода.

```
MAPPING :  
R (122) ONOFF1
```

Параметр ONOFF, равный нулю, выключает преобразование координат, любое другое значение — включает.

```
CLIPPING :  
S (123) ONOFF1
```

Параметр ONOFF, равный 0, выключает отсечение отрезков границами поля вывода, любое другое значение снова включает такое отсечение.

Для задания окна, области вывода и управления отсечением и масштабированием в программах на языке Паскаль необходимы следующие объявления:

```
PROCEDURE  
WINDOW(XWLEFT,YWBOTTOM,XWRIGHT,YWTOP:INTEGER);  
EXTERNAL;  
PROCEDURE  
VIEWPORT(XVLEFT,YVBOTTOM,XVRIGHT,YVTOP:INTEGER);  
EXTERNAL;
```

```
PROCEDURE  
NVIEWPORT(XVLEFT,YVBOTTOM,XVRIGHT,YVTOP:INTEGER);  
EXTERNAL;  
PROCEDURE MAPPING(ONOFF:INTEGER); EXTERNAL;  
PROCEDURE CLIPPING(ONOFF:INTEGER); EXTERNAL;
```

Из программ на языке Паскаль процедуры вызываются операторами:

```
WINDOW(XWLEFT,YWBOTTOM,XWRIGHT,YWTOP);  
VIEWPORT(XVLEFT,YVBOTTOM,XVRIGHT,YVTOP);  
NVIEWPORT(XVLEFT,YVBOTTOM,XVRIGHT,YVTOP);  
MAPPING(ONOFF);  
CLIPPING(ONOFF);
```

На языке Фортран:

```
CALL WINDOW (IXWL,IYWB,IXWR,IYWT)  
CALL VIEWPO (IXVL,IYVB,IXVR,IYVT)  
CALL NVIEWP (IXVL,IYVB,IXVR,IYVT)  
CALL MAPPIN (IONOFF)  
CALL CLIPPI (IONOFF)
```

На языке Си:

```
int XwLeft,YwBottom,XwRight,YwTop;  
int XvLeft,YvBottom,XvRight,YvTop;  
int onoff;  
window (XwLeft,YwBottom,XwRight,YwTop);  
viewport (XvLeft,YvBottom,XvRight,YvTop);  
nviewport (XvLeft,YvBottom,XvRight,YvTop);  
mapping (onoff);  
clipping (onoff);
```

Здесь:
XWLEFT, IXWL — левая граница окна;
YWBOTTOM, IYWB — нижняя граница окна;
XWRIGHT, IXWR — правая граница окна;
YWTOP, IYWT — верхняя граница окна;
XVLEFT, IXVL — левая граница области вывода;
YVBOTTOM, IYVB — нижняя граница области;
XVRIGHT, IXVR — правая граница области (>XVLEFT);
YVTOP, IYVT — верхняя граница области (>YVBOTTOM).

7. ОЧИСТКА ОБЛАСТИ ВЫВОДА

После установки границ области вывода обычно требуется очистить ее, то есть заполнить определенным цветом. Это выполняет команда NEWPICTURE с параметрами DWMODE

и ICOLOR. Параметр DWMODE устанавливает режим записи. Режим инверсии задается кодом —1, режим очистки разрядов пиксела — 0, наложение — 1, замещение цвета — 2 или 3. В тех случаях, когда требуется закрасить поле вывода цветом независимо от старого изображения в нем, необходимо задавать код 2. Параметр ICOLOR задает код цвета закраски.

Формат команды:

```
NEWPICTURE :  
0 (60) (DWMODE+1), ICOLOR;
```

В программе на языке Паскаль необходимо следующее объявление:

```
PROCEDURE NEWPICTURE(DWMODE,ICOLOR:INTEGER); EXTERNAL;
```

Из программы на языке Паскаль процедура вызывается оператором:

```
NEWPICTURE(DWMODE,ICOLOR);
```

На языке Фортран:

```
CALL NEWPIC(IDWM,ICOLOR)
```

На языке Си:

```
int dwmode, icolor;  
newpicture(dwmode,icolor);
```

8. УСТАНОВКА ЦВЕТА И РЕЖИМА ВЫВОДА

Команда COLOR устанавливает так называемый текущий цвет (от 0 до 15), иногда также называемый индексом цвета, которым будут рисоваться последующие примитивы. В большинстве команд предусмотрено локальное указание цвета, однако в командах, в которых координаты передаются в виде смещений, цвет не передается. Эти команды будут рисовать примитивы с кодом цвета, установленным предыдущей командой COLOR. Ее параметр ICOLOR (целое число от 0 до 15) задает индекс текущего цвета.

Команда SETCOLOR устанавливает соответствие между кодом пиксела (индексом цвета с величиной от 0 до 15) и действительным цветом, отображаемым на экране. Она имеет параметры ICOLOR — индекс устанавливаемого цвета

(от 0 до 15), и VALUE — его значение, вычисляемое по формуле $R+4*G+16*B$, где R, G, B — яркости соответственно красного, зеленого и синего основных цветов (четыре градации, от 0 до 3).

В режиме высокого разрешения индексы (номера) цвета могут принимать значения 0, 5, 10, 15 или 0, 1, 2, 3, что равнозначно.

Если необходимо «быстро» изменить значения всех 16 цветов, то можно воспользоваться командой PUTCOLORS.

Пикселы с индексом цвета, равным 15, обычно индицируются мигающими синхронно с курсором — их цвет меняется между цветами с индексами 5 и 10. Это мерцание может быть выключено командой BLINK. Параметр этой команды, равный нулю, выключает мигание 15-го цвета, после чего на него может быть назначен произвольный цвет через команду SETCOLOR. Ненулевое значение параметра включает мигание, после чего 15-й цвет будет попеременно принимать значения 10-го и 5-го цвета.

Текущее содержимое таблицы цветов может быть получено программой через команду GETCOLOR. По этой команде из КЦГД в ЭВМ передаются 16 байтов — значения всех 16 цветов.

При построении изображений коды пикселов могут записываться различными способами. Режим вывода устанавливается командой DRAWMODE и используется при рисовании всех примитивов, кроме графического курсора и локатора, которые всегда рисуются в режиме инверсии битов (для того, чтобы при их перемещении не нарушить изображение). Параметр DWMODE этой команды со значением из диапазона —1...3 задает правило, по которому будут взаимодействовать коды пикселов в графической памяти КЦГД и индекс (номер) цвета выводимых в нее примитивов. В результате взаимодействия в графическую память (а значит и на экран) запишется результат по правилу:

DWMODE код пиксела, записываемый в видео-ОЗУ

—1	<номер цвета>	XOR	<код пиксела в памяти>
0	<номер цвета>	BIC	<код пиксела в памяти>
1	<номер цвета>	BIS	<код пиксела в памяти>
2	<номер цвета>		(с прозрачными пробелами)
3	<номер цвета>		(с непрозрачными пробелами)

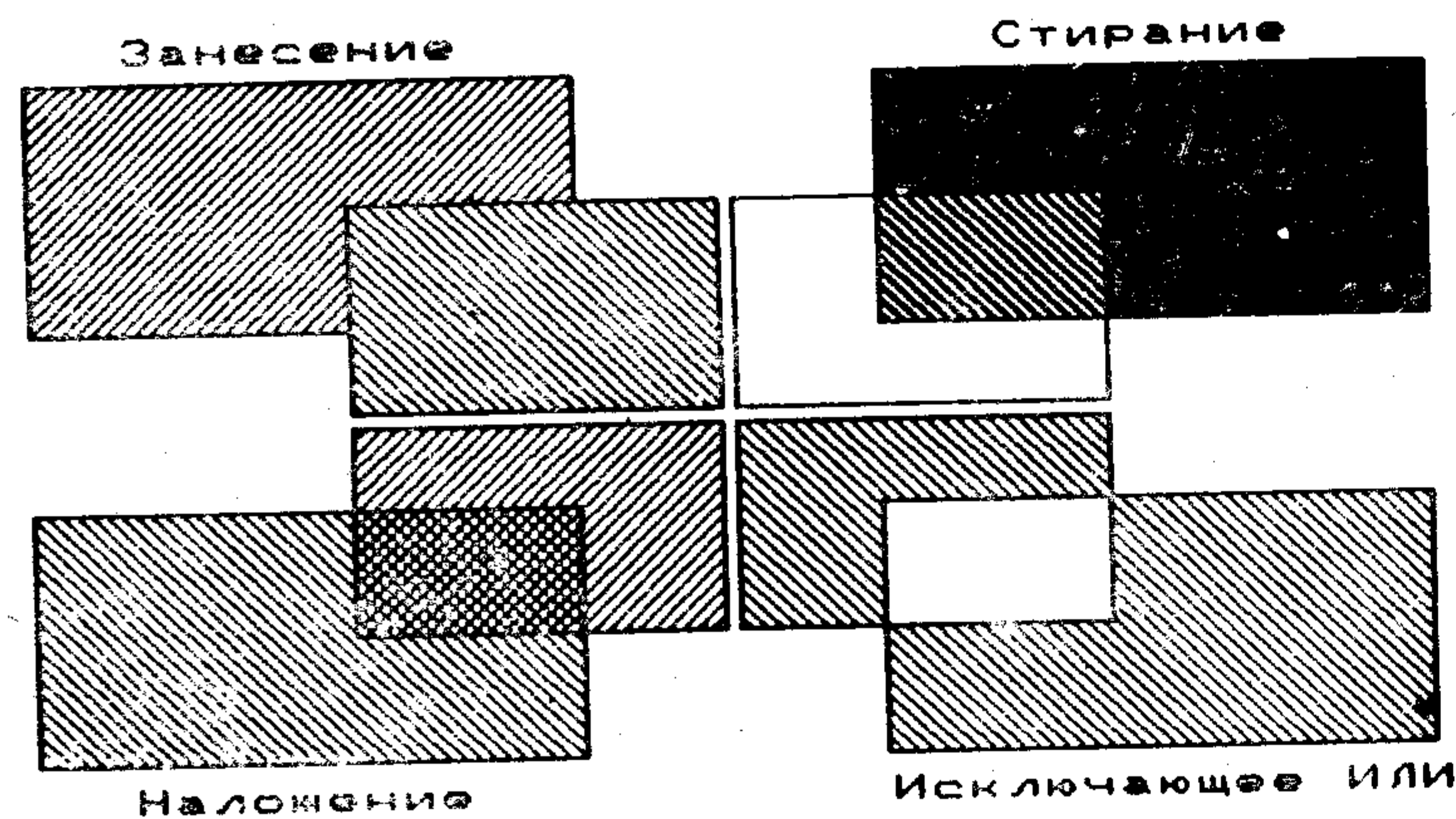
Операции XOR, BIC, BIS соответственно означают исключающее ИЛИ, маскирование, ИЛИ.

Режим записи 2 — запись с прозрачными пробелами.

Когда он установлен, то при выводе пунктирных линий (**LINestyle>0**) выводятся только точки, соответствующие единичным битам образца линии. При рисовании символов выводятся не все точки прямоугольной символьной ячейки, а лишь те, которые соответствуют единичным битам прорисовки, то есть, если на одно и то же место будут выведены различные символы, то они наложатся друг на друга. Кроме того, при заполнении области штриховкой точки области, на которые не попадают линии штриховки, остаются неизменными.

Режим записи 3 — запись с непрозрачными пробелами. При выводе линий точки, соответствующие нулевым битам образца, выводятся текущим цветом фона. При рисовании символов выводится полная ячейка, причем фон ячейки выводится текущим цветом фона, а при заполнении области штриховкой в точки, на которые не попадают линии штриховки, записывается цвет фона. Цвет фона берется от последней команды **NEWPICTURE** (очистка области вывода). Когда установлен режим записи 3, то при выводе различных символов в одно и то же место новый символ полностью «затирает» предыдущий.

Для реализации выборочного стирания элементов изображения необходимо пользоваться режимом вывода —1, при этом при первом выводе некоторого примитива он будет выводиться цветом, являющимся результатом операции **ИСКЛЮЧАЮЩЕЕ ИЛИ** над записываемым цветом и цветом фона. При повторном выводе того же самого примитива вос-



становится цвет фона, а нарисованный примитив «исчезнет».

Операция **BIS** накладывает текущий цвет на имеющийся, **BIC** — исключает его из имеющегося. В большинстве случаев имеет смысл пользоваться режимом замещения (параметр 2).

На иллюстрации показаны примеры взаимодействия цвета фона и записываемого цвета при разных режимах вывода.

Формат команд:

COLOR :
4 (64) **COLOR**₁
SETCOLOR :
5 (65) **ICOLOR**₁ **VALUE**₁
PUTCOLORS :
6 (66) **CO**₁ **C2**₁ **C3**₁ ... **C15**₁

За кодом команды следуют 16 байтов со значениями цветов.

GETCOLORS :

f (146)

BLINK :

(140) **ONOFF**₁

DRAWMODE :

3 (63) (**DWMODE**+1)₁

Для установки текущего цвета, установки представления цветов, быстрой их загрузки, управления миганием пятнадцатого цвета и режимом вывода в программах на языке Паскаль необходимы следующие объявления:

```
TYPE COL=ARRAY[0..15] OF CHAR;
PROCEDURE COLOR(ICOLOR:INTEGER); EXTERNAL;
PROCEDURE SETCOLOR(ICOLOR,VALUE:INTEGER); EXTERNAL;
PROCEDURE PUTCOLORS(VAR COLORS:COL); EXTERNAL;
PROCEDURE GETCOLORS(VAR COLORS:COL); EXTERNAL;
PROCEDURE BLINK(ONOFF:INTEGER); EXTERNAL;
PROCEDURE DRAWMODE(DWMODE:INTEGER); EXTERNAL;
```

Из программы на языке Паскаль процедуры вызываются операторами:

```
COLOR (ICOLOR);
SETCOLOR(ICOLOR,VALUE);
PUTCOLORS(COLORS);
```

```
GETCOLOS (COLORS);
BLINK (ONOFF);
DRAWMODE (DWMODE);
```

На языке Фортран:

```
BYTE COLORS (16)
CALL COLOR (ICOLOR)
CALL SETCOL (ICOLOR, IVALUE)
CALL PUTCOL (COLORS)
CALL GETCOL (COLORS)
CALL BLINK (IONOFF)
CALL DRAWMO (IDWMOD)
```

На языке Си:

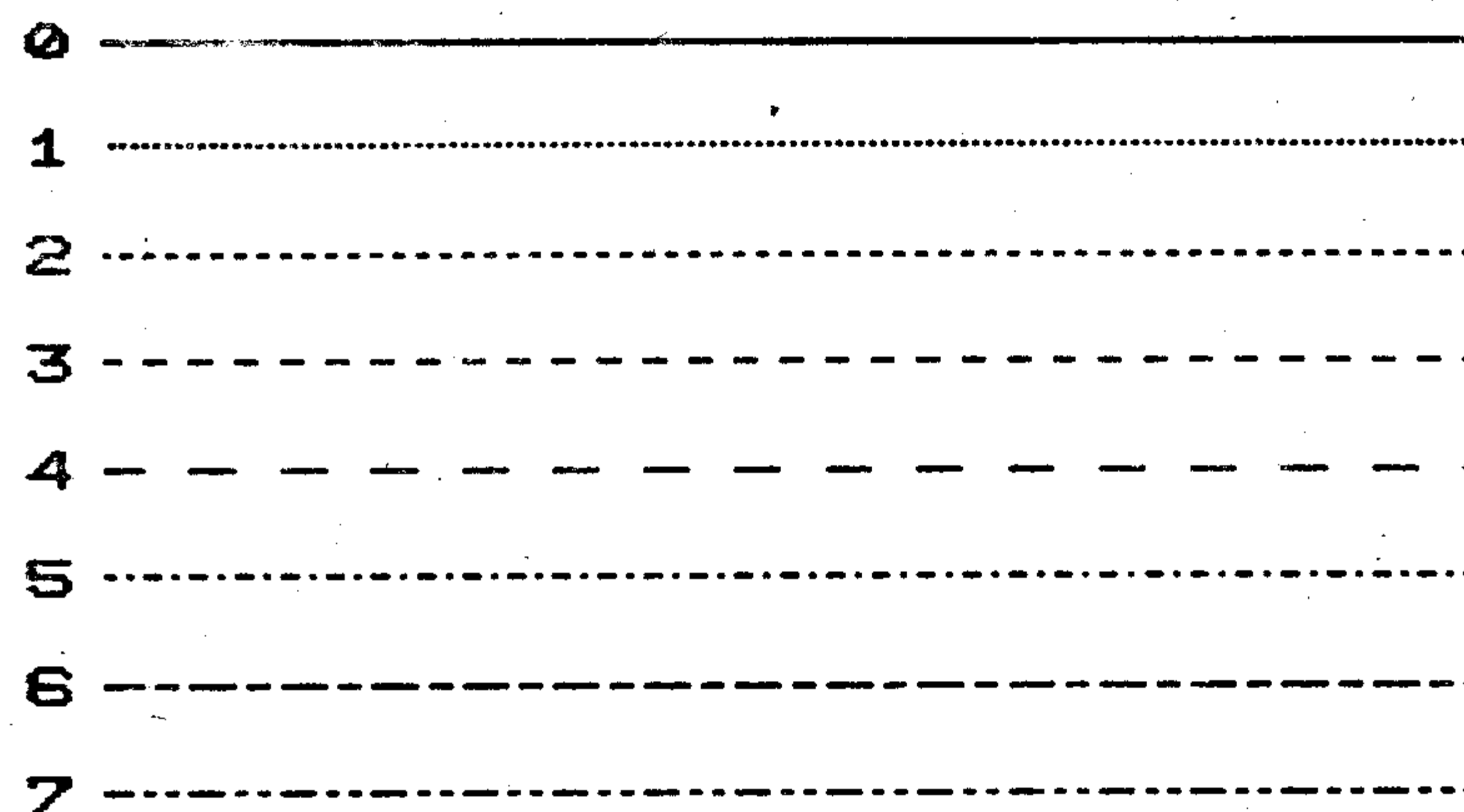
```
int icolor, value;
char colors [16];
int onoff;
int dwmode;

color (icolor);
setcolor (icolor, value);
putcolors (colors);
getcolos (colors);
blink (onoff);
drawmode (dwmoe);
```

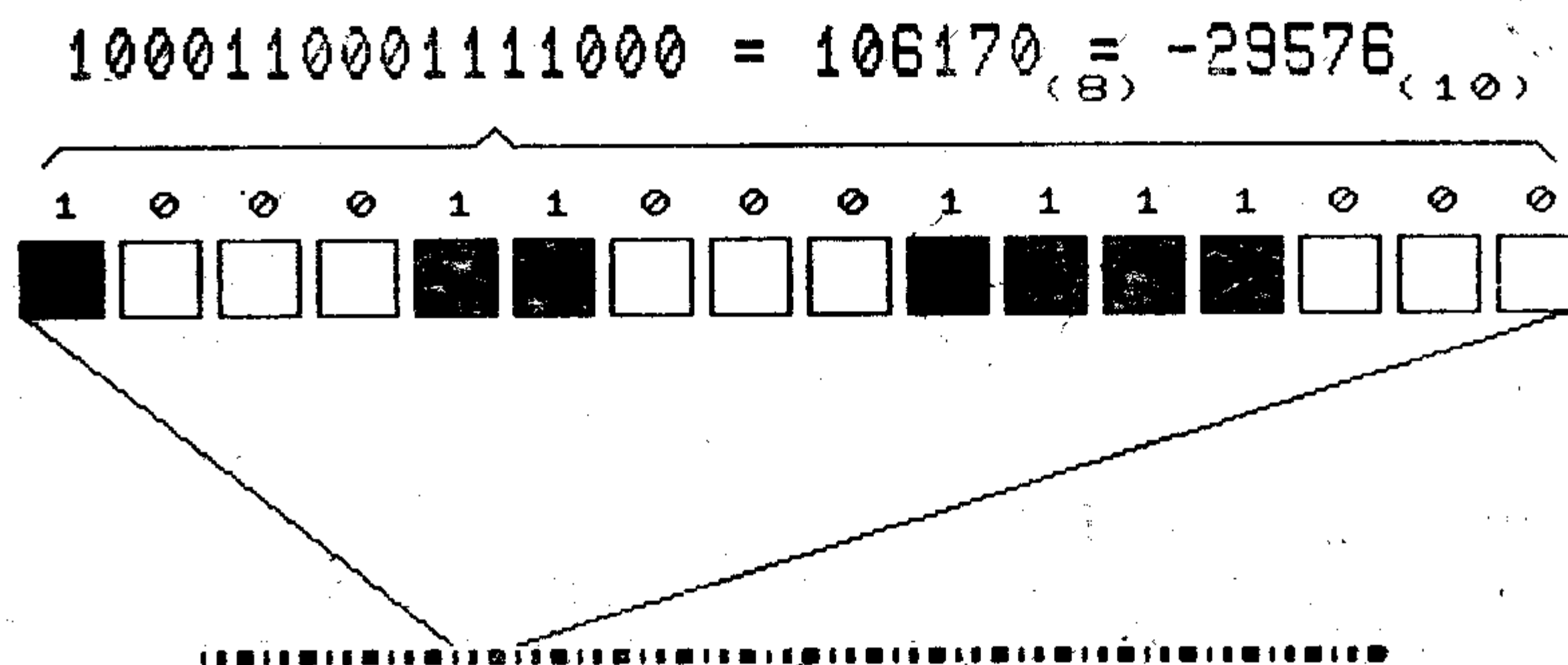
9. ПОСТРОЕНИЕ ТОЧЕК, МАРКЕРОВ И ОТРЕЗКОВ ПРЯМЫХ, УСТАНОВКА ТИПА ЛИНИИ

Протокол содержит команды, посредством которых можно выводить точки, отрезки прямых. Отрезки выводятся либо сплошными линиями, либо разрывными линиями по одному из 7 заранее определенных образцов. Образец представляет собой 16-разрядное целое число без знака. При построении примитивов это число циклически сдвигается, старший разряд числа, полученный после его сдвига, указывает, следует ли выводить очередной пиксел заданным цветом или нет. Образец типа линии может также быть задан пользователем. Номер типа выводимой линии (от 0 до 7) устанавливается командой `LINESTYLE`. При ее выполнении, кроме того, образец устанавливается в начальное состояние.

Образцы встроенных типов линий приведены на рисунке:



Пользовательский образец задается при помощи команды `USERSTYLE`. Параметр команды — 16-разрядное число — задает образец. Для установки этого образца в качестве текущего подается команда `LINESTYLE` с аргументом 8. На рисунке приведен пример конструирования образца типа линии.

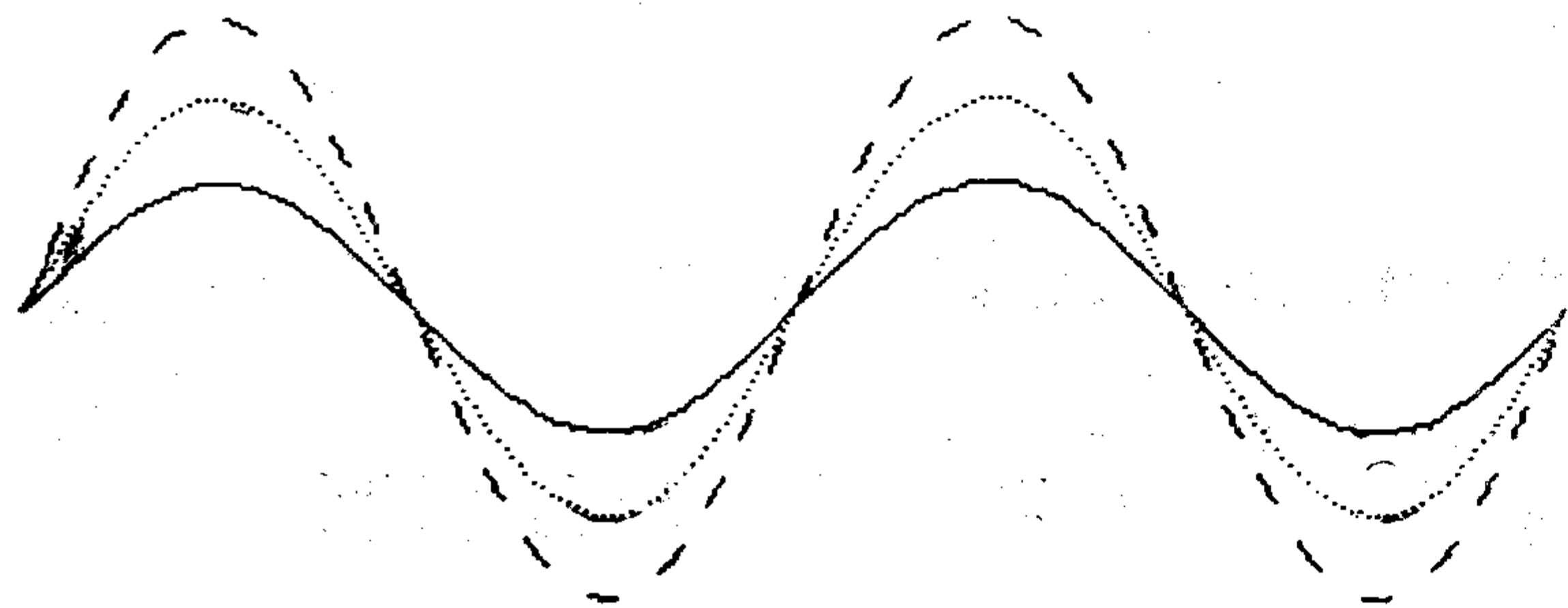


Отрезки прямых строятся командой `LINE` — с заданными координатами начала и конца, и командой `RLINE`, в которой начальной точкой является текущая позиция, а конечная

точка задана смещением от начальной. После построения отрезка текущая позиция перемещается в его конец. При построении отрезка командой RLINE точка в начальной позиции не выводится. Это делается для того, чтобы при построении ломаных линий в режиме вывода инверсией вершины ломаной не выводились дважды, что привело бы к их исчезновению. Команда LINE всегда выводит начальную точку отрезка.

Текущая позиция, полученная в результате отображения мировых координат на экранные, может выходить за пределы поля вывода, естественно, примитивы, построение которых начинается от нее, будут частично или полностью отсечены.

Вывод точек может производиться двумя способами. Команды DOT и RDOT просто выводят точки, а DOTL и RDOTL ставят точку в соответствии с установленным типом линии, они предназначены для построения произвольных кривых с заданным типом линии. Команды DOT и DOTL ставят точку по заданным абсолютным координатам, а для команд RDOT и RDOTL задается смещение от текущей позиции. После вывода точки текущая позиция устанавливается на нее. На рисунке приведен пример построения кривой с разными типами линий из отдельных точек:



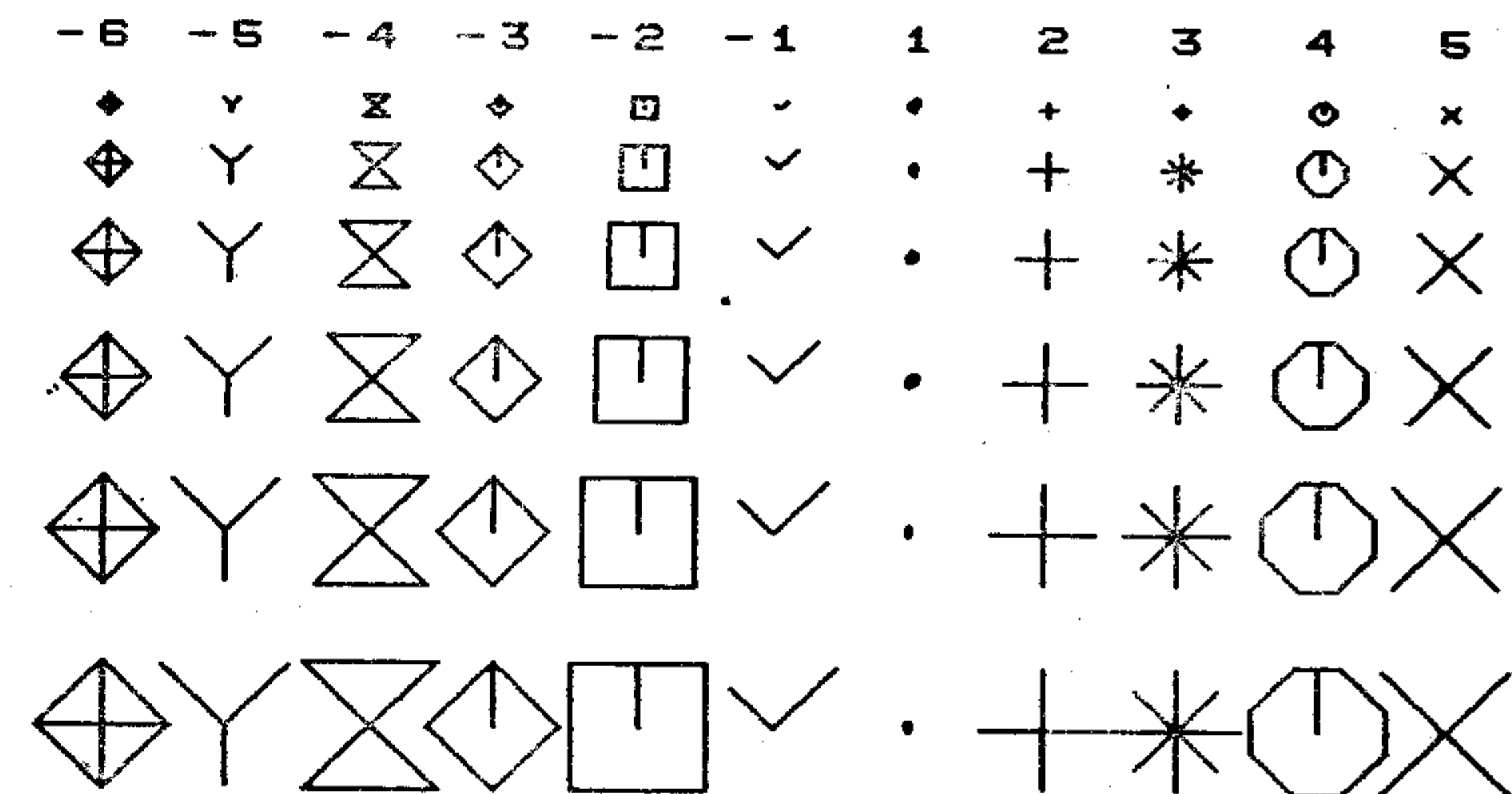
Командой POSITION можно установить мировые координаты текущей позиции, не выводя на это место точку. Команда RPOSITION перемещает текущую позицию на заданную величину смещения.

Маркером называется символ специального вида, выводимый в заданной позиции. Вывод маркеров может быть использован, например, для разметки изображения, построения графиков и т. д. Предусмотрено 11 вариантов (типов)

маркеров. Маркеры могут иметь различный размер (множитель размера от 1 до 8) и цвет. Они рисуются с учетом установленного режима вывода. Маркеры строятся в экранных координатах, то есть их размер и пропорции не зависят от текущих коэффициентов пересчета мировых координат в экранные.

При построении маркеры отсекаются текущими границами установленной области вывода.

На рисунке приведены прорисовки маркеров, которые реализованы в программе графического протокола:



Маркеры с типами 1—5 соответствуют маркерам стандарта GKS. Вид и размер маркера устанавливаются командой SETMARKER с параметрами MTYPE (тип маркера) и MSIZE (множитель размера), который может принимать значения от 1 до 8. Если команда SETMARKER не была подана, то выводится маркер типа 3 (звездочка) одинарного размера.

Вывод маркера в указанной позиции выполняется командой MARKER, а команда RMARKER выводит его в позицию, заданную смещением от текущей позиции.

Формат команд:

DOT :
пробел (40) (X₃, Y₃) + ICOLOR₄

RDOT :
! (41) (DX, DY)₃

DOTL :
 » (42) (X₃,Y₃) + ICOLOR₄
 RDOTL :
 # (43) (DX,DY)₃
 SETMARKER :
 [(133) MTYPE₁ (MSIZE-1)₁
 MARKER :
 L (114) (X₃, Y₃) + ICOLOR₄
 RMARKER :
 M (115) (DX,DY)₃
 LINE :
 \$ (44) X1₃ Y1₃ (X2₃,Y2₃) + ICOLOR₄
 RLINE :
 % (55) (DX,DY)₃
 POSITION :
 > (76) X₃ Y₃
 RPOSITION :
 ? (77) (DX,DY)₃
 LINSTYLE :
 7 (67) NSTYLE₁
 USERSTILE :
 8 (70) PATTRN₃

Для вывода точек, отрезков прямых, задания параметров маркеров, вывода маркеров, в программе на языке Паскаль необходимы следующие объявления:

```

PROCEDURE DOT(X,Y,ICOLOR:INTEGER); EXTERNAL;
PROCEDURE RDOT(DX,DY:INTEGER); EXTERNAL;
PROCEDURE DOTL(X,Y,ICOLOR:INTEGER); EXTERNAL;
PROCEDURE RDOTL(DX,DY:INTEGER); EXTERNAL;
PROCEDURE SETMARKER(MTYPE,MSIZE:INTEGER); EXTERNAL;
PROCEDURE MARKER(X,Y,ICOLOR:INTEGER); EXTERNAL;
PROCEDURE RMARKER(DX,DY:INTEGER); EXTERNAL;
PROCEDURE LINE(X1,Y1,X2,Y2,ICOLOR:INTEGER); EXTERNAL;
PROCEDURE RLINE(DX,DY:INTEGER); EXTERNAL;
PROCEDURE POSITION(X,Y:INTEGER); EXTERNAL;
PROCEDURE RPOSITION(DX,DY:INTEGER); EXTERNAL;
PROCEDURE LINSTYLE(NSTYLE:INTEGER); EXTERNAL;
PROCEDURE USERSTYLE(PATTRN:INTEGER); EXTERNAL;

```

Из программы на языке Паскаль процедуры вызываются операторами:

```

DOT(X,Y,ICOLOR);
RDOT(DX,DY);
DOTL(X,Y,ICOLOR);
RDOTL(DX,DY);
SETMARKER(MTYRE,MSIZE);
MARKER(X,Y,ICOLOR);
RMARKER(DX,DY);
LINE(X1, Y1, X2, Y2, ICOLOR);
RLINE(DX,DY);
POSITION (X,Y);
RPOSITION(DX,DY);
LINSTYLE(NSTYLE);
USERSTYLE(PATTRN);

```

На языке Фортран:

```

CALL DOT(IX,IY,ICOLOR)
CALL RDOT(IDX,IDY)
CALL DOTL(IX,IY,ICOLOR)
CALL RDOTL(IDX,IDY)
CALL SETMAR(MTYPE,MSIZE)
CALL MARKER(IX,IY,ICOLOR)
CALL RMARKER(IDX,IDY)
CALL LINE(IX1,IY1,IX2,IY2,ICOLOR)
CALL LINE(IDX,IDY)
CALL POSITI(IX,IY)
CALL RPOSIT(IDX,IDY)
CALL LINST(NSTYLE)
CALL USERST(IPATRN)

```

На языке Си:

```

int x,y,dx,dy,icolor;
int mtype,msize,xl,x2,y1,y2;
int nstyle;
unsigned patrn;
dot(x,y,icolor);
rdot(dx,dy);
dotl(x,y,icolor);
rdotl(dx,dy);
setmarker(mtype,msize);
marker(x,y,icolor);
rmarker(dx,dy);
line(x1,y1,x2,y2,icolor);

```



```

gline(dx,dy);
position(x,y);
rposition(dx,dy);
linestyle(nstyle);
userstyle(patrn);

```

Здесь X1, Y1 — мировые координаты начала отрезка, X2, Y2 — координаты конца отрезка, DX, DY — смещение от координат X, Y текущей позиции. В команде RLINE это смещение от начальной точки отрезка (текущей позиции) до его конечной точки. PATRN — целое число, передаваемое в качестве образца типа линии.

10. ВЫБОР И ОПРЕДЕЛЕНИЕ ШТРИХОВКИ И УЗОРА

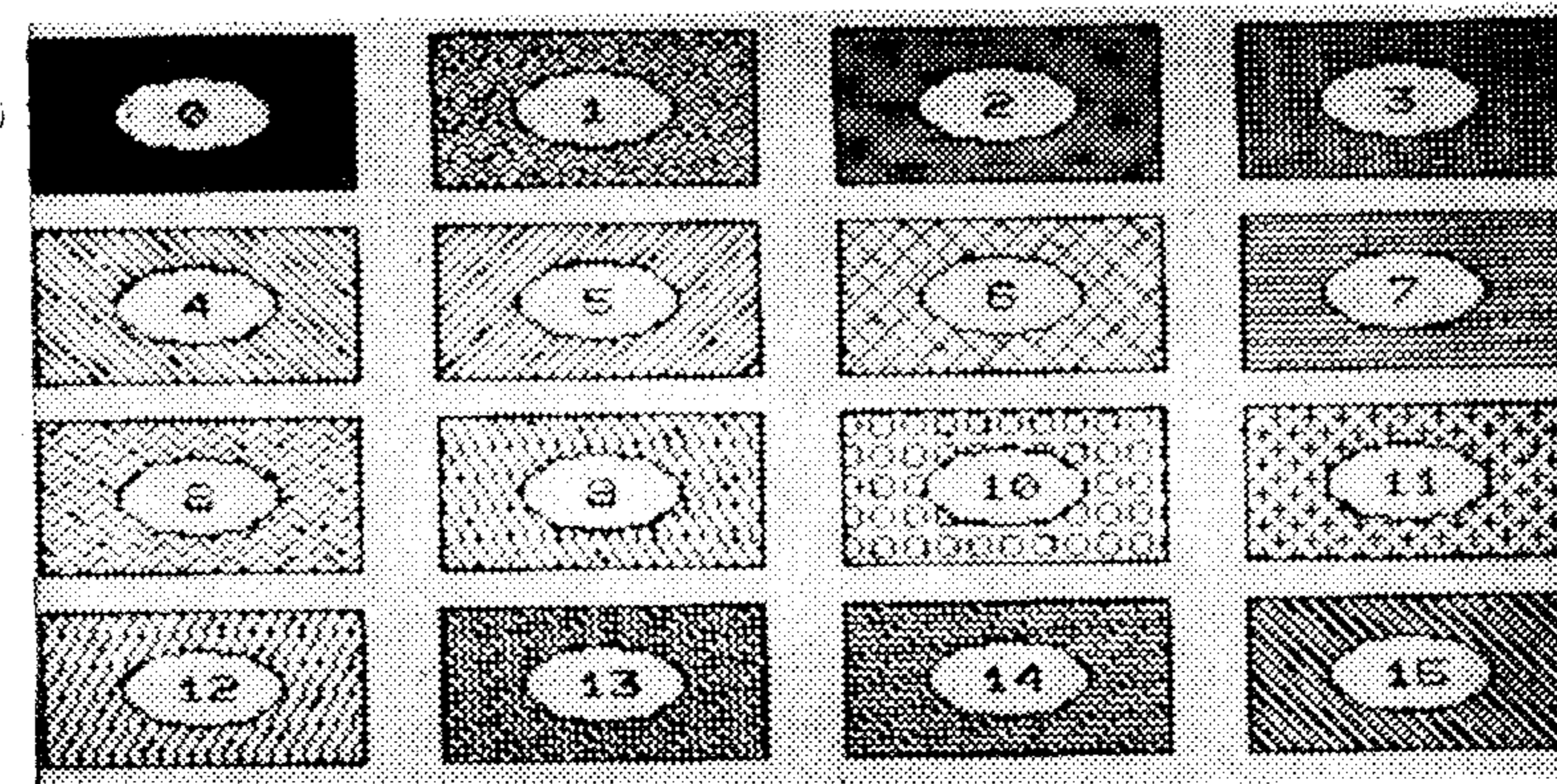
В протоколе имеются команды заполнения области определенным узором или штриховкой. Штриховка выполняется в соответствии с одним из 16 predetermined образцов или по образцу, заданному пользователем, при этом точки получают либо цвет фона, либо текущий цвет. При заполнении узором точки получают цвет, заданный рисунком узора.

Текущий вид заполнителя устанавливается командой SETFILLER. Может быть задан или номер одного из 16 predetermined видов штриховки, либо образец штриховки, заданный пользователем, либо узор, заданный пользователем. Ячейка штриховки и ячейка узора имеют размер 8×8 пикселей в режиме низкого разрешения и 16×8 пикселей в режиме высокого разрешения, таким образом, на экране их размер в обоих режимах одинаковый. Параметр FILLER задает номер: —1 — узор, заданный пользователем с помощью команды USERPATTERN, 0—15 — встроенные штриховки, 0 — сплошная заливка, >15 — заполнение штриховкой, заданной пользователем через команду USERHATCH.

На рисунке приведены виды штриховок, встроенных в программу графического протокола:

Пользовательский образец штриховки загружается посредством команды USERHATCH, а узор — через команду USERPATTERN.

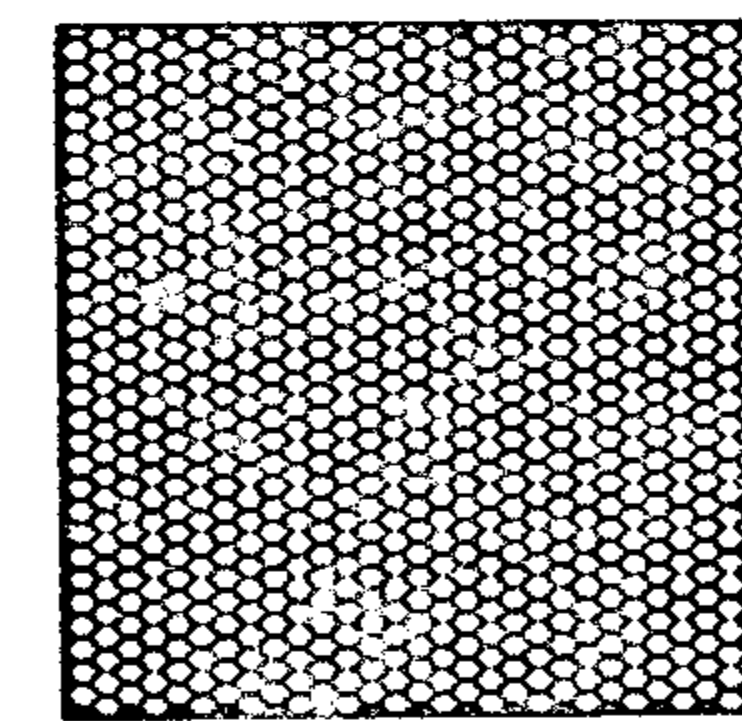
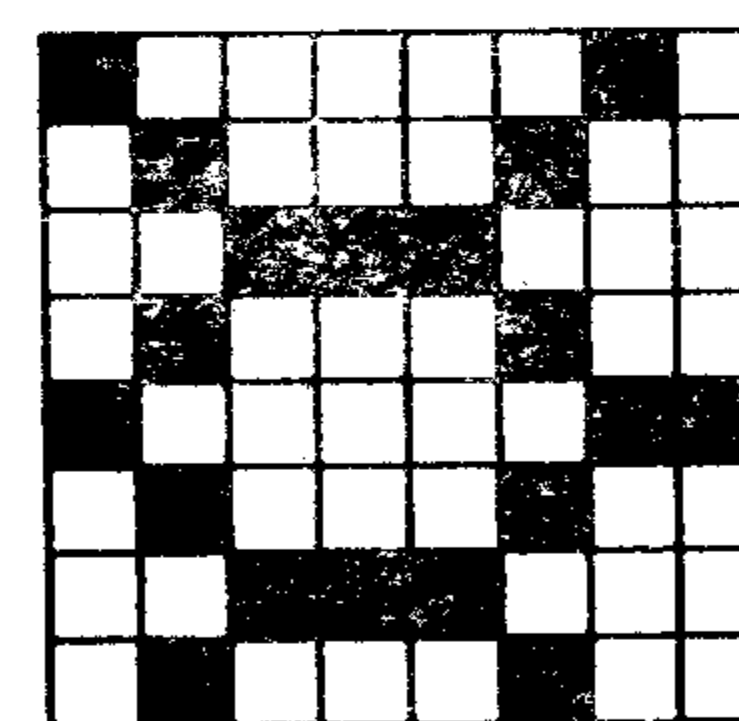
В качестве параметров команды USERHATCH посылаются 4 слова (8 байт), каждый байт кодирует пиксели строки ячейки штриховки, старший бит соответствует левому пикселу. Ячейка штриховки имеет размер 8×8 пикселей в ре-



жиме низкого разрешения и 16×8 пикселей в режиме высокого разрешения. В режиме высокого разрешения каждый бит массива HATCH соответствует двум пикселям изображения, а в режиме низкого разрешения каждый бит соответствует одному пикселу. Таким образом, на экране размер ячейки штриховки всегда одинаковый.

На рисунке приведен пример конструирования образца штриховки:

```
HATCH={1011048, 341048, 1015048, 341048}
```



Аргументом команды USERPATTERN является массив из 16 целых чисел (слов), в каждом слове закодировано 4 пиксела низкого разрешения или 8 пикселей высокого разрешения — по 4 или 2 бита на пиксел. Старшие 4 или 2 бита со-

ответствуют самому левому пикселу. Ячейка узора состоит из 8 строк по два слова в каждой. В режиме низкого разрешения это соответствует области 8×8, а в режиме высокого разрешения — 16×8.

Форматы команд:

SETFILLER:

T (124) (FILLER+1)₁

USERPATTERN:

U (125) P1₃ P2₃ ... P16₃

USERHATCH:

V (126) H1₃ H2₃ H3₃ H4₃

H1, H2, H3, H4 — четыре слова, каждое из которых передается в виде трех символов. В этих словах передаются значения 64 пикселей ячейки штриховки.

Для использования этих команд в программах на языке Паскаль необходимы следующие объявления:

```
TYPE HATCH=ARRAY[1..4] OF INTEGER;
```

```
TYPE PATT=ARRAY[1..16] OF INTEGER;
```

```
PROCEDURE SETFILLER(FILLER:INTEGER); EXTERNAL;
```

```
PROCEDURE USERHATCH(VAR HATCH:HATCH): EXTERNAL;
```

```
PROCEDURE USERPATTERN(VAR PATTERN:PATT); EXTERNAL;
```

Из программы на языке Паскаль процедуры вызываются операторами:

```
SETFILLER(FILLER);
```

```
USERHATCH(HATCH);
```

```
USERPATTERN(PATTERN);
```

Из Фортрана:

```
INTEGER HATCH(4)
```

```
INTEGER PATRN(16)
```

```
CALL SETFIL (IFILL)
```

```
CALL USERHA (HATCH)
```

```
CALL USERPA (PATRN)
```

На языке Си:

```
int filler;
```

```
int hatch[4];
```

```
int pattern[16];
```

```
setfiller (filler);
```

```
userhatch (hatch);
```

```
userpattern (pattern);
```

11. ЗАПОЛНЕНИЕ (РАСКРАСКА) ФИГУРЫ УЗОРОМ ИЛИ ШТРИХОВКОЙ

Графический протокол содержит команды заполнения замкнутых областей определенным узором или штриховкой. Можно раскрашивать уже сформированное на экране изображение или производить заполнение в процессе построения контура фигуры.

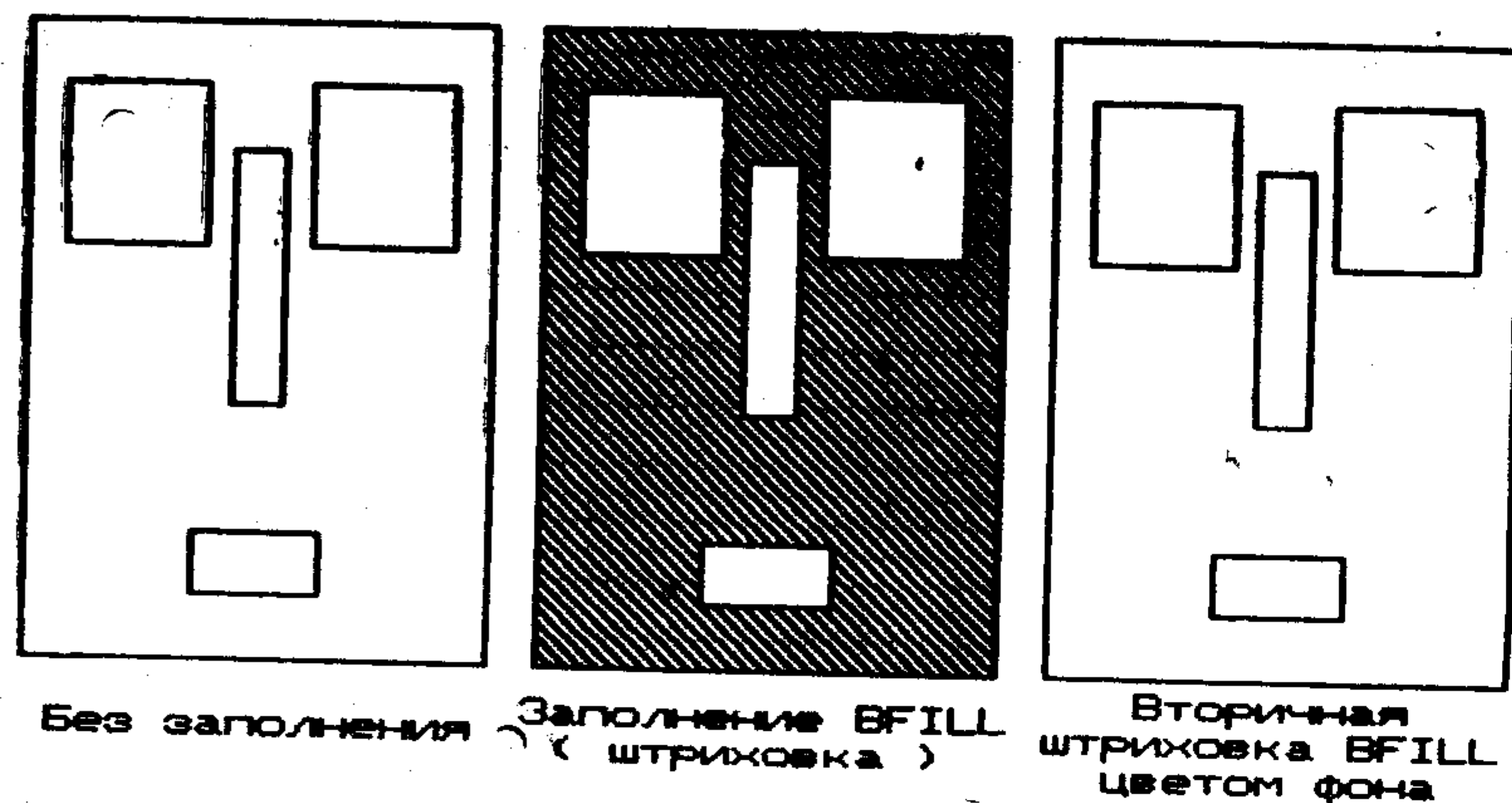
В первом случае раскрашиваемая область может быть определена либо цветом внутренней области, подлежащей раскраске, либо цветом границы, до которой производится раскрашивание. Во втором случае фигура сразу же выводится заполненной.

Область, состоящая из соприкасающихся по горизонтали и вертикали пикселей одинакового цвета, называется внутренне-определенной четырехсвязной областью. Для ее заполнения используются команды FILL и RFILL, которые считывают код цвета заданной начальной точки и затем раскрашивают область с точками такого же цвета.

Область, ограниченная связанными по вертикали, горизонтали и диагонали пикселями одинакового цвета, называется гранично-определенной четырехсвязной областью. Для заполнения такой области используются команды BFILL и RBFILL. Для этих команд задается цвет границы, до которой производится раскраска, причем точки внутри заполняемой области могут иметь различные цвета.

Раскраска области начинается от заданной точки. На время закраски в команде FILL для штриховок используется цвет ICOLOR, а команда RFILL выполняет штриховку с использованием текущего цвета. Параметр FILLER принимает такие же значения, как и в команде SETFILLER и временно устанавливает новый тип заполнителя. При использовании команды BFILL вывод штриховки выполняется текущим цветом, установленным командой COLOR. Цвет границы, до которой производится заполнение, в команде BFILL задается параметром ICOLOR, а команда RBFILL производит заполнение до границы с текущим цветом. Команды BFILL и RBFILL могут применяться для снятия изображений (например, штриховки) в некоторой гранично-определенной области. Если изображения, находящиеся в этой области, не содержат цвета, совпадающего с цветом границы, то после выполнения операции BFILL эта область будет заполнена заданным образцом штриховки или закрашена од-

ним цветом (образец 0), как показано на приведенной ниже иллюстрации:



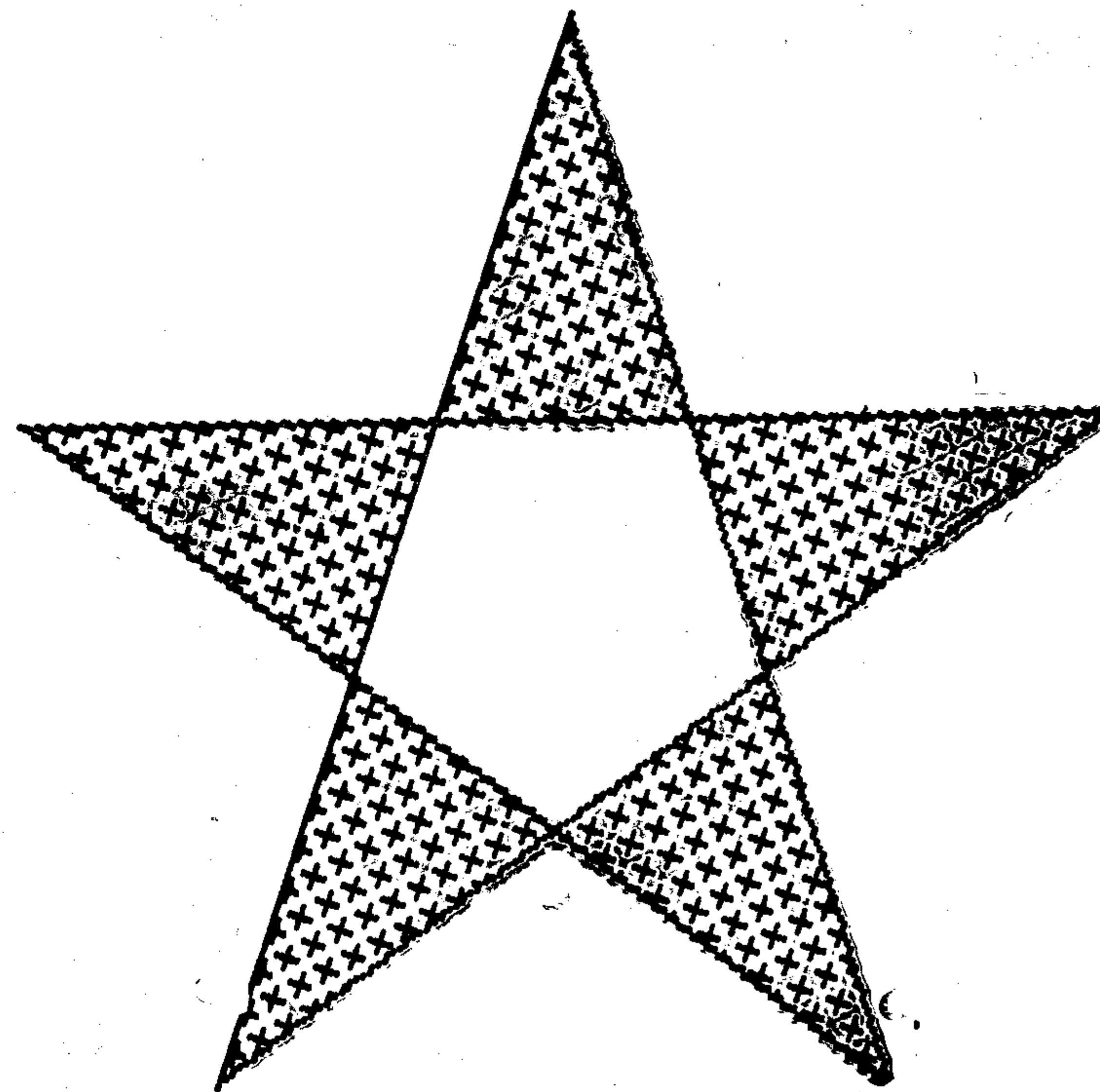
Если контур раскрашиваемой области выходит за пределы установленного поля вывода, то заполнение производится до границы поля вывода. Если начальная точка, указанная в командах раскраски, находится вне поля вывода, то заполнение не выполняется. BFill (RBFILL) работает несколько медленнее, чем Fill (RFILL).

При установленном режиме вывода инверсией (DRAWMODE=-1) возможен случай, когда цвет закрашиваемой области получится таким же, как и цвет границы (для BFill и RBFILL). В результате при повторной выдаче этой команды для снятия закрашки первоначальное изображение не восстановится. Поэтому следует с осторожностью использовать раскраску гранично-определенной области в режиме вывода инверсией.

Режим, когда фигуры сразу строятся заполненными, включается и выключается командой DOFILL. Если параметр команды DOFILL ONOFF > 0, то заполнение включается. Параметр ONOFF = 0 выключает заполнение. Фигуры закрашиваются заполнителем, заданным командой SET-FILLER, и цветом, заданным командой COLOR (текущим). Заполняться могут как простые замкнутые фигуры, рисуемые одной командой протокола (окружность, эллипс, их секторы и сегменты, прямоугольники, многоугольники и их наборы, замкнутые кривые и их наборы), так и сложные фигуры, за-

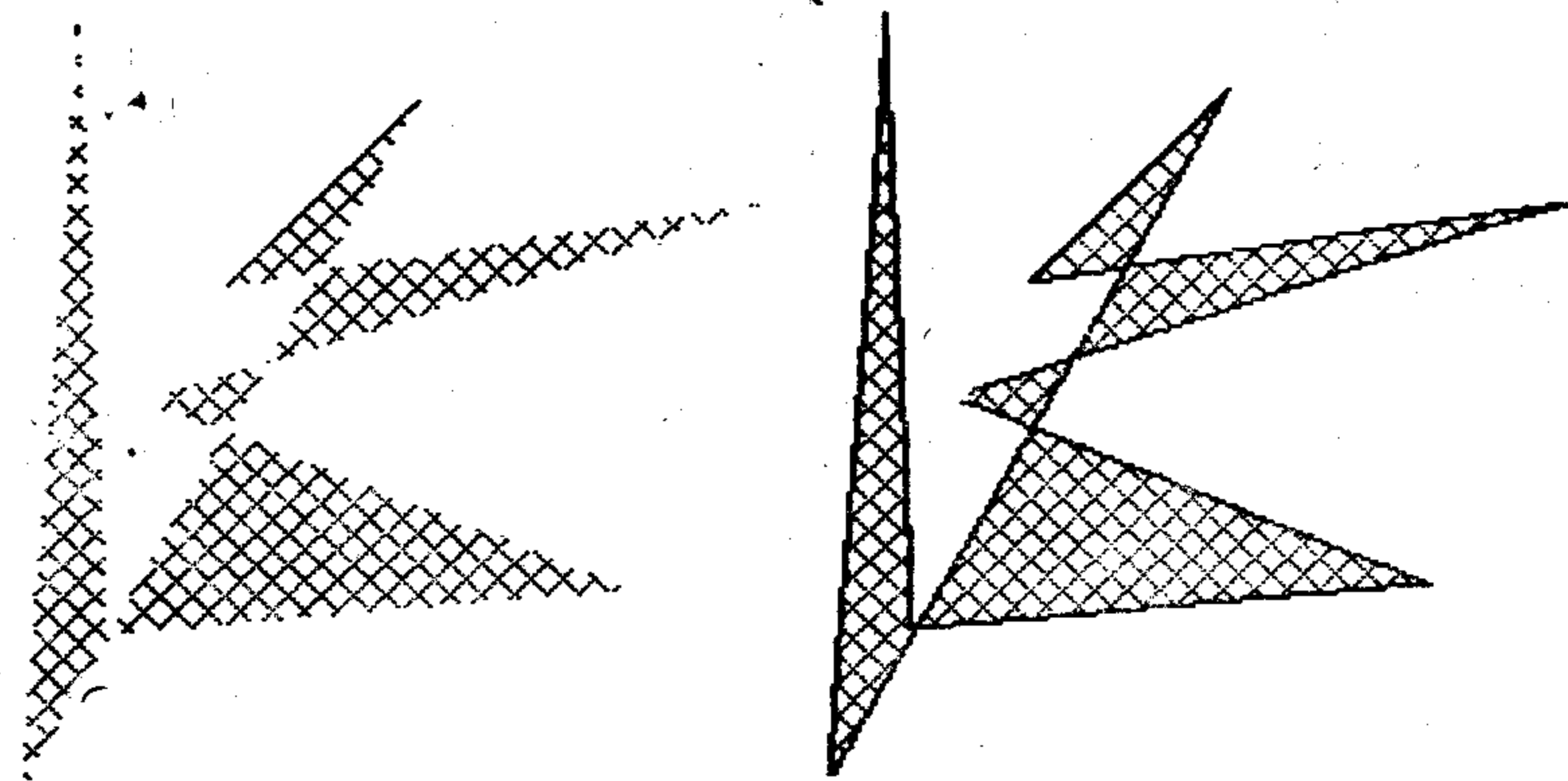
даваемые набором примитивов. Для задания сложной фигуры следует подать команду BEGFIGURE — начало заполненной фигуры. Затем следует набор примитивов, задающих контур фигуры. В качестве их можно использовать следующие команды: точка, отрезок, дуга окружности или эллипса, кривая и ломаная. Определение фигуры завершается командой ENDFIGURE. По этой команде производится отрисовка заполненной фигуры. Если требуется задать фигуру, состоящую из нескольких контуров (например, вложенных), то следует использовать команду NEWREGION, по которой замыкается текущий контур и начинается новый без отрисовки фигуры. Команда NEWREGION в неявном виде выполняется и в случае, если в процессе определения фигуры встретилась команда рисования простой замкнутой фигуры (окружности, прямоугольника и т. д.).

Если фигура состоит из нескольких контуров, то вывод их в заполненном виде производится совместно, при этом дейст-



вует правило «чет-нечет», т. е. заполнение осуществляется от самой левой грани до следующей, пропуск до следующей грани (если она есть), снова заполнение, и т. д. Это же правило распространяется и на самопересекающиеся контуры.

При построении заполненных фигур их граница может или обрисовываться сплошной ломаной линией (закрытые фигуры), или выводиться тем же заполнителем, как показано на рисунке:



Режим вывода границы устанавливается командой SETEDGE. Параметр команды ONOFF > 0 включает «обводку» границы выводимых заполненных многоугольников (начально установленный режим), ONOFF = 0 выключает рисование границ. Команды SETEDGE могут произвольным образом встречаться внутри определения сложной фигуры. Границы контура сложной фигуры выводятся заполнителем, если в момент задания данной грани ONOFF был равен 0, или сплошной линией в противном случае. Тип линии при рисовании границы контура не учитывается.

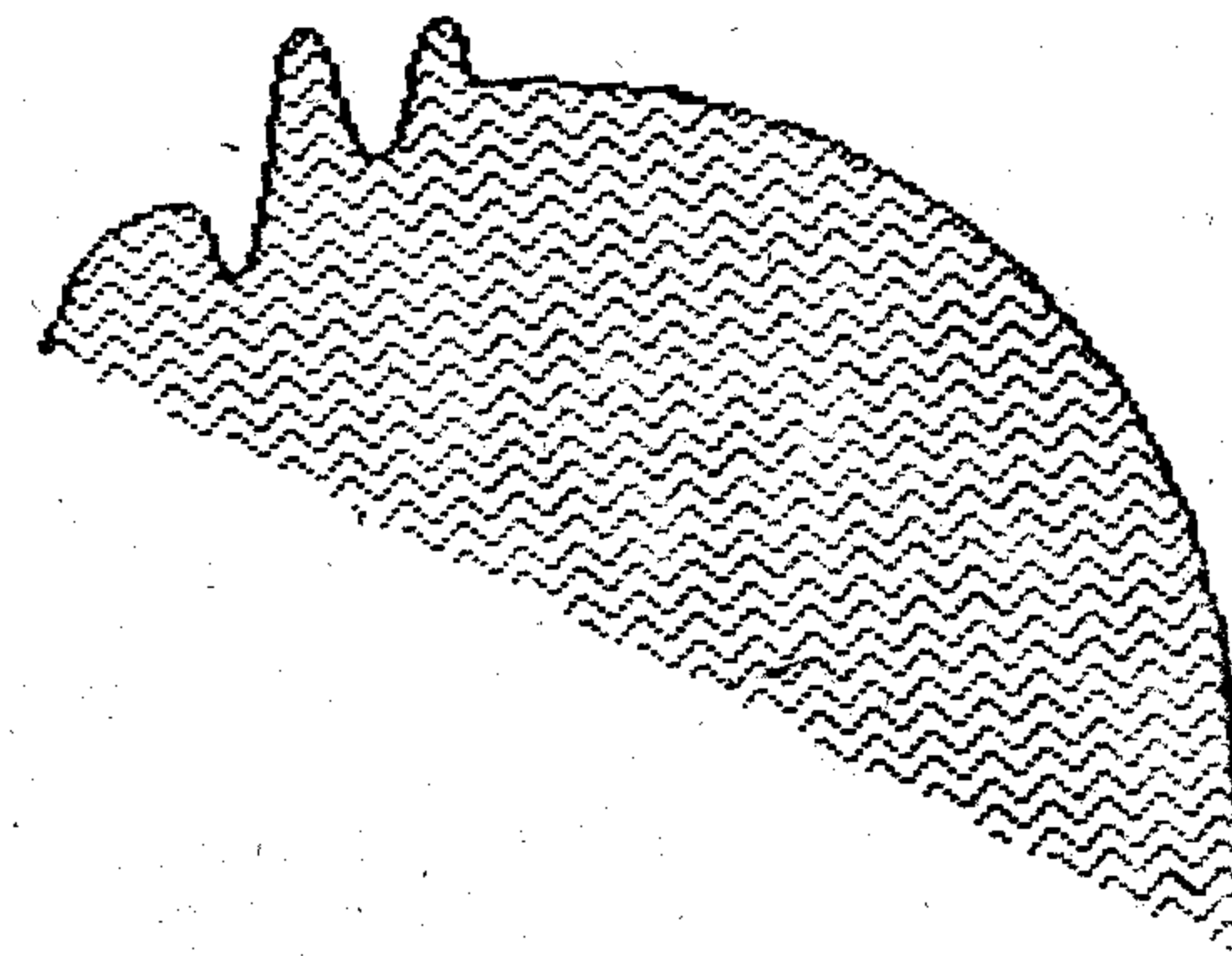
Когда при рисовании сложной фигуры подается команда «нарисовать точку» или рисуется отрезок и т. д. с начальной точкой, не совпадающей с конечной точкой предыдущего примитива, образуется «невидимая» грань. Она образуется также, если конечная точка контура не совпадает с начальной. «Невидимая» грань выводится заполнителем, как будто бы была подана команда SETEDGE (0).

Примечание. При выполнении команд FILL, RFILL, BFILL, RBFILL граница закрашиваемой области не изменяется.

Ниже приведена программа на языке Си, рисующая сложную область, контур которой сформирован с помощью кривой, дуги, отрезка и точки. Примитивы, формирующие контур, заданы разными цветами, «невидимый» отрезок находится в том месте контура, который имеет границу, выведенную типом заполнителя (не обведен):

```
int coords[18] =
    {0,0,0,0,-5,10,-10,-20,-10,20,-10,-35,-10,10,-20,-20,-10,-50};
main()
{
    newpicture(2,0);
    dofill(1);
    setfiller(8);
    color(2);
    begfigure();
    line(300,70,300,110,1);
    rarc(0,0,100,0,-100,0,0,100,0);
    color(3);
    rcurve(9,coords);
    color(4);
    endfigure();
}
```

Данная программа формирует на экране дисплея следующее изображение:



Максимальное количество вершин в замкнутой фигуре не должно превышать 950, причем для окружности требуется от 32 до 64 «вершин» (в зависимости от радиуса), а для кривой — от 8 до 16 «вершин» на каждый ее участок.

В одном горизонтальном сечении не должно находиться более 150 граней фигуры.

При раскраске области сложность ее практически не ограничена, однако если даже будет исчерпана доступная память, то ничего страшного не случится: будет подан звуковой сигнал и заполнение остановится.

Форматы команд:

FILL :
D (104) (X₃, Y₃) + ICOLOR₄ (FILLER + 1)₁
RFILL :
E (105) (DX, DY)₃ (FILLER + 1)₁
BFILL :
F (106) (X₃, Y₃) + ICOLOR₄ (FILLER + 1)₁
RBFILL :
G (107) (DX, DY)₃ (FILLER + 1)₁
SETEDGE :
a (141) ONOFF₁
DOFILL :
b (142) ONOFF₁
BEGFIGURE :
t (164) | (41)
ENDFIGURE :
t (164) пробел (40)
NEWREGION :
t (164) " (42)

Для использования этих команд в программах на языке Паскаль необходимы следующие объявления:

```
PROCEDURE FILL(X,Y,ICOLOR,FILLER:INTEGER); EXTERNAL;  
PROCEDURE RFILL(DX,DY,FILLER:INTEGER); EXTERNAL;  
PROCEDURE BFILL(X,Y,ICOLOR,FILLER:INTEGER); EXTERNAL;  
PROCEDURE RBFILL(DX,DY,FILLER:INTEGER); EXTERNAL;  
PROCEDURE SETEDGE(ONOFF:INTEGER); EXTERNAL;  
PROCEDURE DOFILL(ONOFF:INTEGER); EXTERNAL;  
PROCEDURE BEGFIGURE; EXTERNAL;  
PROCEDURE ENDFIGURE; EXTERNAL;  
PROCEDURE NEWREGION; EXTERNAL;
```

Из программы на языке Паскаль процедуры вызываются операторами:

```
FILL(X,Y,ICOLOR,FILLER);  
RFILL(DX,DY,FILLER);
```

```
BFILL(X,Y,ICOLOR,FILLER);  
RBFILL(DX,DY,FILLER);  
DOFILL(ONOFF);  
SETEDGE(ONOFF);  
BEGFIGURE;  
ENDFIGURE;  
NEWREGION;
```

Из Фортрана :

```
CALL FILL(IX,IY,ICOLOR,IFILL)  
CALL RFILL(IDX,IDY,IFILL)  
CALL BFILL(IX,IY,ICOLOR,IFILL)  
CALL RBFILL(DX,DY,IFILL)  
CALL DOFILL(IONOFF)  
CALL SETEDG(IONOFF)  
CALL BEGFIG  
CALL ENDFIG  
CALL NEWREG
```

На языке Си :

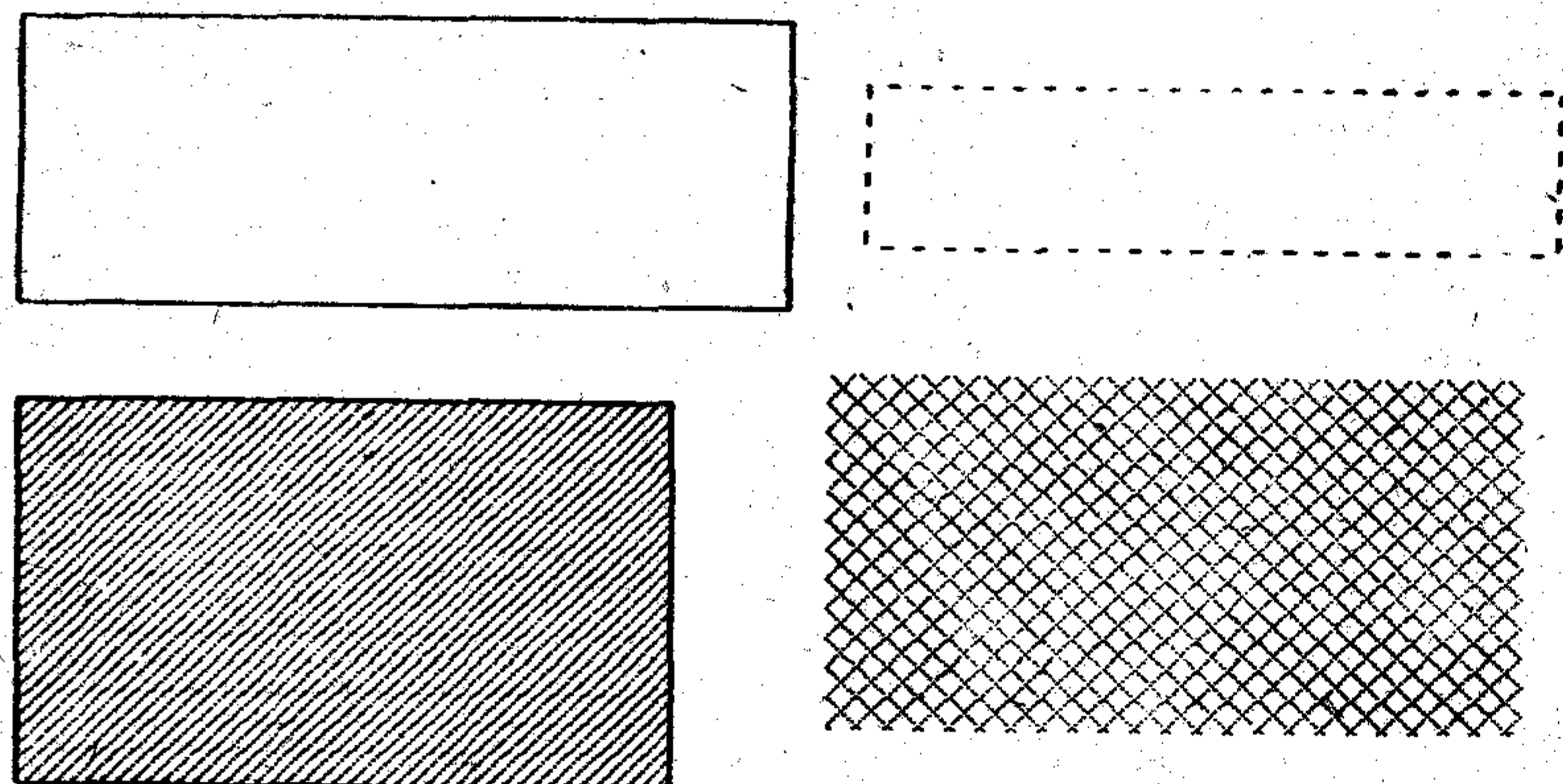
```
int x,y,dx,dy,icolor,filler;  
int onoff;  
fill(x,y,icolor,filler);  
rfill(dx,dy,filler);  
bfill(x,y,icolor,filler);  
rbfill(dx,dy,filler);  
dofill(onoff);  
setedge(onoff);  
begfigure();  
endfigure();  
newregion();
```

12. ПОСТРОЕНИЕ ПРЯМОУГОЛЬНИКОВ, МНОГОУГОЛЬНИКОВ, ЛОМАНЫХ ЛИНИЙ, ПОЛИМАРКЕРОВ

В графическом протоколе имеются команды RRECT и RRECT, посредством которых можно строить прямоугольники. В команде RRECT в качестве одного из углов принимается текущая позиция, а второй угол (по диагонали от первого) задается смещением по координатам X и Y. После вывода прямоугольника текущая позиция устанавливается на вторую заданную точку.

Если перед подачей команд `RECT` и `RRECT` был установлен режим заполнения фигур (командой `DOFILL` с аргументом 1), то будут нарисованы прямоугольники, заполненные штриховкой или узором (устанавливается командой `SETFILLER`). При выводе штриховки будет использован текущий цвет, установленный командой `COLOR`. Вид границ будет зависеть от режима, устанавливаемого командой `SETEDGE`. После команды `SETEDGE` с аргументом 1 прямоугольник будет обведен сплошной линией с цветом `ICOLOR` (может быть задан только в команде `RECT`, по команде `RRECT` обводка выполняется текущим цветом).

Примеры построения прямоугольников показаны на рисунке.

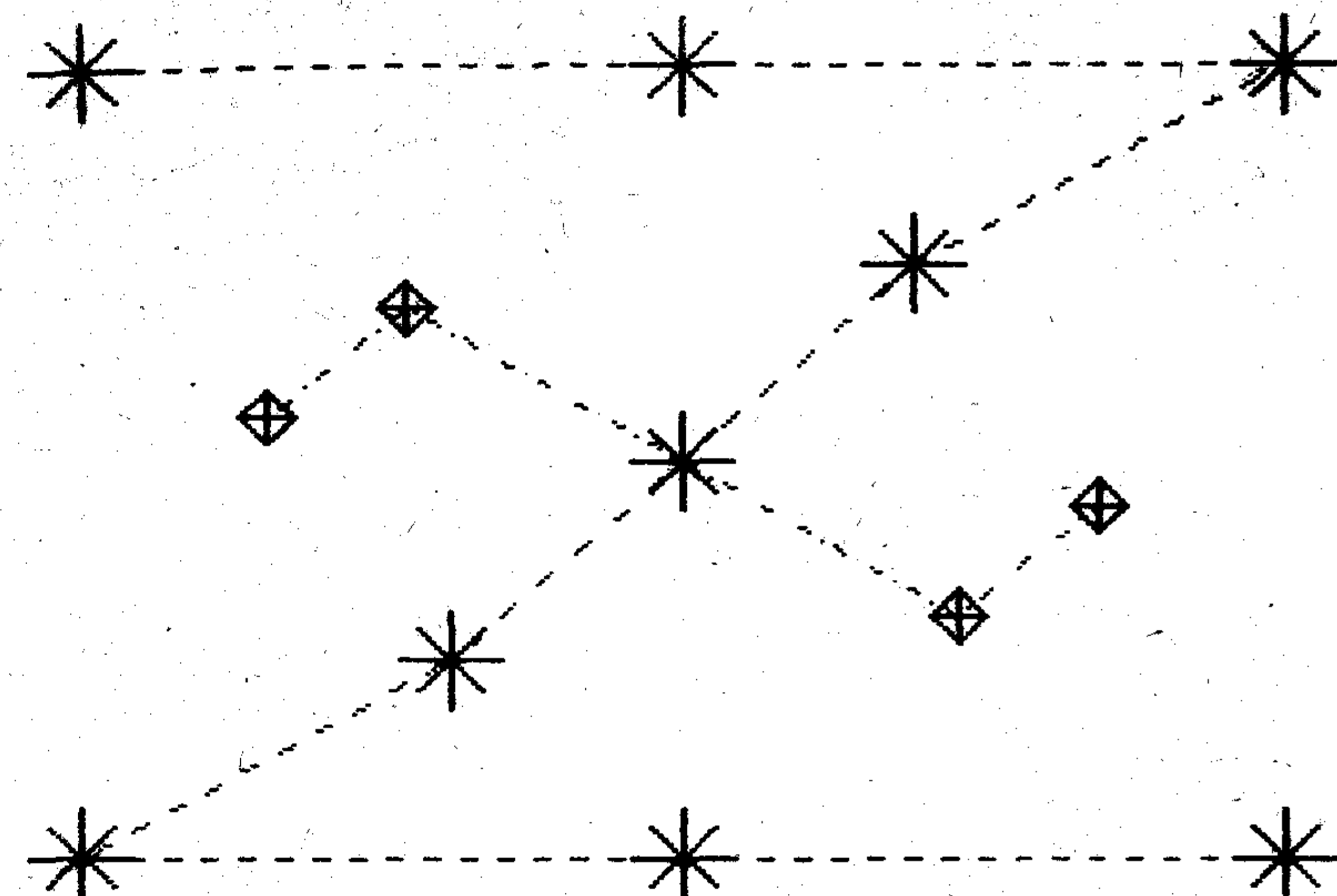


Ломаные линии с произвольным числом вершин строятся командами `POLYLINE` и `RPOLYLINE`. В первой команде задаются координаты вершин. Команда `RPOLYLINE` строит ломаную, координаты вершин которой заданы в виде смещений от позиции предыдущей вершины. Координаты первой вершины задаются смещением от текущей позиции. После построения ломаной текущая позиция перемещается в последнюю вершину. Это же правило действует и для всех аналогичных команд, работающих с массивом координат.

Команды `POLYMARKER` и `RPOLYMARKER` выводят последовательность маркеров, координаты которых заданы через массив. Команда `RPOLYMARKER` строит последователь-

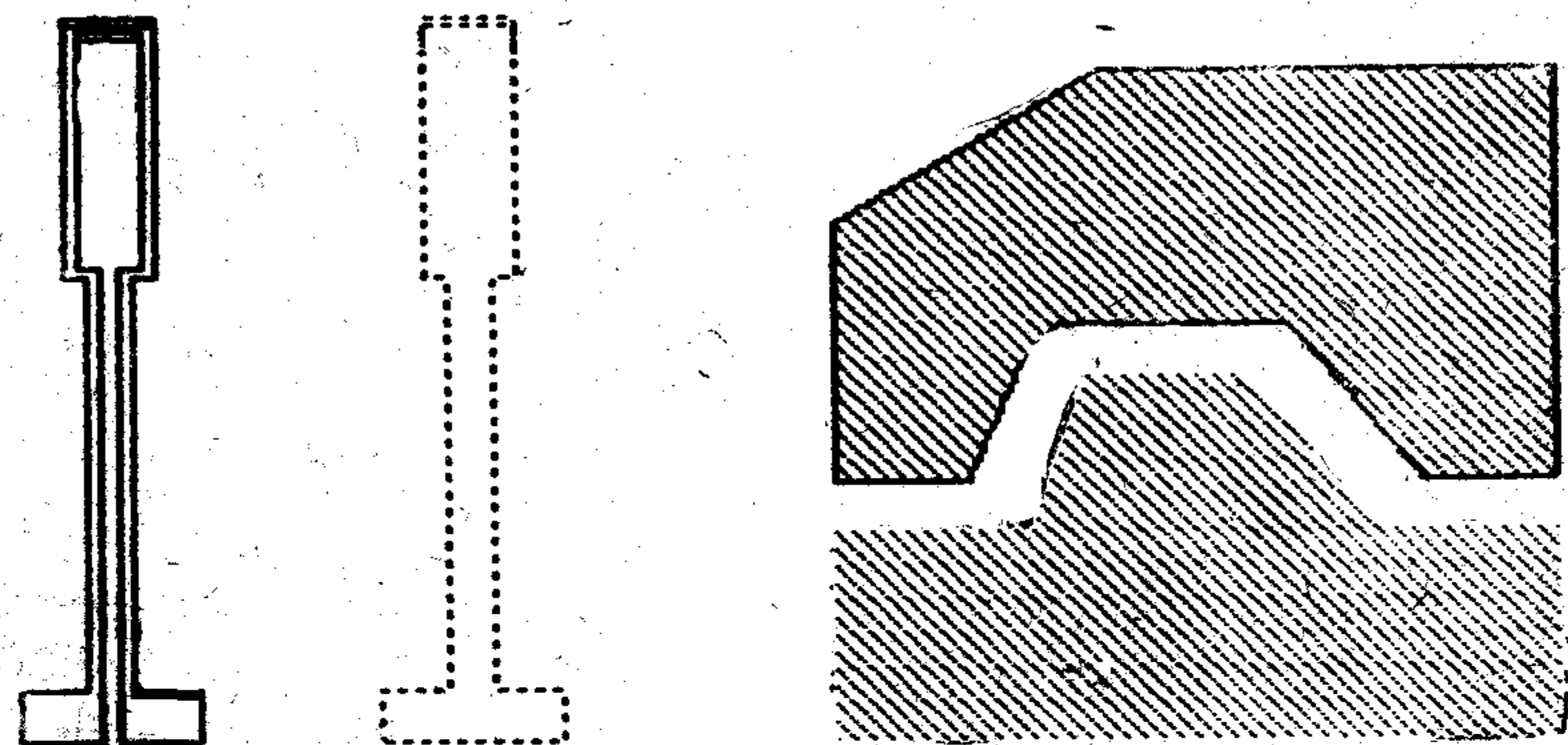
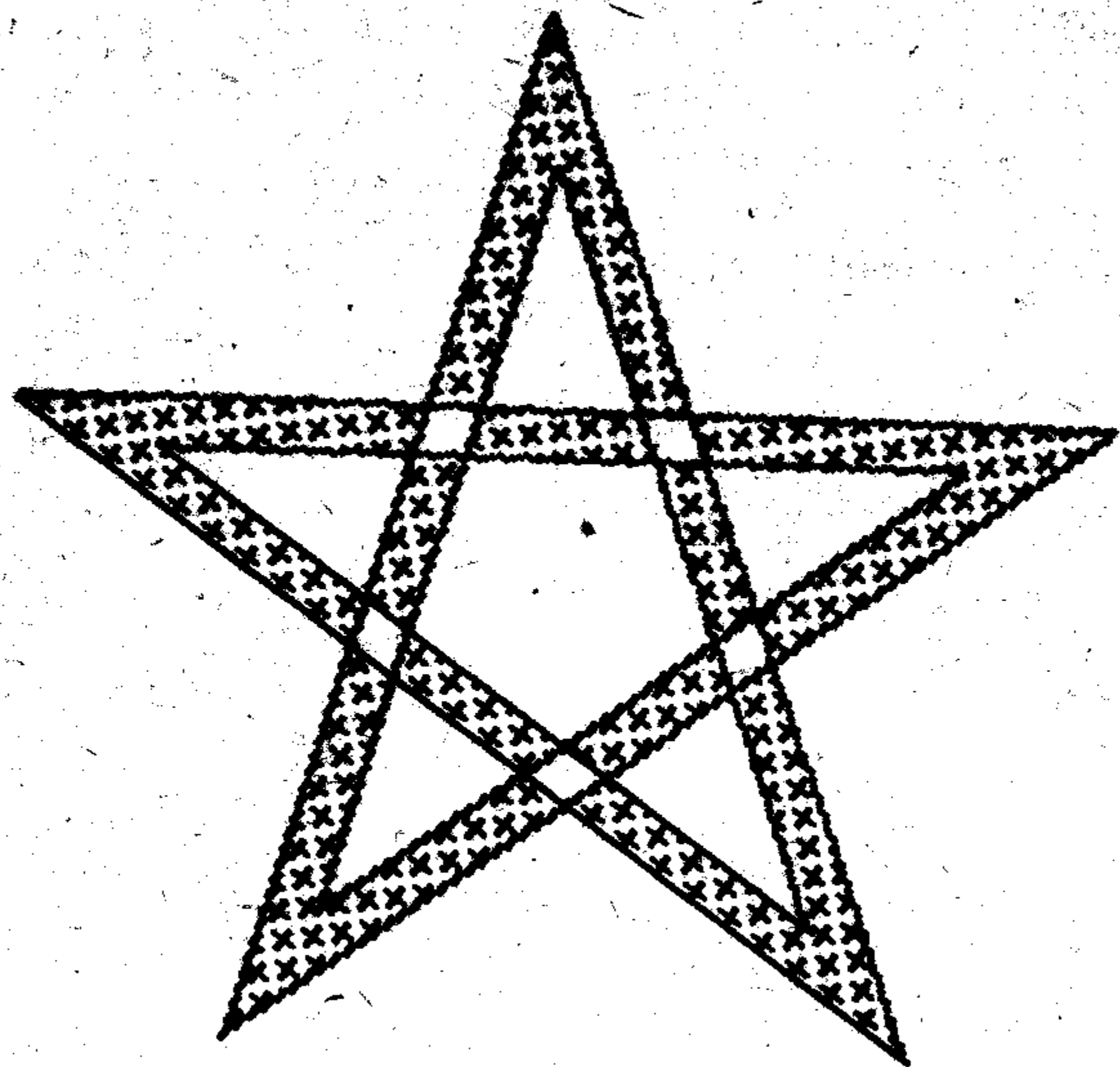
ность маркеров, координаты которых заданы в виде смещений от позиции предыдущего маркера. Для первого маркера задается смещение от текущей позиции.

Пример рисования ломаной линии (`POLYLINE`), в вершинах которой выведены маркеры (`POLYMARKER`), приведен на рисунке:



Команды `POLYGN` и `RPOLYGN` рисуют многоугольник или набор несвязанных многоугольников (`POLYGON SET`). Если командой `DOFILL` был установлен режим заполнения (`DOFILL(1)`), то рисуется многоугольник, заполненный штриховкой или узором, что устанавливается командой `SETFILLER`. При построении заполненного многоугольника, имеющего самопересечения, и заполненного набора многоугольников, области заполняются по принципу «четный-нечетный», как показано на рисунке.

В зависимости от режима представления границы, установленного командой `SETEDGE`, граница многоугольника рисуется либо сплошной линией с цветом, заданным в команде, либо многоугольник полностью рисуется заданным видом штриховки, как показано на рисунке:



При заполнении многоугольника для вывода штриховки используется текущий индекс цвета, установленный командой COLOR, а граница рисуется цветом ICOLOR, указанным в команде.

При рисовании штриховки и границы многоугольника запись информации в видео — ОЗУ производится с учетом текущего режима вывода цвета, установленного командой DRAWMODE.

Заполнение производится посредством специальной процедуры, которая определяет границы заполняемой области без учета прежней информации, находившейся на месте изображаемого многоугольника, в отличие от процедур FILL, BFILL.

Формат команд:

RECT :
& (46) (X1₃, Y1₃) + ICOLOR₄ X2₃ Y2₃

RRECT :
' (47) (DX, DY)₃

POLYLINE :
B (102) N₃ (X1₃, Y1₃) + ICOLOR₄ ... (XN₃, YN₃) + ICOLOR₄

Здесь N — число пар координат. Изображаемая ломаная будет иметь N-1 ребер.

RPOLYLINE :
C (103) N₃ (DX1, DY1)₃ ... (DXN, DYN)₃

POLYMARKER :
N (116) N₃ (X1₃, Y1₃) + ICOLOR₄ ... (XN₃, YN₃) + ICOLOR₄

• Здесь N — число пар координат.

RPOLYMARKER :
Q (117) N₃ (DX1, DY1)₃ ... (DXN, DYN)₃

POLYGN :
@ (100) N₃ + (SET*2)₂ X1₃ Y1₃ ... (XN₃, YN₃) + ICOLOR₄

RPOLYGN :
A (101) N₃ + (SET*2)₂ (DX1, DY1)₃ ... (DXN, DYN)₃

Здесь N — число пар координат, многоугольник будет иметь столько же ребер. Двухбитовый параметр, передаваемый вместе с числом N, признак набора многоугольников (бит 2). Бит 2, равный 1, говорит о том, что команда POLYGN содержит еще один список пар координат формата:

N₃ + (SET*2)₂ X1₃ Y1₃ ... (XN₃, YN₃) + ICOLOR₄

В команде RPOLYGN добавляется такой список:
N₃ + (SET*2)₂ (DX1, DY1)₃ ... (DXN, DYN)₃

Таким образом может быть передано произвольное число списков. В последнем списке координат признак SET (бит 2 параметра) должен быть равен 0.

Для рисования прямоугольников, ломаных линий, много-

угольников, полимаркеров, в программах на языке Паскаль необходимы следующие объявления:

```
TYPE COORDINATES=ARRAY[1..N] OF INTEGER;
PROCEDURE RECT(X1,Y1,X2,Y2,ICOLOR:INTEGER); EXTERNAL;
PROCEDURE RRECT(DX,DY:INTEGER); EXTERNAL;
PROCEDURE
  POLYLN(N:INTEGER;VAR COORD:COORDINATES;
        ICOLOR:INTEGER); EXTERNAL;
PROCEDURE
  RPOLYLN(N:INTEGER;VAR DCOORD:COORDINATES);
  EXTERNAL;
PROCEDURE
  POLYMARKER(N:INTEGER;VAR COORD:COORDINATES;
            ICOLOR:INTEGER); EXTERNAL;
PROCEDURE
  RPOLYMARKER(N:INTEGER;VAR DCOORD:COORDINATES);
  EXTERNAL;
PROCEDURE
  POLYGN(N:INTEGER;VAR COORD:COORDINATES;
        ICOLOR,POLSET:INTEGER); EXTERNAL;
PROCEDURE
  RPOLYGN(N:INTEGER;VAR COORD:COORDINATES;
        POLSET:INTEGER); EXTERNAL;
```

В описании типа COORDINATES вместо N должно стоять конкретное число (максимальное число вершин*2).

Из программы на языке Паскаль процедуры вызываются операторами:

```
RECT(X1, Y1, X2, Y2, ICOLOR);
RRECT(DX, DY);
POLYLN(N DIV 2,COORD, ICOLOR);
RPOLYLN(N DIV 2,DCOORD);
POLYMARKER(N DIV 2,COORD, ICOLOR);
RPOLYMARKER(N DIV 2,DCOORD);
POLYGN(N DIV 2,COORD, ICOLOR, POLSET);
RPOLYGN(N DIV 2,COORD, POLSET);
```

На языке Фортран :

```
CALL RECT(IX1, IY1, IX2, IY2, ICOLOR)
CALL RRECT(IDX, IDY)
INTEGER COORD(N) ! вместо N должно стоять число
CALL POLYLN(N/2, COORD, ICOLOR)
CALL RPOLYLN(N/2, COORD)
CALL POLYMARKER(N/2, COORD, ICOLOR)
CALL RPOLYMARKER(N/2, COORD)
CALL POLYGN(N/2, COORD, ICOLOR, ISET)
CALL RPOLYGN(N/2, COORD, ISET)
```

На языке Си :

```
int x1,y1,x2,y2,icolor,dx,dy;
int n,*coord,polset;
```

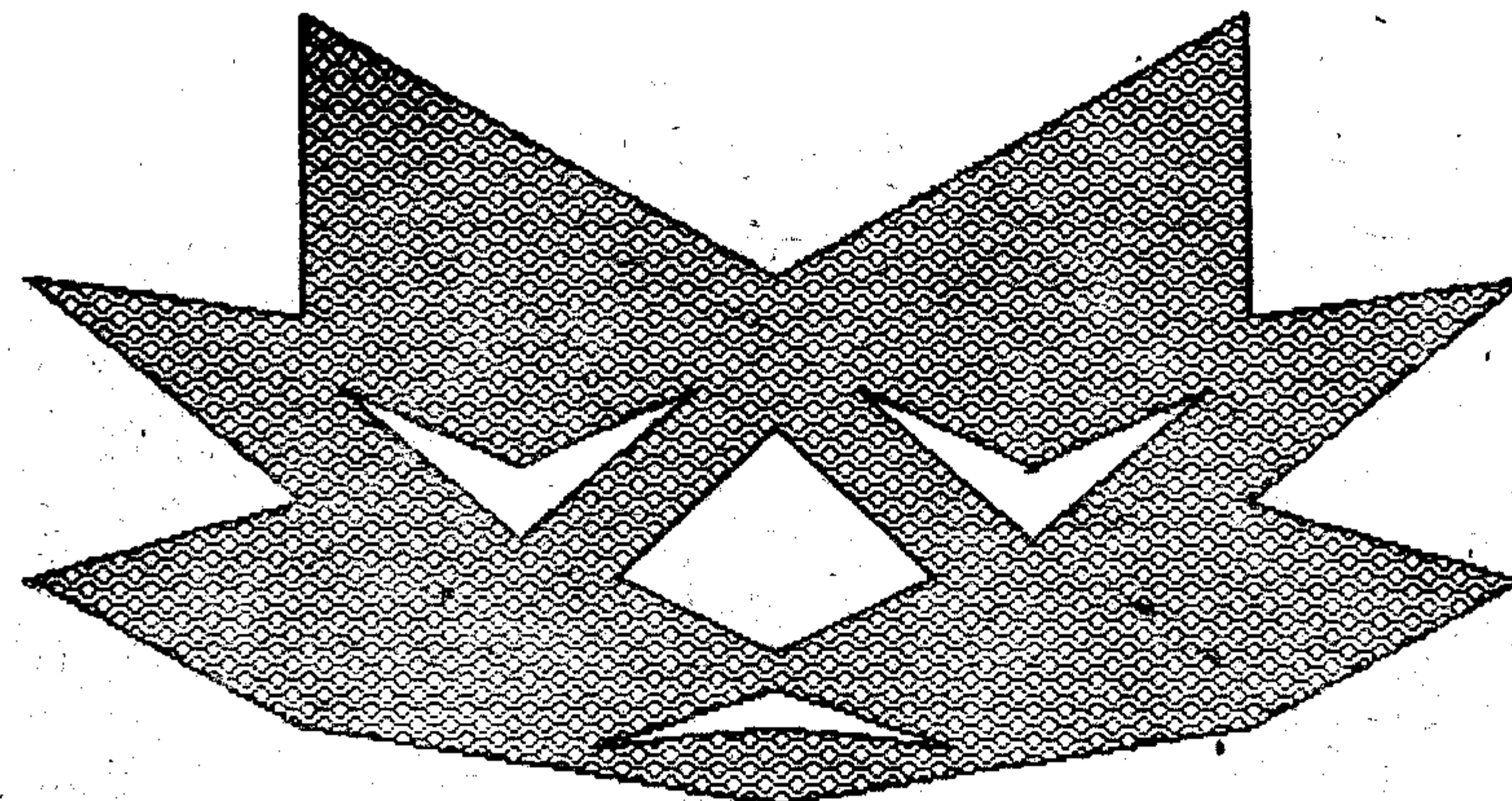
```
rect(x1,y1,x2,y2,icolor);
rrect(dx,dy);
polyln(n,coord,icolor);
rpolyln(n,coord);
polymarker(n,coord,icolor);
rpolymarker(n,coord);
polygn(n,coord,icolor,polset);
rpolygn(n,coord,polset);
```

В массиве COORD координаты располагаются в порядке X0, Y0, X1, Y1, ..., Xn, Yn.

Параметр POLSET процедур POLYGN и RPOLYGN служит для задания набора многоугольников. Для построения простого многоугольника он должен иметь значение 0, в этом случае процедура POLYGN вызывается только один раз. Для построения набора многоугольников процедура POLYGN вызывается несколько раз с разными значениями параметра POLSET, и каждый раз задаются координаты вершин одного многоугольника набора. При первом вызове параметр должен иметь значение 2, при последующих вызовах — 3, при вызове в последний раз — 1, например:

```
POLYGN(N1,TABXY1,ICOLOR,2);
POLYGN(N2,TABXY2,ICOLOR,3);
POLYGN(N3,TABXY3,ICOLOR,3);
POLYGN(N4,TABXY4,ICOLOR,1);
```

На иллюстрации показан пример заполнения набора многоугольников:



13. РИСОВАНИЕ ПЛАВНЫХ КРИВЫХ

В графическом протоколе существует возможность рисования плавных кривых, задаваемых набором точек, через которые они проходят. Для задания кривой требуется не менее четырех точек. Каждый участок кривой определяется по четырем соседним точкам, причем рисуется участок между двумя средними точками.

Кривая может быть разомкнутой (CURVE, RCURVE) и замкнутой (CCURVE, RCCURVE).

При рисовании разомкнутой кривой следует учесть, что участки кривой между первой и второй, а также предпоследней и последней точками не рисуются. Эти точки используются при определении формы соседних с ними участков кривой. Если нужно, чтобы кривая рисовалась и через эти точки, то самым простым способом будет дублирование первой и последней точки при задании кривой.

При рисовании замкнутой кривой последняя точка автоматически соединяется с первой.

При рисовании кривых используются следующие правила:

- кривая проходит через все заданные точки;
- кривая имеет в точности ту же самую форму, вне зависимости от того, задана ли она от первой точки к последней или от последней к первой;
- касательная к кривой в данной точке параллельна линии, проведенной через две другие точки, лежащие по каждую сторону от данной точки.

Формат задания разомкнутой кривой аналогичен формату задания ломаной линии, а формат замкнутой — многоугольнику.

Команды рисования кривых могут быть использованы для указания заштрихованной области таким же образом, как и при рисовании многоугольников и ломаных. Однако следует учесть, что при заполнении на каждый участок кривой записывается от 8 до 16 точек.

Формат команд:

CURVE :
B (102) $N_3 + (1)_2 X1_3 Y1_3 \dots (XN_3, YN_3) + ICOLOR_4$

Здесь N — число пар координат. Изображаемая кривая будет иметь N—2 участков.

RCURVE :
C (103) $N_3 + (1)_2 (DX1, DY1)_3 \dots (DXN, DYN)_3$

CCURVE :
E (100) $N_3 + (SET*2+1)_2 X1_3 Y1_3 \dots (XN_3, YN_3) + ICOLOR_4$

RCCURVE :
A (101) $N_3 + (SET*2+1)_2 (DX1, DY1)_3 \dots (DXN, DYN)_3$

Здесь N — число пар координат, кривая будет иметь столько же участков. Двухбитовый параметр, передаваемый вместе с числом N, содержит признак отличия кривой от ломаной (бит 1) и признак набора кривых (бит 2). Бит 2, равный 1, говорит о том, что команда CCURVE содержит еще один список пар координат формата:

$N_3 + (SET*2)_2 X1_3 Y1_3 \dots (XN_3, YN_3) + ICOLOR_4$

В команде RCCURVE добавляется такой список:

$N_3 + (SET*2)_2 (DX1, DY1)_3 \dots (DXN, DYN)_3$

Таким образом может быть передано произвольное число списков. В последнем списке координат признак SET (бит 2 параметра) должен быть равен 0.

Для рисования кривых в программах на языке Паскаль необходимы следующие объявления:

TYPE COORDINATES=ARRAY[1..N] OF INTEGER;

PROCEDURE

CURVE(N:INTEGER;VAR COORD:COORDINATES;
ICOLOR:INTEGER); EXTERNAL;

PROCEDURE

RCURVE(N:INTEGER;VAR DCOORD:COORDINATES);

EXTERNAL;

PROCEDURE

CCURVE(N:INTEGER;VAR COORD:COORDINATES;
ICOLOR,CURSET:INTEGER); EXTERNAL;

PROCEDURE

RCCURVE(N:INTEGER;VAR COORD:COORDINATES;
CURSET:INTEGER); EXTERNAL;

В описании типа COORDINATES вместо N должно стоять конкретное число (максимальное число точек*2).

Из программы на языке Паскаль процедуры вызываются операторами:

CURVE(N DIV 2,COORD,ICOLOR);

RCURVE(N DIV 2,DCOORD);

CCURVE(N DIV 2,COORD,ICOLOR,CURSET);

RCCURVE(N DIV 2,COORD,CURSET);

На языке Фортран :

INTEGER COORD(N) ! вместо N должно стоять число
CALL CURVE(N/2,COORD,ICOLOR)

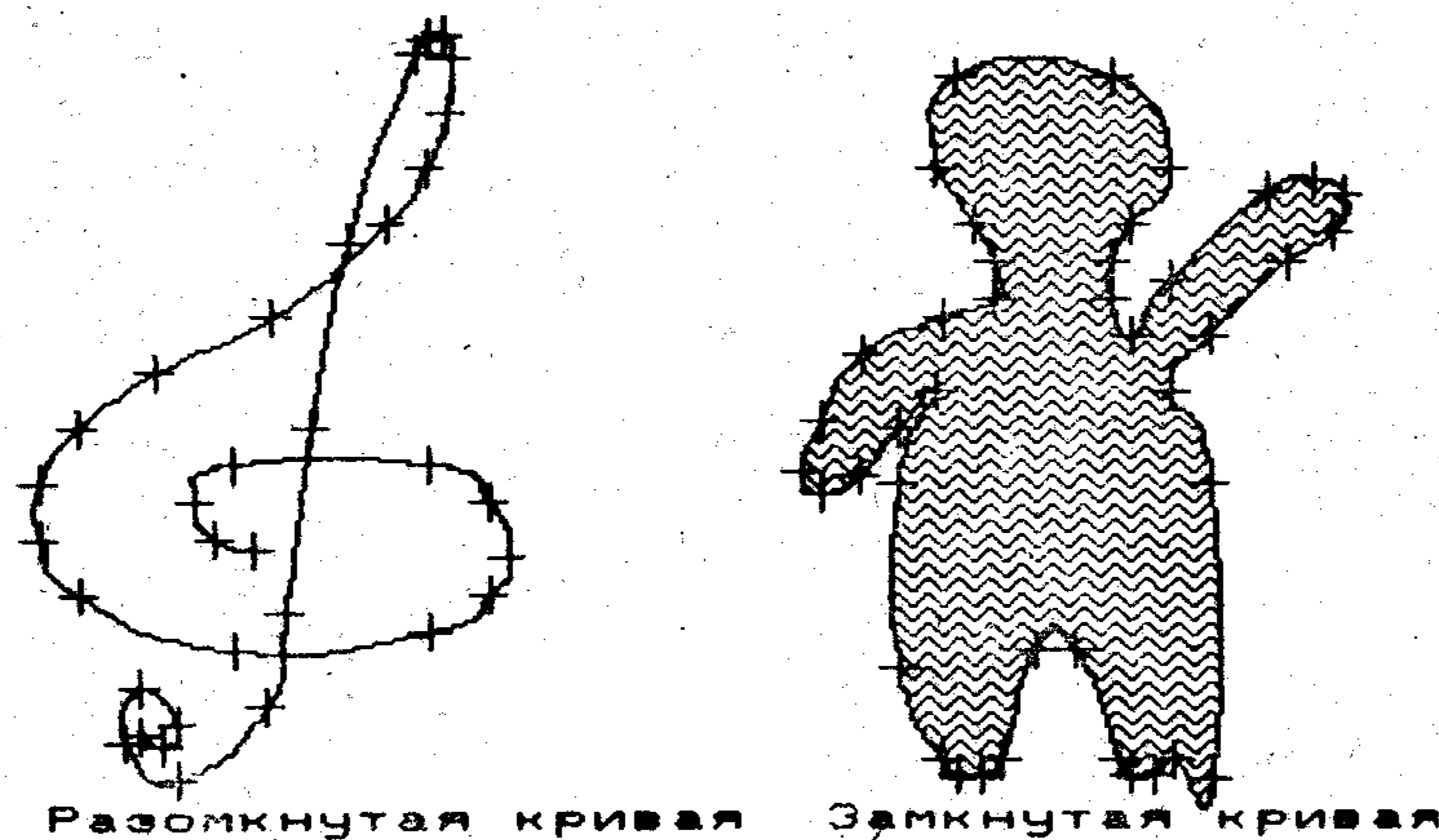
```
CALL RCURVE(N/2,COORD)
CALL CCURVE(N/2,COORD,ICOLOR,ISET)
CALL RCCURV(N/2,COORD,ISET)
```

На языке Си :

```
int icolor,n,*coord,curset;
curve(n,coord,icolor);
rcurve(n,coord);
ccurve(n,coord,icolor,curset);
rccurve(n,coord,curset);
```

В массиве COORD координаты располагаются в порядке $X_0, Y_0, X_1, Y_1, \dots, X_n, Y_n$.

Ниже приведен пример рисования разомкнутой и замкнутой заполненной кривых. Вершины, по которым проводилось построение, отмечены маркерами.



Параметр CURSET процедур CCURVE и RCCURVE служит для задания набора кривых. Для построения простой кривой он должен иметь значение 0, в этом случае процедура CCURVE вызывается только один раз. Для построения набора замкнутых кривых процедура CCURVE вызывается несколько раз с разными значениями параметра CURSET, и каждый раз задаются координаты вершин одной кривой из набора. При первом вызове параметр должен иметь значение 2, при последующих вызовах — 3, при вызове в последний раз — 1.

Параметр ICOLOR во всех вызовах, кроме последнего, игнорируется.

14. РИСОВАНИЕ ОКРУЖНОСТЕЙ, ЭЛЛИПСОВ, ИХ ДУГ, СЕГМЕНТОВ, СЕКТОРОВ

В графическом протоколе имеются команды построения окружностей, эллипсов, их дуг, секторов, сегментов, которые могут быть заполнены штриховкой или узором, в зависимости от режима, установленного командой DOFILL, и вида заполнителя, заданного командой SETFILLER. Радиусы окружностей и эллипсов задаются в мировых координатах, так же как и координаты центра, начала дуг и т. п. Из-за преобразования координат при несовпадении коэффициентов пересчета по вертикали и горизонтали окружность может превратиться в эллипс и наоборот. Для построения окружности указывается только один радиус, а для построения эллипса указываются два радиуса — по оси X и по оси Y.

На иллюстрациях показаны примеры рисования окружностей, эллипсов, их дуг, секторов и сегментов.

Команда рисования окружности по центру и радиусу называется CIRCLE. Для рисования окружности с заданием координат центра смещением от текущей позиции используется команда RCIRCLE. После выполнения команды CIRCLE текущая позиция устанавливается на центр нарисованной окружности, таким образом, последующими командами RCIRCLE с параметрами DXC=0, DYC=0 можно дополнительно нарисовать произвольное число концентрических окружностей. Так как при построении окружностей на дискретном растре точки, лежащие на окружности одного радиуса, могут быть несмежны с точками окружностей меньшего или большего радиуса, то выведенный набор концентрических окружностей не будет представлять собой кольцо. На нем будут видны точки, на которые не выводились окружности. Поэтому для построения колец лучше использовать две концентрические окружности, а пространство между ними заполнять с помощью команд заполнения области.

Окружность может задаваться и другими способами: по центру и точке на окружности или по трем точкам на окружности. Команда рисования окружности по центру и точке называется CIRC2 и RCIRC2, а по трем точкам — CIRC3 и RCIRC3. Если заданные точки лежат на одной прямой или совпадают, то построение окружности не производится.

При выполнении команд CIRC2 и RCIRC2 текущая позиция также устанавливается в центр окружности. В команде RCIRC2 задается смещение от текущей точки до центра, а затем — от центра до точки на окружности. После выполнения команд CIRC3 и RCIRC3 текущая позиция устанавливается в третью точку окружности. В команде RCIRC3 задаются смещения первой точки от текущей, затем второй от первой и третьей от второй.

Для рисования эллипса подается команда ELLIPSE, в которой задается не один, а два радиуса — по оси X и по оси Y. Радиусы, как и координаты центра, задаются в мировых координатах, и при выполнении преобразования их к экранным пропорции эллипса могут измениться. В команде RELLIPSE задается смещение центра относительно координат текущей позиции. После выполнения команд ELLIPSE и RELLIPSE текущая позиция устанавливается на центр нарисованного эллипса, таким образом, последующими командами RELLIPSE (с указанием $DX=0$, $DY=0$) можно дополнительно нарисовать произвольное число концентрических эллипсов. По этой команде рисуется эллипс с осями, параллельными осям координат.

Эллипс может быть задан при помощи указания центра и двух точек, соответствующих концам двух его сопряженных диаметров. В этом случае может быть нарисован наклонный эллипс. Для этого используются команды ELLI2 и RELLI2. По окончании выполнения этих команд текущая позиция также остается в центре эллипса. В команде RELLI2 координаты концов его сопряженных диаметров задаются относительно центра, а не друг относительно друга.

Если командой DOFILL включен режим заполнения примитивов, то нарисованная окружность или эллипс заполняются текущим типом штриховки или узором (задается командой SETFILLER). При заполнении штриховка выводится текущим цветом, установленным командой COLOR. Заполнение и рисование границы производится с учетом режима вывода и режима рисования границы (SETEDGE).

Наряду с рисованием полных окружностей и эллипсов, можно нарисовать дугу, а также сегмент (начало и конец дуги соединены хордой) и сектор (дополнительно два отрезка из центра в концы дуги). Команды типа ARC рисуют дугу, сектор и сегмент окружности, а EARC рисуют дугу, сектор и сегмент эллипса. В командах, начинающихся с R, координаты начала, центра и конца дуги задаются в виде смещений. В командах построения дуг имеется параметр

CW — направление рисования дуги: 1 — по часовой стрелке, 0 — против, и параметр PIE — признак дуги, сектора, сегмента: 0 — рисуется дуга, 1 — сектор, 2 — сегмент.

При рисовании дуги командами RARC, REARC и т. д. точка в текущей позиции не ставится. Причины, по которым это делается, были изложены при описании команды LINE. После вывода дуги текущая позиция перемещается в ее конечную точку.

Дуга может задаваться одним из трех способов:

— заданием начальной точки дуги, радиуса (радиусов для эллипса), центра и конечной точки дуги (ARC, RARC, EARC, REARC);

— заданием координат центра дуги, радиуса дуги для окружности или координат двух сопряженных диаметров для эллипса и смещений относительно центра до двух точек, задающих лучи, вырезающие на окружности (эллипсе) требуемую дугу (ARC2, RARC2, EARC2, REARC2);

— заданием координат трех точек: начальной, промежуточной и конечной. В этом случае может быть нарисована только дуга окружности, причем параметр CW не используется — дуга всегда рисуется от первой точки к третьей через вторую (ARC3, RARC3).

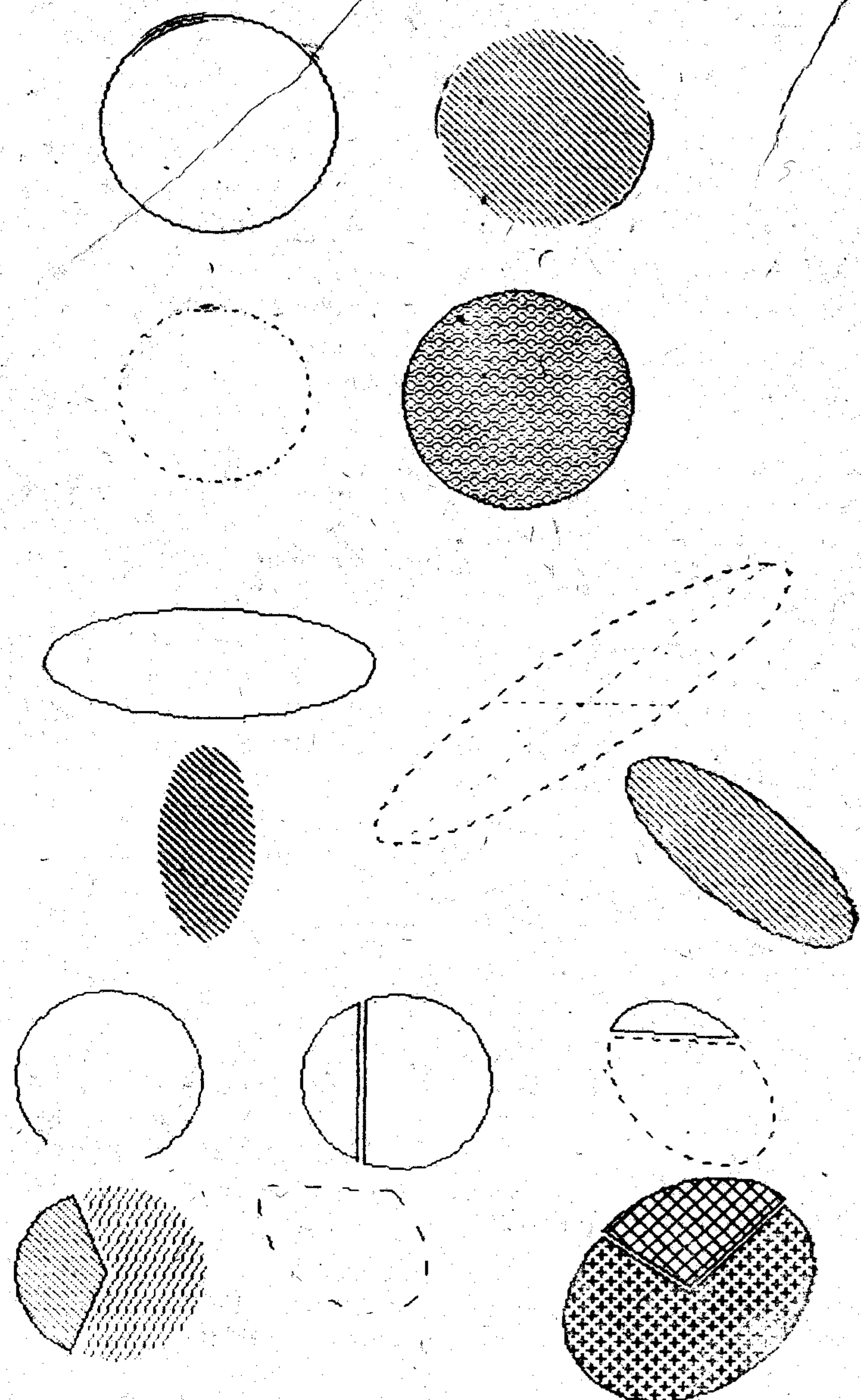
При задании дуги эллипса с сопряженными диаметрами параметр CW не требуется — дуга рисуется по направлению от первого диаметра ко второму.

В командах ARC, RARC, EARC, REARC текущая точка устанавливается в заданную конечную точку дуги. Относительные координаты отсчитываются в том же порядке, как они указаны в вызове подпрограммы.

В командах ARC2, RARC2, EARC2, REARC2 текущая точка устанавливается в центр. В этих командах все координаты задаются относительно центра, за исключением команды EARC2, в которой концы сопряженных диаметров задаются в абсолютных координатах.

В командах ARC3 и RARC3 текущая точка устанавливается в конечную (третью) точку дуги. В команде RARC3 координаты отсчитываются в том же порядке, как они указаны в вызове.

На иллюстрации приведены примеры рисования различных окружностей, эллипсов, их дуг, сегментов и секторов:



Команды имеют следующий формат:

- CIRCLE :
 ((50) R₃ (XC₃,YC₃) +ICOLOR₄
- RCIRCLE:
) (51) R₃ (DXC,DYC)₃
- CIRC2 :
 h (150) (XC₃,YC₃) +ICOLOR₄ X1₃ Y1₃
- RCIRC2 :
 i (151) (DXC,DYC)₃ (DX1,DY1)₃
- CIRC3 :
 j (152) (X1₃,Y1₃) +ICOLOR₄ X2₃ Y2₃ X3₃ Y3₃
- RCIRC3 :
 k (153) (DX1,DY1)₃ (DX2,DY2)₃ (DX3,DY3)₃
- ELLIPSE :
 * (52) RX₃ RY₃ (XC₃,YC₃) +ICOLOR₄
- RELLIPSE :
 + (53) RX₃ RY₃ (DXC,DYC)₃
- ELLI2 :
 r (162) (XC₃,YC₃) +ICOLOR₄ XR1₃ YR1₃ XR2₃ YR2₃
- RELLI2 :
 s (163) (DXC,DYC)₃ (DXR1,DYR1)₃ (DXR2,DYR2)₃
- ARC :
 , (54) R₃ (X1₃,Y1₃) +ICOLOR₄ XC₃ YC₃ (X2₃,Y2₃) + (CW*4+PIE)₄
- RARC :
 - (55) R₃ (DX1,DY1)₃ (DXC,DYC)₃ (DX2,DY2)₃ (CW*4+PIE)₁
- EARC :
 . (56) RX₃ RY₃ (X1₃,Y1₃) +ICOLOR₄
 XC₃ YC₃ (X2₃,Y2₃) + (CW*4+PIE)₄
- REARC :
 / (57) RX₃ RY₃ (DX1,DY1)₃
 (DXC,DYC)₃ (DX2,DY2)₃ (CW*4+PIE)₁
- ARC2 :
 l (154) (XC₃,YC₃) +ICOLOR₄ R₃+CW₂ DX1₃ DY1₃ DX2₃ DY2₃ PIE₁
- RARC2 :
 m (155) (DXC,DYC)₃ R₃+CW₂ (DX1,DY1)₃ (DX2,DY2)₃ PIE₁
- EARC2 :
 n (156) (XC₃,YC₃) +ICOLOR₄ XR1₃ YR1₃
 XR2₃ YR2₃ DX1₃ DY1₃ DX2₃ DY2₃ PIE₁
- REARC2 :
 o (157) (DXC,DYC)₃ (DXR1,DYR1)₃ (DXR2,DYR2)₃
 (DX1,DY1)₃ (DX2,DY2)₃ PIE₁
- ARC3 :
 p (160) (X1₃,Y1₃+ICOLOR₄ X2₃ Y2₃
 X3₃ Y3₃ PIE₁
- RARC3 :
 q (161) (DX1,DY1)₃ (DX2,DY2)₃ (DX3,DY3)₃ PIE₁

Здесь:

X1, Y1 — координаты начальной точки дуги;

X2, Y2 — координаты конечной (промежуточной для 3)

точки;

X3, Y3 — координаты конечной точки дуги для вызовов типа 3;

XC, YC — координаты центра;

DX1, DY1 — смещение от текущей позиции до начальной точки или до конца луча, задающего начало дуги;

DXC, DYC — смещение от начальной точки дуги до центра;

DX2, DY2 — смещение от центра дуги до ее конечной точки или до конца луча, задающего конец дуги;

Для построения окружностей, эллипсов, их дуг, сегментов и секторов в программах на языке Паскаль необходимы следующие объявления:

PROCEDURE CIRCLE (XC,YC,ICOLOR,R:INTEGER); EXTERNAL;

PROCEDURE RCIRCLE (DXC,DYC,R:INTEGER); EXTERNAL;

PROCEDURE CIRC2 (XC,YC,X1,Y1,ICOLOR:INTEGER); EXTERNAL;

PROCEDURE RCIRC2 (DXC,DYC,DX1,DY1:INTEGER); EXTERNAL;

PROCEDURE CIRC3 (X1,Y1,X2,Y2,X3,Y3,ICOLOR:INTEGER);

EXTERNAL;

PROCEDURE RCIRC3 (DX1,DY1,DX2,DY2,DX3,DY3:INTEGER);

EXTERNAL;

PROCEDURE ELLIPSE (XC,YC,ICOLOR,RX,RY:INTEGER); EXTERNAL;

PROCEDURE RELIPSE (DXC,DYC,RX,RY:INTEGER); EXTERNAL;

PROCEDURE ELLI2 (XC,YC,ICOLOR,XR1,YR1,XR2,YR2:INTEGER);

EXTERNAL;

PROCEDURE RELI2 (DXC,DYC,DXR1,DYR1,DXR2,DYR2:INTEGER);

EXTERNAL;

PROCEDURE

ARC (X1,Y1,R,CW,XC,YC,ICOLOR,X2,Y2,PIE:INTEGER);

EXTERNAL;

PROCEDURE

RARC (DX1,DY1,R,CW,DXC,DYC,DX2,DY2,PIE:INTEGER);

EXTERNAL;

PROCEDURE

EARC (X1,Y1,RX,RY,CW,XC,YC,ICOLOR,X2,Y2,PIE:INTEGER);

EXTERNAL;

PROCEDURE

REARC (DX1,DY1,RX,RY,CW,DXC,DYC,DX2,DY2,PIE:INTEGER);

EXTERNAL;

PROCEDURE

ARC2 (XC,YC,ICOLOR,R,CW,DX1,DY1,DX2,DY2,PIE:INTEGER);

EXTERNAL;

PROCEDURE

RARC2 (DXC,DYC,R,CW,DX1,DY1,DX2,DY2,PIE:INTEGER);

EXTERNAL;

PROCEDURE

EARC2 (XC,YC,ICOLOR,XR1,YR1,XR2,YR2,DX1,DY1,DX2,DY2,PIE:INTEGER); EXTERNAL;

PROCEDURE

REARC2 (DXC,DYC,DXR1,DYR1,DXR2,DYR2,DX1,DY1,DX2,DY2,PIE:INTEGER); EXTERNAL;

PROCEDURE

ARC3 (X1,Y1,X2,Y2,X3,Y3,ICOLOR,PIE:INTEGER);

EXTERNAL;

PROCEDURE

RARC3 (DX1,DY1,DX2,DY2,DX3,DY3,PIE:INTEGER);

EXTERNAL;

В программе на языке Паскаль эти процедуры выполняются операторами:

CIRCLE (XC,YC,ICOLOR,R);

RCIRCLE (DXC,DYC,R);

CIRC2 (XC,YC,X1,Y1,ICOLOR);

RCIRC2 (DXC,DYC,DX1,DY1);

CIRC3 (X1,Y1,X2,Y2,X3,Y3,ICOLOR);

RCIRC3 (DX1,DY1,DX2,DY2,DX3,DY3);

ELLIPSE (XC,YC,ICOLOR,RX,RY);

RELLIPSE (DXC,DYC,RX,RY);

ELLI2 (XC,YC,ICOLOR,XR1,YR1,XR2,YR2);

RELI2 (DXC,DYC,DXR1,DYR1,DXR2,DYR2);

ARC (X1,Y1,R,CW,XC,YC,ICOLOR,X2,Y2,PIE);

RARC (DX1,DY1,R,CW,DXC,DYC,DX2,DY2,PIE);

EARC (X1,Y1,RX,RY,CW,XC,YC,ICOLOR,X2,Y2,PIE);

REARC (DX1,DY1,RX,RY,CW,DXC,DYC,DX2,DY2,PIE);

ARC2 (XC,YC,ICOLOR,R,CW,DX1,DY1,DX2,DY2,PIE);

RARC2 (DXC,DYC,R,CW,DX1,DY1,DX2,DY2,PIE);

EARC2 (XC,YC,ICOLOR,XR1,YR1,XR2,YR2,

DX1,DY1,DX2,DY2,PIE);

REARC2 (DXC,DYC,DXR1,DYR1,DXR2,DYR2,

DX1,DY1,DX2,DY2,PIE);

ARC3 (X1,Y1,X2,Y2,X3,Y3,ICOLOR,PIE);

RARC3 (DX1,DY1,DX2,DY2,DX3,DY3,PIE);

На языке Фортран:

CALL CIRCLE (IXC,IYC,ICOLOR,IR)

CALL RCIRCLE (IDXC,IDYC,IR)

CALL CIRC2 (IXC,IYC,IX1,IY1,ICOLOR)

CALL RCIRC2 (IDXC,IDYC>IDX1>IDY1)

CALL CIRC3 (IX1,IY1,IX2,IY2,IX3,IY3,ICOLOR)

CALL RCIRC3 (IDX1>IDY1>IDX2>IDY2>IDX3>IDY3)

CALL ELLIPS (IXC,IYC,ICOLOR,IRX,IRY)

CALL RELIP (IDXC>IDYC>IDRX>IDRY)

CALL ELLI2 (IXC,IYC,ICOLOR,IXR1,IYR1,IXR2,IYR2)

CALL RELI2 (IDXC>IDYC>IDXR1>IDYR1>IDXR2>IDYR2)

CALL ARC (IX1,IY1,IR,ICW,IXC,IYC,ICOLOR,IX2,IY2,PIE)

CALL RARC (IDX1>IDY1,IR,ICW>IDXC>IDYC>IDX2>IDY2,PIE)

CALL EARC (IX1,IY1,IRX,IRY,ICW,IXC,IYC,

DX1,DY1,DX2,DY2,PIE);

REARC (DX1,DY1,IRX,IRY,ICW,IXC,IYC,

DX1,DY1,DX2,DY2,PIE);

ARC3 (IX1,IY1,IX2,IY2,IX3,IY3,ICOLOR,PIE);

RARC3 (IDX1>IDY1>IDX2>IDY2>IDX3>IDY3,PIE);

```

* ICOLOR,IX2,IY2,IPIE)
CALL REARC(IDX1,IDY1,IRX,IRY,ICW,IDX2,IDYC,
* IDX2,IDY2,IPIE)
CALL ARC2(IXC,IYC,ICOLOR,IR,ICW,IDX1,IDY1,
* IDX2,IDY2,IPIE)
CALL RARC2(IDXC,IDYC,IR,ICW,IDX1,IDY1,IDX2,IDY2,IPIE)
CALL EARC2(IXC,IYC,ICOLOR,IXR1,IYR1,IXR2,IYR2,
* IDX1,IDY1,IDX2,IDY2,IPIE)
CALL REARC2(DXC,IDYC,IDX1,IDY1,IDX2,IDY2,IPIE)
* IDX1,IDY1,IDX2,IDY2,IPIE)
CALL ARC3(IX1,IY1,IX2,IY2,IX3,IY3,ICOLOR,IPIE)
CALL RARC3(IDX1,IDY1,IDX2,IDY2,IDX3,IDY3,IPIE)

```

На языке Си:

```

int x1,y1,x2,y2,r,rx,ry,cw,icolor,pie;
int xc,yc,dxc,dyc,icolor,r;
int dx1,dy1,dxc,dyc,dx2,dy2;
circle (xc,yc,icolor,r);
rcircle (dxc,dyc,r);
circ2 (xc,yc,x1,y1,icolor);
rcirc2 (dxc,dyc,dx1,dy1);
circ3 (x1,y1,x2,y2,x3,y3,icolor);
rcirc3 (dx1,dy1,dx2,dy2,dx3,dy3);
ellipse (xc,yc,icolor,rx,ry);
rellipse (dxc,dyc,rx,ry);
elli2 (xc,yc,icolor,xr1,yr1,xr2,yr2);
relli2 (dxc,dyc,dxr1,dyr1,dxr2,dyr2);
arc (x1,y1,r,cw,xc,yc,icolor,x2,y2,pie);
rarc (dx1,dy1,r,cw,dxc,dyc,dx2,dy2,pie);
earc (x1,y1,rx,ry,cw,xc,yc,icolor,x2,y2,pie);
rearc (dx1,dy1,rx,ry,cw,dxc,dyc,dx2,dy2,pie);
arc2 (xc,yc,icolor,r,cw,dx1,dy1,dx2,dy2,pie);
rarc2 (dxc,dyc,r,cw,dx1,dy1,dx2,dy2,pie);
earc2 (xc,yc,icolor,xr1,yr1,xr2,yr2,
dx1,dy1,dx2,dy2,pie);
rearc2 (dxc,dyc,dxr1,dyr1,dxr2,dyr2,
dx1,dy1,dx2,dy2,pie);
arc3 (x1,y1,x2,y2,x3,y3,icolor,pie);
rarc3 (dx1,dy1,dx2,dy2,dx3,dy3,pie);

```

15. ВЫВОД СИМВОЛОВ И ТЕКСТА, И УСТАНОВКА ИХ ПАРАМЕТРОВ

Графический протокол включает в себя команды вывода отдельных символов и строк текста, выбора фонта, а также установки размера, направления осей символа, ориентации и направления построения символа.

Программа графического протокола может выводить символы по одному из трех наборов прорисовок (фонтов). Во первых, это фонт текстового дисплея, имеющий матрицу 10×10 или 6×10 (в режиме 132 символа на строке). Ему присвоен номер 0. Прорисовки фонта 1 построены в матрице 9×10 и аналогичны по форме символам алфавитно-цифрового дисплея, но без дублирования точек вертикальных линий, то есть вертикальные линии символов имеют толщину 1 пиксел. Символы фонта 2 построены в матрице 16×18, имеют улучшенную форму и могут применяться для рисования символов увеличенного размера. Номер используемого фонта устанавливается командой SETFONT.

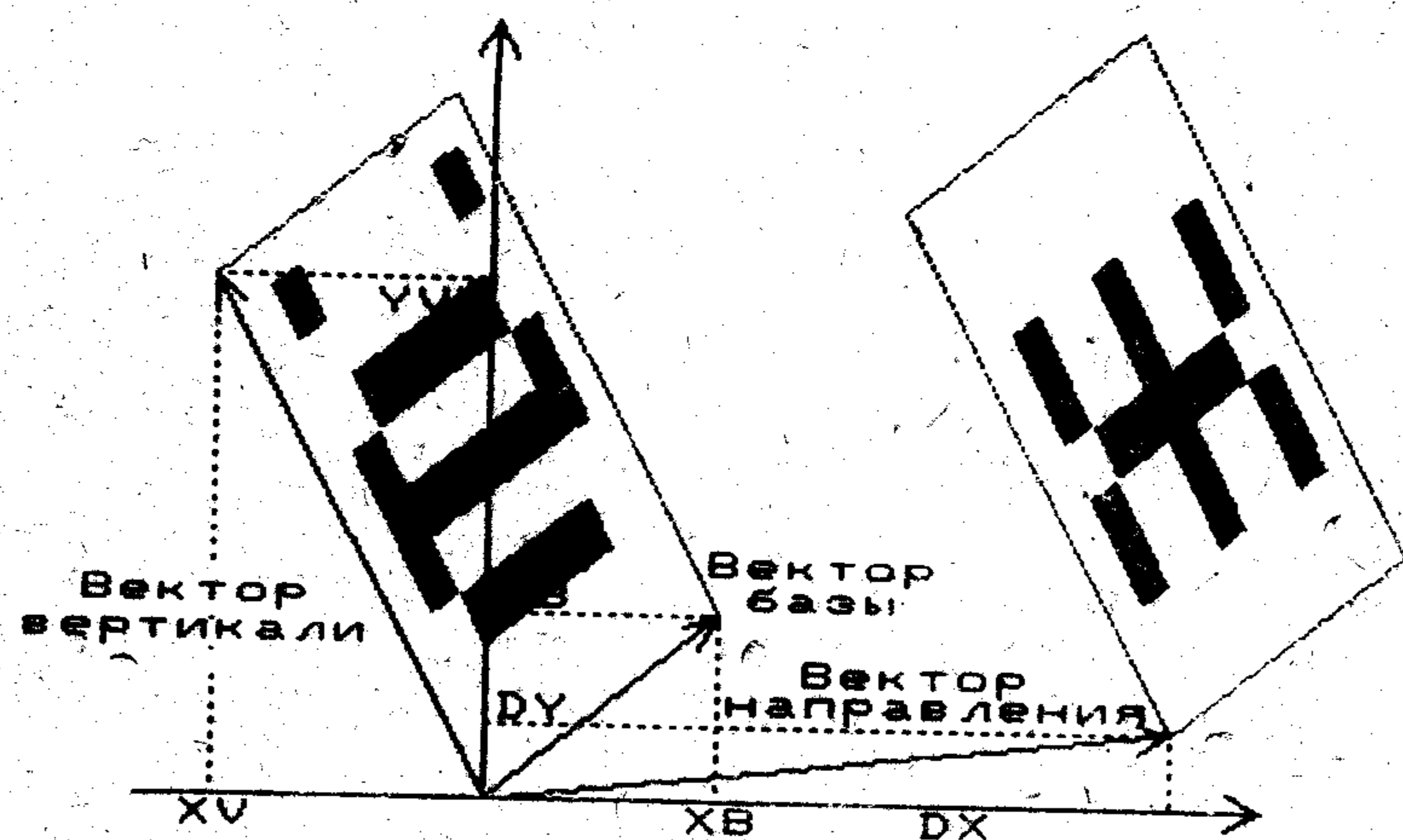
Примечание. Версия программы графического протокола, работающая с управляющей программой в ПЗУ КР1801РЕ2-181, выводит символы фонта 0 и фонта 1 одинаковыми, и с единичной толщиной.

Как и в символьном дисплее, при рисовании символов программа графического протокола учитывает текущий регистр, а точнее, назначение наборов символов G0—G3 и таблиц GR, GL. Эти назначения могут быть изменены управляющими символами SO, SI (PUC, LAT), и специальными ESC-последовательностями, описанными в документации на символьный дисплей. Когда установлен фонт 0, то доступен также набор загруженных прорисовок (Down Reloadable Character Set — DRCS), при использовании соответствующей SCS-последовательности, таким образом, можно рисовать символы, определенные пользователем. При работе в режиме ANSI следует помнить, что посылка любой ESC-последовательности прерывает выполнение DCS-последовательности, поэтому для продолжения последовательности графических команд следует вызвать процедуру DCSOPEN.

При использовании фонтов 1 и 2 загруженные алфавитно-цифровые прорисовки недоступны, однако можно переопределять прорисовку любого символа установленного фонта. Загрузка прорисовки выполняется командой LOADFONT, параметрами которой являются символ, прорисовка которого должна быть заменена, и массив из 18—16-разрядных слов, определяющих саму прорисовку. Каждое слово является строкой матрицы символа, старший бит слова соответствует левому пикселу строки, строки следуют сверху вниз. При загрузке символа для фонта 1 (матрица 9×10) лишняя информация игнорируется. Следует помнить, что для различных символов, имеющих одинаковый вид, например, Р рус-

ская (код 162 набора кириллицы) и P латинская (код 120 набора ASCII), фактически хранится одна прорисовка, поэтому при перезагрузке одной изменяется и другая.

Ориентация, наклон, размеры и пропорции выводимых символов определяются координатами так называемых векторов базы и вертикали символа. При нормальном положении символа (без поворота) вектор базы идет из левого нижнего угла знакоместа символа в правый нижний. Вектор вертикали идет из левого нижнего угла в левый верхний. Если векторы не перпендикулярны, то символ приобретает наклон (курсив), а если их направление не совпадает с осями координат, то символ выводится повернутым. Для построения символа указывается его координата левого нижнего угла. После вывода одного символа точка, с которой будет строиться следующий (текущая позиция), перемещается на так называемый вектор направления текста. Обычно он совпадает с вектором базы, однако может быть задан произвольным образом, и, таким образом, например, можно строить текст с нормально ориентированными символами, идущими по вертикали или по диагонали. На рисунке приведен пример векторов базы, вертикали и направления символа.



Полный набор векторов символа (в мировых координатах) задается командой SETCHVEC. Она имеет параметры XВ, YВ — координаты вектора базы, XV, YV — координаты

вектора вертикали, DX, DY — вектор направления текста. Задаются мировые координаты, поэтому масштаб и пропорции выводимого символа зависят также от текущих границ окна и поля вывода. Значения параметров должны находиться в диапазоне от -256 до $+255$. Для символа нормальной ориентации одинарного размера координаты векторов имеют следующие значения: XВ=10, YВ=0, XV=0, YV=10, DX=10, DY=0.

Обычно требуемые параметры ориентации символа принимают значения из некоторого ограниченного набора, поэтому удобнее применять команду SETCHTYP, которая позволяет задать высоту и ширину символа, направление вектора базы (8 вариантов через 45), ориентацию вектора направления текста (через 45), дополнительный интервал между символами, наклон символа (± 22 , ± 45) от вертикали. Команда SETCHTYP имеет параметры SX — размер символа по базовому вектору, одинарный размер — 10; SY — размер символа по вектору вертикали (одинарный размер — 10). Если SX или SY имеет отрицательное значение, то символ будет зеркально отражен относительно соответствующей оси. CHUP — ориентация вектора базы (от 0 до 7). 0 соответствует ориентации вектора базы по оси X экрана, от 1 до 7 — поворот с шагом 45 против часовой стрелки. CHDIR — направление вывода текста относительно базового вектора: 0 — по базовому вектору (нормальное направление), 1—7 — поворот с шагом 45 против часовой стрелки. SPACE — дополнительный интервал между символами по вектору направления (допускаются положительные и отрицательные значения). Если SPACE=0, то символы будут идти с нормальным шагом, если SPACE=SX, то между символами будет интервал, равный размеру символа. Последний аргумент SKEW задает наклон символа (угол между векторами базы и вертикали). 0 — выводятся нормальные символы, 1 — наклон вправо на 22, 2 — наклон вправо на 45, 7 — наклон влево на 22, 6 — наклон влево на 45. При других значениях символы выводятся без наклона, как при SKEW=0. Программа графического протокола пересчитывает принятые значения в мировые координаты векторов базы, вертикали и направления символов и запоминает их. Значения SX и SY должны находиться в диапазоне от -256 до $+255$.

Пример построения символов с различной ориентацией, наклоном и направлением приведен на рисунках.

Вывод символов в различных направлениях:

Направление 2
 Направление 1
 Направление 4
 Направление 3

Направление 2
 Направление 1
 Направление 4
 Направление 3

Вывод символов различного размера:

Р Р Р Р Р
 Р Р Р Р Р
 Р Р Р Р Р
 Р Р Р Р Р
 Р Р Р Р Р

Вывод символов с различным просветом, направлением вывода текста и наклоном:

Д Д Д Д Д Д Д Д Д Д

В В В В В В В В В В

К К К К К К К К К К

Ниже приведен пример построения символов различных фонтов:

Фонт 0

ABCDEFabcdefАБВГДЕабвгде012345+!"#\$%&: <

Фонт 1

ABCDEFabcdefАБВГДЕабвгде012345+!"#\$%&: <

Фонт 2

ABCDEFabcdefАБВГДЕабвгде

При построении символов мировые координаты векторов базы, вертикали и направления, заданные командой SETCHVEC или вычисленные по команде SETCHTYP, пересчитываются в экранные координаты согласно текущим размерам окна и поля вывода. Если вычисленный размер знака места для вывода символа меньше 10 пикселей по направле-

нию базы или вертикали, то выводится лишь часть символа, как показано на рисунке:

ОТСЕЧЕНИЕ СИМВОЛОВ

ОТСЕЧЕНИЕ СИМВОЛ

Ячейки достаточных размеров

Ячейки недостаточных размеров

Команда CHARACTER выводит один символ в заданную позицию. Команда RCHAR строит символ, координаты которого заданы смещением от текущей позиции. Команда TCHAR выводит символ в текущую позицию. После построения символа текущая позиция перемещается по вектору направления текста. Таким образом, при последовательном выполнении команд TCHAR выводится строка текста.

Управляющие символы РУС, ЛАТ не должны являться аргументами команд CHARACTER, RCHAR, TCHAR, так как они обрабатываются программой текстового дисплея, не доходя до программы графического протокола. Эти символы следует посылать обычными операторами языка, такими, как WRITE в Паскале, TTYOUR в Фортране. Они не прерывают DCS-последовательность.

Команда TEXT выводит сразу целую строку текста по установленному вектору направления текста. В команде RTEXT координаты точки, с которой начинается построение строки текста, задаются смещением от текущей позиции. В составе графической команды передается параметр NCHAR, указывающий число передаваемых печатных символов, за ним следуют символы выводимого текста. Среди них могут присутствовать символы переключения регистров РУС (код 16) и ЛАТ (код 17), они переключают алфавит таким же образом, как и при работе алфавитно-цифрового дисплея, их число не входит в NCHAR. Аргументом процедур TEXT и RTEXT является строка выводимого текста, ее длина задается аргументом N. Если в строке нет байта с кодом 0 (используемого как индикатор конца строки в Фортране и Си), то выводится N символов, а если есть байт 0, то выво-

дятся символы до него. Таким образом, в программах на Фортране и Си в качестве аргумента N можно указывать константу, которая заведомо больше длины выводимой строки. Максимально в строке может быть передано 95 печатных символов.

Формат команд:

```
SETFONT :  
c (143) NFONT1  
LOADFONT :  
d (144) CHAR WORD1; WORD2; ... WORD18  
SETCHVEC :  
Q (121) (XB,YB)3 (XV,YV)3 (DX,DY)3  
SETCHTYP :  
P (120) (SX,SY)3 CHUP1 CHDIR1 SPACE3 SKEW1  
TCHAR :  
9 (71) CHAR  
CHARACTER :  
: (72) (X3,Y3) + ICOLOR4 CHAR  
RCHAR :  
; (73) (DX,DY)3 CHAR  
TEXT :  
< (74) (X3,Y3) + ICOLOR4 NCHAR1 CHAR1 CHAR2 ... CHARN  
RTEXT :  
= (75) (DX,DY)3 NCHAR1 CHAR1 CHAR2 ... CHARN
```

Для вывода одиночных символов и строк текста в программах на языке Паскаль необходимы следующие объявления:

```
TYPE LINTXT=ARRAY[1..N] OF CHAR;  
TYPE CELL:ARRAY[1..18] OF INTEGER;  
PROCEDURE SETFONT(NFONT:INTEGER); EXTERNAL;  
PROCEDURE LOADFONT(CHR:CHAR;FONT:CELL); EXTERNAL;  
PROCEDURE  
  SETCHVEC(XB,YB,XV,YV,DX,DY:INTEGER);  
EXTERNAL;  
PROCEDURE  
  SETCHTYP(SX,SY,CHUP,CHDIR,SPACE,SKEW:INTEGER);  
EXTERNAL;  
PROCEDURE CHARACTER (X,Y,ICOLOR:INTEGER;CHR:CHAR);  
  EXTERNAL;  
PROCEDURE RCHAR(DX,DY:INTEGER;CHR:CHAR); EXTERNAL;  
PROCEDURE TCHAR(CHR:CHAR); EXTERNAL;  
PROCEDURE  
  TEXT(X,Y,ICOLOR,N:INTEGER;VAR TXT:LINTXT);  
EXTERNAL;  
PROCEDURE  
  RTEXT(DX,DY,N:INTEGER;VAR TXT:LINTXT);  
EXTERNAL;
```

Из программы на языке Паскаль процедуры вызываются операторами:

```
SETFONT(NFONT);  
LOADFONT(CHR, FONT);  
SETCHVEC (XB, YB, XV, YV, DX, DY);  
SETCHTYP (SX, SY, CHUP, CHDIR, SPACE, SKEW);  
CHARACTER(X, Y, ICOLOR, CHR);  
RCHAR(DX, DY, CHR);  
TCHAR(CHR);  
TEXT(X, Y, ICOLOR, N, TXT);  
RTEXT(DX, DY, N, TXT);
```

На языке Фортран:

```
INTEGER FONT(18)  
CALL SETFON(NFONT)  
CALL LOADFO(ICHAR, FONT)  
CALL SETCHV(IXB, IYB, IXV, IYV, IDX, IDY)  
CALL SETCHT(ISX, ISY, ICHUP, ICHDIR, ISPC, ISKEW)  
CALL CHARAC(IX, IY, ICOLOR, ICHAR)  
CALL RCHAR(IDX, IDY, ICHAR)  
CALL TCHAR(ICHAR)  
BYTE TXT(N) ! вместо N должно быть число  
CALL TEXT(IX, IY, ICOLOR, N, TXT)  
CALL RTEXT(IDX, IDY, N, TXT)
```

На языке Си:

```
int font[18];  
char nfont, chr;  
int x, y, dx, dy, icolor;  
char chr, *txt;  
int xb, yb, xv, yv, dx, dy, sx, sy, chup, chdir, space, skew;  
setfont(nfont);  
loadfont(chr, font);  
setchevec (xb, yb, xv, yv, dx, dy);  
setchtyp (sx, sy, chup, chdir, space, skew);  
character(x, y, icolor, chr);  
rchar(dx, dy, chr);  
tchar(chr);  
text(x, y, icolor, n, txt);  
rtext(dx, dy, n, txt);
```

16. РИСОВАНИЕ ГРАФИЧЕСКОГО КУРСОРА И ЛОКАТОРА, ЗАПРОС КООРДИНАТ КУРСОРА И ЛОКАТОРА

Графический курсор — это мигающий (синхронно с текстовым курсором) символ специального вида, выводимый в текущей позиции. Предусмотрено 12 вариантов (типов) кур-

сора, из них 11 аналогичны типам маркеров, кроме того, имеется курсор в виде пересекающихся горизонтальной и вертикальной линий, простирающихся до границ поля вывода. Курсор может иметь различный размер (множитель размера от 1 до 8) и цвет. Курсор выводится с режимом вывода инверсией, благодаря этому при его стирании на экране не остается следов. Курсор, так же как и маркеры, строится в экранных координатах, то есть его размер и пропорции не зависят от соотношения размеров окна и области вывода.

Графический локатор — это специальный подвижный курсор, положение которого может изменяться с помощью клавиш управления курсором или манипулятора «мышь». Графический локатор может выводиться, во-первых, подобным одному из типов маркеров, во-вторых, в виде пересекающихся горизонтальной и вертикальной линий, простирающихся до границ поля вывода, в виде отрезка прямой от текущей позиции до позиции графического локатора («резинная линия»), в виде прямоугольника, одним углом которого является текущая позиция, другим углом — позиция локатора («резинный прямоугольник»), и, в-третьих, в виде произвольно задаваемого рисунка.

При построении курсор и локатор отсекаются текущими границами установленного поля вывода.

Вид и размер графического курсора, а также цвет, которым он будет выводиться на экран, устанавливаются командой SETGCURSOR, имеющей аргументы STYPE — тип, CSIZE — размер (от 1 до 8), CCOLOR — цвет курсора. Для вывода курсора в виде двух пересекающихся линий параметр STYPE должен иметь значение 0. Курсор по умолчанию имеет вид ромба, перечеркнутого диагоналями (тип — 6), двойного размера.

Индикация графического курсора включается и выключается командой GCURSOR.

Вид и размер графического локатора, а также его индекс цвета, устанавливаются командой SETLOCATOR с аргументами LTYPE, LSIZE, LCOLOR — тип, размер и цвет. Локатор по умолчанию имеет вид пересекающихся линий. Индикация локатора включается и выключается командой LOCATOR. Начальную позицию локатора можно задать с помощью команды PUTLOCATOR, а если такой команды не было, то локатор будет находиться в левом нижнем углу экрана.

Для вывода локатора, имеющего вид «резинной нити», на конце которой выводится маркер, в команде SETLOCATOR

необходимо указывать код типа конечного маркера, к которому прибавлено 16дес (20восьм). Если указать код типа 17дес, (точечный маркер), то будет выводиться просто линия. Для вывода локатора в виде «резинового прямоугольника» с маркером в углу, в команде SETLOCATOR необходимо указывать код типа маркера, к которому прибавлено 32дес (40 восьм).

С помощью локатора можно перемещать по экрану произвольный рисунок. Рисунок задается с помощью команды DEFLOCATOR, команда с аргументом 1 начинает запоминание последующих графических команд, а команда с аргументом 0 оканчивает запоминание. Графические команды, принимаемые в КЦГД между командами DEFLOCATOR(1) и DEFLOCATOR(0), запоминаются в буфере подвижного рисунка, размер которого равен 256 байт. Следует отметить, что запоминаются не все команды, а лишь выполняющие непосредственно рисование или устанавливающие параметры линий (LINESTYLE, USERSTYLE). Во время вывода такого рисунка устанавливается режим записи —1 (исключающее ИЛИ), цвет — установленный командой SETLOCATOR, заповнение выключается.

Во время определения подвижного рисунка необходимо установить нулевой текущий цвет и временный цвет в примитивах, а режим вывода — любой, кроме занесения, для того, чтобы задание подвижного рисунка не нарушало имеющееся изображение.

Для вывода подвижного рисунка подается команда SETLOCATOR, в которой в качестве аргумента LTYPE указана номер одного из маркеров (—7..6), к которому прибавлено число 48дес (60восьм).

Не следует задавать слишком сложные рисунки, так как при этом будет затрачиваться слишком большое время на его отрисовку и стирание.

Локатор перемещается с помощью клавиш управления курсором, если они нажимаются при нажатой клавише ВР или СУ. Если нажата клавиша СУ, то за одно нажатие клавиши локатор перемещается в соответствующем направлении на 10 пикселей, а если нажата клавиша ВР, то курсор перемещается на один пиксел. При этом в ЭВМ дисплей никаких кодов не посылает. Когда клавиши СУ и ВР не нажаты, то при нажатии клавиш-стрелок в ЭВМ посылаются их обычные коды, описанные в документации на алфавитно-цифровой дисплей. Прочие клавиши (кроме 0 и ВВОД) функционируют независимо от того, включен ли локатор или нет.

Если локатор был включен командой LOCATOR с аргументом 1, то при выходе его за одну из границ текущего поля вывода он переходит на противоположную границу, то есть движется как бы по кольцу. Если же локатор был включен командой с аргументом 2, то такого перехода не происходит, и при выходе за границы поля вывода локатор становится невидимым.

Если вывод локатора был выключен командой LOCATOR, а затем снова включен, то он выводится в прежней позиции.

Для управления локатором можно использовать также устройство типа «мышь», если оно подключено к соответствующему разъему КЦГД. Скорость передвижения локатора может быть замедлена в 8 раз путем нажатия клавиши СУ или ВР, что позволяет осуществить более точное позиционирование. Когда локатор выключен, то при перемещении «мыши» дисплей посылает в ЭВМ последовательности, аналогичные посылаемым при нажатии клавиш управления курсором, то есть «мышь» можно использовать и для управления алфавитно-цифровым курсором.

При нажатии и отпускании клавиш «мыши» в ЭВМ посылаются запрограммированные последовательности символов (до 6 символов). Программирование этих последовательностей производится командой DCS-последовательностью следующего вида:

```
ESC P Pc ; Pv $ w N / Spush / Srelease ; ESC \
```

(коды 33 120 Pc 73 Pv 44 167

N 57 Spush 57 Srelease 73 33 134)

Pc — это параметр режима очистки. Если он равен 0 или опущен, то перед программированием старые значения клавиш очищаются, иначе — изменяются только указанные клавиши.

Pv — это параметр, указывающий скорость перемещения локатора. Если он опущен или равен 0, то сохраняется старое значение. По умолчанию он имеет значение 17. При больших значениях скорость передвижения увеличивается, при меньших — уменьшается в два раза на каждую единицу Pv.

Значения параметров Pc и Pv задаются десятичными числами, аналогично CSI-последовательности алфавитно-цифрового режима работы дисплея.

N — это номер программируемой клавиши. Он равен 1 (61восьм) для левой клавиши и 2 (62восьм) для правой.

Spush и Srelease — это закодированные в шестнадцатичном виде последовательности, которые должны посылать-

ся в ЭВМ соответственно при нажатии и отпускании клавиши. Каждый посылаемый байт кодируется двумя шестнадцатичными цифрами (символы 0—9 с кодами 60—71, А—F с кодами 101—105 и а—f с кодами 141—145). Всего в последовательности не может быть больше 6 байт. Если вводится больше 6 байт, то лишние игнорируются.

Клавиша 0 на дополнительной клавиатуре с нажатой клавишей СУ или ВР эквивалентна левой кнопке, а клавиша ВВОД — правой. При нажатии на эти клавиши совместно с СУ в машину передается код нажатия, а совместно с ВР — код отжатия клавиши, запрограммированные описанной выше DCS-последовательностью.

Ввод координат локатора в ЭВМ производится командами GETLOCATOR и GETWLOCATOR, по первой команде в ЭВМ выдаются экранные координаты, а по второй команде выдаются мировые координаты. Команды GETCURRENT и GETWCURRENT посылают в ЭВМ соответственно экранные и мировые координаты текущей позиции. Программа графического протокола, приняв одну из перечисленных команд, посылает пару координат X, Y, каждая закодирована в виде трех символов, подобно тому, как передаются координаты из ЭВМ в КЦГД. Кроме значений координат, в дополнительных битах трехсимвольного представления чисел передается код, указывающий расположение текущей позиции относительно границ поля вывода. Если эта позиция находится в поле вывода, то передаваемый код равен нулю, в противном случае передается сумма следующих величин:

- 1 — X < VXLEFT
- 2 — X > VXRIGHT
- 4 — Y < VYBOTT
- 8 — Y > VYTOP

При выдаче координат локатора в дополнительных битах передаются признаки нажатия кнопок «мыши»:

- 0 — кнопки не нажаты;
- 1 — нажата левая кнопка;
- 2 — нажата правая кнопка;
- 3 — нажаты обе кнопки.

Этот код возвращается, как значение функции. Если локатор находится за пределами поля вывода, то к возвращаемому значению будет прибавлено 4.

Процедуры GETLOCATOR и GETWLOCATOR, имеющиеся в интерфейсном пакете, очищают системный буфер ввода, посылают команду и сразу же принимают координаты. Эти

процедуры реализуют простейший способ работы с локатором.

Некоторые интерактивные прикладные программы требуют такой режим работы, когда одновременно выполняются и работа с клавиатурой и с локатором. В этом случае в буфере ввода могут находиться и символы, посылаемые клавиатуры, и последовательности, в которых закодированы координаты локатора, и прикладной программе необходимо их различать. Программа графического протокола позволяет запрограммировать «префикс» и «постфикс», которые дисплей будет добавлять к координатам, посылаемым по командам GETLOCATOR, GETWLOCATOR, GETCURRENT, GETWCURRENT. «Префикс» и «постфикс» программируются таким же образом, как и последовательности, передаваемые при нажатии клавиш «мыши», но в качестве параметра N (номер кнопки) указывается 9 (код 71). В интерфейсном пакете, кроме того, предусмотрены процедуры, которые позволяют послать в КЦГД команду и отдельно принять ответные координаты. Прикладная программа посылает команду запроса координат, после чего продолжает обычный прием информации с клавиатуры. Для посылки команд в интерфейсных пакетах имеются процедуры REQLOC, REQWLO, REQCUR, REQWCU, не имеющие аргументов. Когда программа опознает префикс координат, то принимает эти координаты, используя процедуру GETCOORD. Посылка координат никогда не разрывается информацией с клавиатуры.

В интерфейсных пакетах имеется функция приема одного слова, переданного дисплеем. Это может быть, например, значение координаты. Функция называется GPRTIW, она присваивает аргументу WORD значение принятого слова, а в качестве значения возвращает два бита дополнительной информации, которые дисплей посылает вместе со словом. Ввод производится через системный запрос .TTYIN, в отличие от функции GPGETW, которая будет описана в следующем разделе.

Прикладная программа может использовать локатор следующим образом. Сначала командой SETLOCATOR устанавливается тип локатора, затем командой LOCATOR включается его отображение. Пользователь, с помощью «мыши», а если ее нет, то клавишами-стрелками, устанавливает локатор в нужную ему позицию и нажимает либо некоторую клавишу основной клавиатуры, либо клавишу «мыши». После этого командой GETLOCATOR или GETWLOCATOR необходимо считать координаты локатора. Для выбора некоторой

области, например, для печати ее копии, сначала выбирается один угол, считываются мировые координаты локатора и командой POSITION на эту точку устанавливается текущая позиция. Затем включается локатор типа «резиновый прямоугольник» (тип 33) и выбирается второй угол области.

Форматы команд:

SETG CUR :
\
(134) CTYPE₁ (CSIZE-1)₁ CCOLOR₁

GCURSOR :
Z (132) ONOFF₁

SETLOCATOR :
]
(135) LTYPE₁ (LSIZE-1)₁ LCOLOR₁

LOCATOR :
— (137) ONOFF₁

PUTLOCATOR :
^ (136) X₃ Y₃
X, Y — мировые координаты

DEFLOCATOR :
g (147) ONOFF₁

REQLOCATOR, GETLOCATOR :
H (110)

REQWLOCATOR, GETWLOCATOR :
I (111)

REQCURRENT, GETCURRENT :
J (112)

REQWCURRENT, GETWCURRENT :
K (113)

Для работы с графическим курсором и локатором в программах на языке Паскаль необходимы следующие объявления:

```
PROCEDURE SETG CURSOR(CTYPE,CSIZE,CCOLOR:INTEGER);  
EXTERNAL;  
PROCEDURE GCURSOR(ONOFF:INTEGER); EXTERNAL;  
PROCEDURE SETLOCATOR(LTYPE,LSIZE,LCOLOR:INTEGER);  
EXTERNAL;  
PROCEDURE LOCATOR(ONOFF:INTEGER); EXTERNAL;  
PROCEDURE PUTLOCATOR(X,Y:INTEGER); EXTERNAL;  
PROCEDURE DEFLOCATOR(ONOFF:INTEGER); EXTERNAL;  
PROCEDURE REQLOCATOR; EXTERNAL;  
PROCEDURE REQWLOCATOR; EXTERNAL;  
PROCEDURE REQCURRENT; EXTERNAL;  
PROCEDURE REQWCURRENT; EXTERNAL;
```

```
FUNCTION GETLOCATOR(VAR X,Y:INTEGER):INTEGER; EXTERNAL;  
FUNCTION GETWLOCATOR(VAR X,Y:INTEGER):INTEGER; EXTERNAL;  
FUNCTION GETCURRENT(VAR X,Y:INTEGER):INTEGER; EXTERNAL;  
FUNCTION GETWCURRENT(VAR X,Y:INTEGER):INTEGER; EXTERNAL;  
FUNCTION GETCOORD(VAR X,Y:INTEGER):INTEGER; EXTERNAL;  
FUNCTION GPTTIW(VAR WORD:INTEGER):INTEGER; EXTERNAL;
```

Из программы на языке Паскаль процедуры вызываются операторами:

```
SETG CURSOR(CTYPE,CSIZE,CCOLOR) ;  
GCURSOR(ONOFF);  
SETLOCATOR(LTYPE,LSIZE,LCOLOR);  
DEFLOCATOR(ONOFF);  
REQLOCATOR;  
REQWLOCATOR;  
REQCURRENT ;  
REQWCURRENT;  
I:=GETLOCATOR(X,Y);  
I:=GETWLOCATOR(X,Y);  
I:=GETCURRENT(X,Y);  
I:=GETWCURRENT(X,Y);  
I:=GETCOORD(X,Y);  
I:=GPTTIW(WORD);
```

На языке Фортран:

```
CALL SETGCU(CTYPE,CSIZE,CCOLR)  
CALL GCURSO(IONOFF)  
CALL SETLOC(LTYPE,LSIZE,LCOLOR)  
CALL LOCATO(IONOFF)  
CALL PUTLOC(IX,IY)  
CALL DEFLOC(IONOFF)  
CALL REQLOC  
CALL REQWLO  
CALL REQCUR  
CALL REQWCU
```

Процедуры могут вызываться и как функции, и как подпрограммы:

```
INTEGER GETLOC,GETWLO,GETCUR,GETWCU,GETCOO,GPTTIW  
I=GETLOC(IX,IY)  
I=GETWLO(IX,IY)  
I=GETCUR(IX,IY)  
I=GETWCU(IX,IY)  
I=GETCOO(IX,IY)  
I=GPTTIW(IWORD)  
CALL GETLOC(IX,IY)  
CALL GETWLO(IX,IY)  
CALL GETCUR(IX,IY)
```

CALL GETWCU(IX,IY)
 CALL GETCOO(IX,IY)
 CALL GPTTIW(IWORD)

На языке Си:

```
int ctype, csize, ccolor, onoff;
  setgcursor(ctype, csize, ccolor);
  gcursor(onoff);
  setlocator(ltype, lsize, lcolor);
  locator(onoff);
  putlocator(x, y);
  deflocator(onoff);
  reqlocator();
  reqwlocator();
  reqcurrent();
  reqwcurrent();
int getlocator, getwlocator, getcurrent, getwcurrent,
  getcoord, gpttiw;
int x, y, icode, iword;
  icode = getlocator(&x, &y);
  icode = getwlocator(&x, &y);
  icode = getcurrent(&x, &y);
  icode = getwcurrent(&x, &y);
  icode = getcoord(&x, &y);
  icode = gpttiw(&iword);
/* возвращаемое значение можно не использовать : */
getlocator(&x, &y);
getwlocator(&x, &y);
getcurrent(&x, &y);
getwcurrent(&x, &y);
getcoord(&x, &y);
gpttiw(&iword);
```

**17. ЗАГРУЗКА ИЗОБРАЖЕНИЙ,
 ЧТЕНИЕ ИЗОБРАЖЕНИЙ
 И КОДА ОДНОГО ПИКСЕЛА**

Команда графического протокола LOADPICTURE загружает в заданную область экрана некоторое изображение. Кроме того, загружается аппаратная таблица цветов. Параметрами команды являются 16 байтов таблицы цветов, X, Y — экранные координаты начальной точки загружаемой области, NX — ширина области, деленная на 4 (число слов), NY — число строк в области, DEN800 — признак плотности пикселей загружаемого изображения. Если DEN800=1, то загружается изображение с пикселями высокой плотности, а если DEN800=0, то загружается изображение с пикселями низкой плотности. В видеопамати в одном слове хранится 4 пикселя низкой плотности или 8 пикселей высокой плотности. Параметр NX задает именно число записываемых

слов на одну строку, а всего передается NX*NY слов. Параметр TB указывает направление загрузки строк. При TB=0 изображение загружается снизу вверх, при этом X, Y задают координаты левой нижней точки области, а при TB>0 изображение загружается сверху вниз, при этом X, Y задают координаты левой верхней точки области.

После подачи команды необходимо передать в дисплей загружаемую информацию. Загружаемые слова передаются по два байта — сначала младший, затем старший. Старшие биты слова соответствуют самому левому пикселу, младшие биты — правому. Для передачи загружаемой информации в дисплей в интерфейсных пакетах имеется процедура PUTPICTURE, аргументами которой является буфер и число передаваемых слов. Для полной загрузки изображения процедуру PUTPICTURE можно вызывать несколько раз (например, если информация читается из файла по одному блоку). Всего должно быть послано NX*NY слов.

Выгрузка изображения из КЦГД выполняется по команде графического протокола DUMMPICTURE, параметры X, Y которой задают экранные координаты точки, от которой начинается выгрузка, NX задает число выгружаемых слов по строке, а NY — число строк. Если NY>0, то выгрузка идет снизу вверх, а если NY<0, то выгрузка идет сверху вниз. Параметр DIR задает вид выгрузки. DIR=0 — изображение передается по столбцам слева направо, DIR=1 — изображение передается по строкам слева направо.

Информация из КЦГД передается в виде слов, закодированных в виде трех байтов, так же, как передаются координаты в КЦГД. В двух дополнительных битах такой посылки передается признак плотности пикселей в данном слове: 0 — низкая плотность (4 пикселя в слове), 1 — высокая плотность (8 пикселей в слове). Для приема этой информации в интерфейсных пакетах имеется процедура GETPICTURE, которой в качестве аргумента RECEIVER передается имя процедуры, написанной на языке программирования высокого уровня, которой будут переданы раскодированные слова, принятые из КЦГД. В этой процедуре принятая информация, например, может накапливаться и по мере заполнения буфера записываться на диск. Передаваемая процедура должна иметь следующий заголовок:

```
PROCEDURE RCV(WORD, DENSITY: INTEGER);
BEGIN
.....
END;
```


При написании процедуры RCV на языке Паскаль следует учесть, что в ней разрешен доступ только к глобальным переменным программы. Доступ к локальным переменным процедур не разрешен.

На время работы процедур LOADPICTURE/PUTPICTURE и DUMPPICTURE/GETPICTURE запрещаются прерывания от клавиатуры. По окончании работы процедур прерывания от клавиатуры разрешаются. Это сделано для того, чтобы информация, идущая из КЦГД, не попадала в системный буфер ввода. Информация передается настолько быстро, что в противном случае она переполнит системный буфер, и часть ее пропадет.

Программа может принять информацию из КЦГД, и не обращаясь к процедуре GETPICTURE. Для этого она должна сразу же после обращения к DUMPPICTURE запретить прерывания от клавиатуры командой:

```
BIS #AO 100, @ #AO 177560
```

В программе на Паскале такую команду можно поместить в виде ассемблерной вставки. В программе на Фортране можно поставить следующий оператор:

```
CALL POKE("177560, IPEEK("177560).AND."177677)
```

В программе на языке Си это выполняется следующим оператором:

```
*(int*)0177560 &= 0177677;
```

Далее программа получает слова, обращаясь к процедуре GPGETW, имеющейся в интерфейсном пакете. Процедура возвращает принятое из КЦГД слово и его признак разрешения, причем информация от дисплея принимается непосредственно из регистра данных, в отличие от описанной выше процедуры GPTTIW. После того, как все слова приняты, необходимо вновь разрешить прерывания от клавиатуры. В программе на Паскале это делается следующей ассемблерной вставкой:

```
{ $C BIS #AO 100, @ #AO 177560  
}
```

В программе на Фортране можно поставить следующий оператор:

```
CALL POKE("177560, IPEEK("177560).OR."100)
```

В программе на языке Си это выполняется следующим оператором:

```
*(int*)0177560 |= 0100;
```

Команда получения кода пиксела называется GETPIXEL. Аргументами ее являются мировые координаты точки. Функция GETPIXEL, входящая в состав интерфейсных пакетов, возвращает код пиксела в диапазоне 0..15. Если указанная точка находится вне пределов поля вывода, то возвращается значение 16.

Формат команд:

```
LOADPICTURE :  
W (127) COL01 COL11... COL151 X3 Y3  
                  (NX3, NY3) + DEN8004
```

```
DUMPPICTURE :  
X (130) X3 Y3 NX3 NY3 + DIR2
```

```
GETPIXEL :  
Y (131) X3 Y3
```

Для загрузки и выгрузки информации в тексте программ на Паскале должны быть следующие объявления:

```
TYPE COL=ARRAY[1..8] OF INTEGER;  
      BUF=ARRAY[1..256] OF INTEGER;  
PROCEDURE  
  LOADPICTURE(COLORS:COL;X,Y,NX,NY,DEN800:INTEGER);  
EXTERNAL;  
PROCEDURE PUTPICTURE(BUFFER:BUF;N:INTEGER); EXTERNAL  
PROCEDURE DUMPPICTURE(X,Y,NX,NY,DIR:INTEGER); EXTERNAL;  
PROCEDURE  
  GETPICTURE(N:INTEGER;PROCEDURE RECEIVER);  
EXTERNAL;  
FUNCTION GPGETW(VAR WORD:INTEGER):INTEGER; EXTERNAL;  
FUNCTION GETPIXEL(X,Y:INTEGER):INTEGER; EXTERNAL;
```

В программах на языке Паскаль для выполнения команд LOADPICTURE и DUMPPICTURE необходимы следующие операторы:

```
LOADPICTURE(COLORS,X,Y,NX,NY,DEN800);  
PUTPICTURE(BUFFER,N);  
DUMPPICTURE(X,Y,NX,NY,DIR);  
GETPICTURE(NX*NY,RCV);  
{ Если изображение пишется в файл, то далее  
  следует записать последний блок и закрыть файл }  
DENSITY:=GPGETW(WORD);  
PIXEL:=GETPIXEL(X,Y);
```

На языке Фортран:

```
EXTERNAL RCV  
BYTE COLORS(16)  
INTEGER BUFFER(256),GPGETW  
CALL LOADPI(COLORS,IX,IY,NX,NY,ID800)  
CALL PUTPIC(BUFFER,N)
```

```
CALL DUMPPPI (IX,IY,NX,NY,NDIR)
CALL GETPIC (NX*NY,RCV)
```

```
.....
IDENS=GPGETW (IWORD)
SUBROUTINE RCV (IWORD,IDIR)
```

```
.....
RETURN
END
INTEGER GETPIX
IPIX=GETPIX (IX,IY)
```

На языке Си:

```
rcv (word,dir)
int word,dir;
{ /* тело процедуры */
.....
int x,y,nx,ny,dir,den800,buffer[256],word,density,gpgetw;
char colors[16];
loadpicture (colors,x,y,nx,ny,den800);
putpicture (buffer,n);
dumppicture (x,y,nx,ny,dir);
getpicture (nx*ny,rcv);
density = gpgetw (&word);
int x,y,pixel,getpixel;
pixel = getpixel (x,y);
```

18. ВЫВОД ТВЕРДОЙ КОПИИ

Загружаемая программа графического протокола обрабатывает несколько дополнительных ESC-последовательностей, которые обеспечивают вывод на печатающее устройство «твердой копии» экрана. Эта копия может быть получена по инициативе оператора или по запросу из прикладной программы.

Прикладная программа посылает в КЦГД команду **HARDCOPY**, где задаются экранные координаты границ области, которая должна быть напечатана, а также требуемое смещение начала распечатки относительно левого края листа и способ печати: вдоль или поперек печатного листа. Получив эту команду, программа графического протокола посылает символ с кодом 200, который служит сигналом для специального драйвера получения «твердой копии». Если драйвер не загружен, то монитор операционной системы игнорирует этот символ. Драйвер, получив этот код, должен послать в КЦГД команду **GETHARDCOPY**, по которой ему передаются параметры, необходимые для печати: таблица цветов

и те координаты области, которые установила прикладная программа командой **HARDCOPY**. Получив эту информацию, драйвер запрашивает передачу информации из видео-ОЗУ, посылая одну или несколько команд **DUMPPICTURE**. Полученная графическая информация выводится драйвером на печатающее устройство.

Передача графической информации на печать может быть начата и по запросу оператора. Для этого он должен нажать клавишу **ПЕЧАТЬ КАДРА** совместно с клавишей **СУ** или **ВР**. При этом в ЭВМ передается код 200, и если имеется загруженный драйвер печати копии, то он будет работать в описанной выше последовательности. По запросу оператора всегда печатается полный экран. Режим печати зависит от того, какая клавиша была нажата совместно с клавишей **ПЕЧАТЬ КАДРА**. Если это была **СУ**, то копия будет печататься поперек листа в четырех градациях серого, а если была нажата **ВР**, то печать вдоль листа в 16 градациях.

При нажатии клавиши **ПЕЧАТЬ КАДРА**, без клавиш **ВР** или **СУ**, управляющая программа устанавливает признак прекращения распечатки, по получению которого драйвер должен прервать вывод. Этот признак передается в дополнительном бите графической информации.

ESC-последовательность инициирования печати копии имеет следующий формат:

```
HARDCOPY :
ESC 1 (61) (X03, Y03) + DIR4 X13 Y13 OFFSET3
```

Параметры команды задают экранные координаты границ печатаемой области: **X0** — левая граница, **X1** — правая граница, **Y0** — верхняя граница, **Y1** — нижняя граница. Параметр **DIR** определяет вид печати: 0 — печать с осью **X** экрана поперек печатаемого листа, **DIR=1** — ось **X** вдоль листа. **OFFSET** — смещение распечатки от левой границы листа.

Для использования этой команды в программе на языке Паскаль должны быть следующие объявления:

```
PROCEDURE HARDCOPY (X0,Y0,X1,Y1,DIR,OFFSET); EXTERNAL;
```

Из программы на языке Паскаль процедура вызывается следующим оператором:

```
HARDCOPY (X0,Y0,X1,Y1,DIR,OFFSET);
```

На языке Фортран:

```
CALL HARDCO (IX0, IY0, IX1, IY1, IDIR, IOFF)
```


На языке Си:

```
int x0,y0,x1,y1,dir,offset;  
hardcopy(x0,y0,x1,y1,dir,offset);
```

Команда получения параметров твердой копии имеет формат:

```
GETHARDCOPY :  
ESC 2 (33 62)
```

В ответ программа графического протокола передает следующую информацию:

```
C01 C11 C21 ... C151  
(X03, Y03) + DIR4 X13 Y13 OFFSET3
```

Здесь C0..C15 — байты со значениями цветов из аппаратной таблицы цветности. Они формируются следующим образом: к значениям цветов из диапазона 0..77 прибавляется число 40восьм, полученные коды передаются в виде печатных символов. X0, Y0, X1, Y1 — экранные координаты границ печатаемой области, установленные командой HARDCOPY. DIR — режим печати, OFFSET — смещение распечатки от левого края листа в условных единицах, интерпретируемых конкретным драйвером получения «твердой копии».

19. ПОДПРОГРАММЫ УСТАНОВКИ РЕЖИМОВ ДИСПЛЕЯ

В интерфейсных пакетах имеются процедуры, которые не посылают команды графического протокола, а устанавливают режимы работы дисплея посредством послышки на терминал ESC-последовательностей.

Процедура MODE52, не имеющая аргументов, переводит дисплей в режим совместимости с VT52 и КСМ ДВК-3. Действие ее заключается в послышке в дисплей последовательности, устанавливающей в дисплее режим совместимости. Кроме того, в интерфейсном пакете устанавливается внутренний признак, указывающий на режим передачи команд графического протокола через последовательность ESC M.

Для работы в режиме ANSI с послышкой команд графического протокола в составе DCS-последовательностей, в интерфейсный пакет включена процедура MODE100. Она не имеет аргументов и при вызове посылает в дисплей команду перевода его в режим ANSI. Кроме того, в пакете устанавливается

лируется внутренний признак, заставляющий передавать графические команды, как часть DCS-последовательности. Для открытия DCS-последовательности в нужном месте прикладной программы должна быть вызвана процедура DCSOPEN, а для закрытия ее — DCSCLOSE. DCS-последовательность закрывается также при вызове процедур MODE52, MODE100:

Если при выполнении графических построений не установлена область рулона для алфавитно-цифровой информации, то на выводимые графические примитивы может быть наложен текстовый курсор, что приведет к искажению графического изображения. Для того, чтобы это не случилось, желательно устанавливать область рулона, а поле вывода назначать так, чтобы область рулона была вне его. Однако иногда требуется производить вывод графической информации на полный экран. В этом случае нужно выключать текстовый курсор. Для этого в интерфейсных пакетах предусмотрена процедура TCURSOR с параметром ONOFF, включающим или выключающим текстовый курсор. Эта процедура посылает в дисплей ESC-последовательность, описанную в документации на алфавитно-цифровой дисплей.

Процедура SCROLLREGION устанавливает область рулона алфавитно-цифровой информации. Ее два параметра задают номера начальной и конечной строк области рулона (от 1 до 24). Строки нумеруются, начиная от 1, сверху вниз. Минимальный размер области рулона — 2 строки. После вызова этой процедуры текстовый курсор устанавливается в левый верхний угол экрана.

Процедура PUTTCURSOR устанавливает текстовый курсор дисплея на заданную строку и позицию в строке. Первый ее аргумент задает номер строки (1..24 с отсчетом от верхней строки экрана или области рулона), второй аргумент задает номер позиции в строке в диапазоне 1..80 или 1..132.

Примечание. Вызов процедур TCURSOR, PUTTCURSOR и SCROLLREGION не изменяет режим передачи графических команд и режим дисплея, установленный процедурами MODE100 и MODE52. Кроме того, вызов этих процедур прерывает DCS-последовательность.

20. НАЧАЛЬНАЯ УСТАНОВКА РЕЖИМОВ ГРАФИЧЕСКОГО ПРОТОКОЛА

При выполнении частичного сброса (CSI ! r в режиме ANSI) или операции СБРОС ДИСПЛЕЯ через меню выполняется частичный сброс дисплея и начальная установка гра-

фических процедур. Частичный сброс можно выполнить, обратившись к процедуре SOFTRESET, имеющейся в интерфейсных модулях. Начальная установка графических процедур выполняется также при выполнении операции НАЧАЛЬНЫЕ УСТАНОВКИ. Для графических команд устанавливаются следующие режимы:

- окно в мировых координатах — от 0 до 399 по горизонтали и от 0 до 239 по вертикали;
- поле вывода — полный экран;
- масштабирование включено;
- отсечение включено;
- координаты текущей позиции — 0,0;
- тип линии — 0 (сплошная);
- режим записи — 2 (занесение с прозрачными пробелами);
- графический курсор выключен;
- тип курсора — 6, размер 2, цвет 5;
- графический локатор выключен;
- тип локатора 0 (пересечение вертикальной и горизонтальной линий), цвет 10;
- мерцание 15-го цвета включено;
- текущий цвет — 5;
- цвет фона — 0;
- фонт номер 1;
- вектор базы символов (10,0), вектор вертикали — (0,10), вектор направления — (10,0);
- заполнение многоугольников, окружностей, эллипсов и т. д. выключено;
- образец заполнения 0 — сплошная заливка;
- границы заполняемых многоугольников рисуются;
- буфер подвижного рисунка сброшен;

Кроме всего этого, при выполнении этой операции прерывается текущая ESC- или DCS-последовательность. Прорисовки символов, загруженные командами LOADFONT, не изменяются.

ИСПОЛЬЗОВАНИЕ ПРОГРАММЫ ГРАФИЧЕСКОГО ПРОТОКОЛА KeyGP РУКОВОДСТВО ПРОГРАММИСТА

1. ВВЕДЕНИЕ

Программа графического протокола KeyGP добавляет к управляющей программе в ПЗУ KP1801PE2-181 (далее ПЗУ-181) или KP1801PE2-182 (далее ПЗУ-182) новые алфавитно-цифровые, графические и сервисные функции. После загрузки программы графического протокола, помимо реализации собственно графических функций, в КЦГД с ПЗУ-182 появляется возможность использовать команды загрузки прорисовок символов и кодов клавиш, таким образом, отрабатывается полная система команд дисплея типа VT220, а выбор режимов работы дисплея можно производить при помощи «меню». При использовании графического протокола вместе с ПЗУ-181 исправляется обработка некоторых ранее неправильно реализованных функций и также появляется возможность установки режимов дисплея посредством меню.

Программа графического протокола поставляется вместе с некоторыми вспомогательными модулями и библиотеками, облегчающими разработку прикладных программ.

В данной главе описано, какие компоненты входят в комплект поставки, как перенести их на пользовательский носитель для использования, как связывать программы, разработанные пользователем, с функциями графического протокола. Описана также программа GX, позволяющая пользователю непосредственно проверить исполнение команд графического протокола, и драйвер HC.SYS, позволяющий выводить копию экрана на печатающие устройства EPSON FX-80/800/1000, LX-800, Роботрон CM6329, CP-H80 и другие программно совместимые с перечисленными.

2. КОМПЛЕКТ ПОСТАВКИ

Программные средства графического протокола для КЦГД поставляются в виде гибкого магнитного диска и комплекта документации. Диск имеет специальный формат и состоит из двух частей. Первая часть диска имеет стандартную файловую структуру операционной системы ФОДОС-2 (RT-11, АДОС, TSX). На ней находятся следующие файлы:

- INSTAL.COM — командный файл для копирования файлов с дискеты на системный диск;
- INSTAL.IND — командный файл для программы IND для выборочного копирования файлов с дискеты;
- README.TXT — информация по текущей версии, не вошедшая в документацию;
- FGPACK.OBJ — модуль программного интерфейса для ФОРТРАНа;
- CGPACK.OBJ — модуль программного интерфейса для Си;
- PGPACK.OBJ — модуль программного интерфейса для Паскаля;
- PGLIB.OBJ — библиотека программного интерфейса для Паскаля;
- FGLIB.OBJ — библиотека программного интерфейса для ФОРТРАНа;
- CGLIB.OBJ — библиотека программного интерфейса для Си;
- GMACRO.MLB — библиотека макроопределений для Макро;
- GPDEFS.PAS — файл деклараций для программ на языке Паскаль;
- GX.SAV — программа-интерпретатор;
- HC.SYS — универсальный драйвер печати копии экрана;
- GPDEMO.SAV — демонстрационная программа для цветного монитора;
- GPMOND.SAV — демонстрационная программа для монохромного монитора;

Вторая часть дискеты, недоступная стандартным средствам операционной системы, содержит программу KeyGP в закодированном формате и процедуру создания на жестком диске с контроллером КЖД файла DW:KEYGP.SAV. Эта часть дискеты защищена от несанкционированного копирования. После загрузки с поставочной дискеты можно по выбору либо загрузить программу графического протокола в КЦГД (любое количество раз), либо произвести установку (запись)

программы графического протокола на винчестер (ограниченное количество раз). Записываемая копия имеет специальный формат, она не копируется стандартными средствами ОС, при операции «сжатия» она не перемещается по диску, а программа DUP выдает сообщение об ошибке чтения файла DW:KEYGP.SAV. Если в КЦГД после записи файла KEYGP.SAV был установлен другой тип ПЗУ, то этот файл необходимо переписать заново.

Программное обеспечение графического протокола защищено знаком (с) Copyright, а также специальными методами как от копирования, так и от «просмотра» процесса его работы. Разработчики гарантируют безопасность его работы в штатном режиме, но не несут ответственности за любые поломки и отказы, возникшие от попыток несанкционированного копирования программных средств, трассирования программ и их «взлома».

Ограничения на копируемость программного обеспечения графического протокола преследуют цель защиты прав разработчиков и организации — владельца права на тиражирование. Связанные с этими ограничениями неудобства минимальны, так как имеется некоторый запас для получения копий в случаях порчи содержимого жесткого диска. Для конфигураций ЭВМ без жесткого диска эту программу целесообразно хранить на системном диске, потому что она имеет большой объем. В этом случае графический протокол следует загружать в КЦГД с поставочной дискеты.

3. КОПИРОВАНИЕ ПРОГРАММЫ KeyGP НА ЖЕСТКИЙ ДИСК

Программа графического протокола хранится на поставочной дискете в особом формате, недоступном стандартным средствам операционной системы. Для копирования этой программы на жесткий диск (DW:) на этой же дискете записана специальная процедура. Для ее выполнения необходимо вначале командой В пультового режима ЭВМ или командой BOOT/FOREIGN операционной системы выполнить загрузку с поставочной дискеты. При этом начинает работать программа, которая определяет конфигурацию вычислительного комплекса, и в случае, когда контроллер жесткого диска отсутствует, производит автоматическую загрузку графического протокола. Это же происходит, если исчерпан счетчик числа копий, либо оказалось невозможным выполнить опера-

цию записи на поставочную дискету (например, заклеена боковая прорезь на конверте дискеты).

Если стартовая программа не обнаружила условий, препятствующих перезаписи на винчестер, то после выполнения необходимых предварительных действий она выводит на экран сообщение:

Оставшееся число копий : n

Копировать программу загрузки графического протокола на DW: (Д/Н) ?

В случае утвердительного ответа (Д или Y) будет выполнена запись файла KEYGP.SAV на жесткий диск, причем, если этот файл был ранее удален или разрушен в результате работы другой программы, но жесткий диск не был отформатирован, то запись произойдет без уменьшения счетчика выполненных копий. Для получения копии на жестком диске должна иметься свободная область размером около 200 блоков.

Полученная в результате записи копия настроена на тип управляющей программы (ПЗУ-181 или ПЗУ-182) и другие характеристики вычислительного комплекса.

После записи файла на экране появляется сообщение :
Файл DW : KEYGP.SAV создан

и одно из следующих сообщений:

У Вас остались n копии (n — 2,3,4)

У Вас осталась одна копия

Это была последняя копия

По завершении загрузки графического протокола или окончании операции копирования ЭВМ переводится в пультовый режим.

В процессе записи файла KEYGP.SAV могут появиться диагностические сообщения об ошибках, перечисленные ниже:

*****Ошибка чтения каталога*****

*****Ошибка записи каталога*****

*****Ошибка форматирования*****

*****Ошибка записи на винчестер*****

*****Ошибка чтения с винчестера*****

*****Ошибка верификации*****

*****Ошибка чтения с дискеты*****

— эти сообщения говорят об ошибках аппаратуры;

*****Неверный формат каталога*****

— жесткий диск не имеет стандартного оглавления в формате операционной системы RT-11 (ФОДОС, АДОС);

*****Переполнение каталога*****

*****Нет места на устройстве*****

— на жестком диске нет непрерывной свободной области длиной около 200 блоков или в оглавлении нет места для записи заголовка файла;

*****Файл DW:KEYGP.SAV уже существует*****

— на жестком диске уже имеется файл-загрузчик протокола.

4. ЗАГРУЗКА ПРОГРАММЫ ГРАФИЧЕСКОГО ПРОТОКОЛА

Если программа KEYGP.SAV записана на жесткий диск, то для загрузки графического протокола необходимо просто подать команду R KEYGP. Лучше всего, если эта команда будет в стартовом командном файле, автоматически выполняемом при загрузке ОС. Если же вычислительный комплекс не содержит жесткого диска, то для загрузки программы необходимо через команду В пультового режима ЭВМ загрузиться с поставочной дискеты. Если программа графического протокола еще не загружена, то экран очищается и появляется надпись «Идет загрузка», и через некоторое время экран вновь очищается. Процесс загрузки длится около одной минуты. Затем в КЦГД посылается время, установленное в системе (только для ПЗУ-182), и передается записанный в файле графического протокола буфер сохраненных установок, и эти режимы устанавливаются в дисплее. Системное время может быть установлено программой DATIME.SAV, запускаемой из командного файла.

Повторный вызов программы KEYGP.SAV или повторная загрузка с поставочной дискеты при загруженном протоколе приводит к записи буфера сохраненных установок в теле программы на диске. Это дает возможность сохранения параметров дисплея, выбранных оператором для последующих загрузок. Сохраненные режимы дисплея будут автоматически устанавливаться при последующих загрузках протокола.

Убедиться в том, что программа графического протокола загружена в КЦГД, можно, нажав клавишу УСТ РЕЖИМА. Если протокол загружен, то в нижних 8 строках будет выведено меню установки режимов, в противном случае дисплей перейдет в автономный режим. Для возврата в исходное

состояние необходимо еще раз нажать клавишу УСТ РЕ-ЖИМА.

При загрузке с дискеты могут появиться следующие сообщения:

****Ошибка позиционирования КМД****
****Контроллер КМД неисправен****
?BOOT—U—I/O error

****Ошибка чтения с дискеты****
****Неправильная информация****
****Ошибка номер xx ****

Первые два свидетельствуют о неисправности аппаратуры ЭВМ. Следующие два появляются при обнаружении ошибки чтения на дискете. В этом случае можно попытаться вынуть дискету, снова поставить ее и повторить загрузку. Два последних сообщения говорят о серьезном нарушении в программе. В этом случае следует обратиться к поставщику данного программного продукта.

5. КОПИРОВАНИЕ ФАЙЛОВ С ПОСТАВОЧНОЙ ДИСКЕТЫ

Файлы интерфейсных модулей и библиотек, необходимые для разработки прикладных программ, перед использованием необходимо скопировать с поставочной дискеты на диск, на котором записана операционная система. Копирование выполняется стандартными средствами операционной системы. Необходимо вставить дискету в привод накопителя, командой ASSIGN назначить на него имя DK: и подать команду \$@INSTAL. Если на системном диске содержится программа IND, то командой:

```
IND INSTAL.IND
```

можно запустить процедуру выборочного копирования файлов, которая позволяет также включить файлы объектных библиотек в системную библиотеку SYSLIB.OBJ или библиотеку PASCAL.OBJ.

Файл README.TXT с дополнениями и замечаниями, которые не включены в комплект поставляемой документации, можно либо просмотреть с помощью редактора текстов, либо распечатать на дисплее или принтере командами TYPE или PRINT соответственно.

6. СВЯЗЬ С ПОЛЬЗОВАТЕЛЬСКИМИ ПРОГРАММАМИ

Прикладная программа пользователя компонуется, в зависимости от языка, на котором она написана, с одним из интерфейсных объектных модулей: FGPACK.OBJ — для программ на Фортране, CGPACK.OBJ — для программ на Си, PGPACK.OBJ — для программ на Паскале и Макроассемблере. Имя интерфейсного модуля указывается в команде LINK после имени объектного файла программы пользователя, например:

```
LINK PROG,SY:FGPACK
```

Каждый файл представляет собой один объектный модуль, в котором имеются все точки входа. В скомпонованной загрузочной программе он занимает около 2К байт. Если программа использует не все функции графического протокола и желательно уменьшить ее размер, то ее можно компоновать с библиотекой FGLIB.OBJ, CGLIB.OBJ, PGLIB.OBJ, в зависимости от языка, на котором написана программа. Для программы на языке Фортран библиотека подключается с помощью ключа /LINKLIB:SY:FGLIB, например:

```
EXEC PROG.FOR/LINK:SY:FGLIB или  
LINK PROG.OBJ,SY:FGLIB
```

При компоновке программы, написанной на языке Паскаль или Си, соответствующая библиотека также должна быть указана в команде LINK.

Библиотеки FGLIB, PGLIB, CGLIB можно объединить с библиотеками исполняющей системы соответствующего языка программирования. Это позволяет не указывать их каждый раз в команде LINK. FGLIB.OBJ может быть объединен с библиотекой FORLIB.OBJ или SYSLIB.OBJ. PGLIB.OBJ может быть объединен с библиотекой PASCAL.OBJ, а CGLIB.OBJ можно объединить с CXLIB.OBJ. Это выполняется командой LIBRARY:

```
LIBR SY:FORLIB FGLIB  
LIBR SY:PASCAL PGLIB  
LIBR SY:CXLIB CGLIB
```

Эти действия производятся также при выполнении файла INSTAL.IND, если на соответствующий вопрос программы IND будет дан утвердительный ответ.

При разработке программ на языке Паскаль необходимо описать все используемые процедуры пакета. Объявления всех имеющихся процедур содержатся в файле GPDEFS.PAS,

и нужно лишь с помощью редактора текстов «вырезать» из него требуемые и включить в текст программы.

При разработке программы на Макроассемблере из нее можно обращаться к точкам входа пакета PGPACK.OBJ, либо использовать библиотеку PGLIB.OBJ. Для формирования списка параметров можно использовать макроопределения, содержащиеся в библиотеке GMACRO.MLB. Эти макроопределения имеют те же имена, что и соответствующие графические функции, и имеют тот же порядок аргументов. Макроопределения формируют обычно следующую последовательность команд:

```
MOV ARG1,— (SP)
MOV ARG2,— (SP)
...
MOV ARGn,— (SP)
CALL PROGR
```

Параметры, как и в языке Паскаль, передаются по значению, кроме символьных строк и массивов координат, для которых указываются их адреса. Адреса указываются также для аргументов, значения которых возвращаются, как в функции запроса координат локатора или текущей позиции. Если значением аргумента является константа, то ее необходимо указывать со знаком непосредственного операнда «#». Пример использования макрокоманды LINE :

```
LINE #0,#0,X1,Y1,#5
```

Будет нарисован отрезок из точки с координатами 0,0 до точки, координаты которой хранятся в ячейках X1, Y1, цветом с индексом 5.

Для доступа к библиотеке GMACRO.MLB она должна быть указана в команде MACRO при трансляции программы:

```
MACRO PROGR,SY:GMACRO/LIBR
```

Эта библиотека может также быть указана непосредственно в самой программе с помощью директивы:

```
.LIBRARY «SY:GMACRO»
```

Макроопределения из библиотеки могут считываться либо автоматически, если в программе указана директива «.ENABL MCL», либо посредством директивы «.MCALL», что несколько быстрее автоматического вызова. В последнем случае в

директиву «.MCALL» следует включить вызов макрокоманды «...GM0». Оттранслированная программа компонуется с модулем PGPACK.OBJ или с библиотекой PGLIB.OBJ.

Все подпрограммы интерфейсных модулей не изменяют содержимого регистров R0—R5 центрального процессора. Для языков Фортран и Си в регистре R0 находится возвращаемое значение функции.

7. ПРОГРАММА GX

Программа GX предназначена для просмотра и исполнения в непосредственном режиме команд графического протокола, вызываемых через интерфейсный пакет для языка Паскаль.

После запуска она выводит подсказку: GX> и считывает строку по запросу .GTLIN, что позволяет вызывать ее с одинаковым успехом и с терминала, и из командного файла. Введенная строка анализируется следующим образом:

— если строка начинается с символа «/» (например, /H), то распечатывается краткая справочная информация по работе с программой;

— если строка начинается с точки с запятой «;», то эта строка просто печатается на терминале (без начального символа «;»). Этот режим может быть использован (главным образом, при запуске из командного файла) для вывода на экран различных сообщений или послышки управляющих последовательностей;

— если строка начинается с символа «?» (например, ?CIR), то на экран выводятся определения всех процедур и функций пакета, начинающихся с данной строки. Этот режим может быть использован в качестве быстрого справочника по пакету;

— в остальных случаях вызывается первая процедура или функция пакета, начало имени которой совпадает с началом введенной строки. После того, как было найдено подходящее имя процедуры (функции), считываются (из той же строки) аргументы вызова.

Имя процедуры отделяется от аргументов пробелом или скобкой, а сами аргументы друг от друга — любым нецифровым символом (например, пробелом или запятой).

Целочисленные (INTEGER) и символьные (CHAR) аргументы вводятся в той же строке, что и имя процедуры, при-

чем целочисленные параметры вводятся десятичными числами, а символьные — либо десятичными числами (код символа), либо самим символом, перед которым стоит апостроф. Таким образом, вызовы:

```
GX> CHARAC 100,100,1,65
GX> CHARAC 100,100,1,'A
```

аналогичны. Оба вызова выведут в точке (100,100) символ «А» цветом с индексом 1.

Аргумент, являющийся символьным или целочисленным массивом или строкой текста, в самой команде опускается, и вводится в следующей строке. Для строки текста выводится подсказка: Text>, в ответ на которую следует ввести строку текста. Она будет заканчиваться кодом 0. Для целочисленного массива будет выведена подсказка: Int array>, в ответ на которую следует ввести соответствующее число десятичных чисел. Их количество программа GX не проверяет. Символьные массивы вводятся после подсказки Char array>. Они могут задаваться и как десятичные коды, и как символы с предшествующим апострофом.

Если была вызвана функция, то ее возвращаемое значение (типа INTEGER) распечатывается на экране в виде десятичного числа. Также распечатываются возвращаемые целочисленные аргументы (типа VAR X,Y : INTEGER). Заданные значения VAR-параметров игнорируются.

Если программа GX запущена с терминала или из командного файла в режиме SET TT NOQUIET (режим по умолчанию), то все вводимые ею данные отображаются на экране, что вызывает сдвиг (рулон, скроллинг) изображения на экране. Для предотвращения этого следует задать зону рулона, например командой SCROLL 20,24. После этого следует установить курсор в зону рулона, например командой PUTTCUR 24,1.

При вводе команд с терминала следует использовать режим совместимости с VT52, так как в этом случае не возникнет затруднений, связанных с выдачей кодов, закрывающих командные последовательности.

8. ДРАЙВЕР ПЕЧАТИ КОПИИ ЭКРАНА HC.SYS

Драйвер HC.SYS предназначен для вывода копии содержимого экрана на печатающие устройства EPSON FX-80/800/1000, LX-800, CP-H80, Роботрон CM6329 и совме-

стимые с ними. Печать копии экрана инициируется при нажатии клавиши ПЕЧАТЬ КАДРА, находящейся в верхнем ряду клавиатуры, совместно с клавишей СУ или ВР. При нажатии ее с клавишей СУ печать производится поперек листа в четырех градациях серого — за счет печати с разным числом проходов. При нажатии клавиши ПЕЧАТЬ КАДРА совместно с клавишей ВР экран печатается большего размера вдоль листа в 16 градациях. Для имитации такого числа градаций пиксели представляются матрицей 2×4 печатных точек и выводятся разным количеством проходов. Для остановки печати необходимо нажать клавишу ПЕЧАТЬ КАДРА без нажатия СУ и ВР. Печать может быть также инициирована по запросу прикладной программы командой HARD-COPY.

Для того, чтобы драйвер можно было использовать, он должен быть записан на системный диск и настроен на параметры генерации системы посредством команд:

```
SET HC SYSGEN
INSTALL HC
```

Эти команды требуются лишь один раз после копирования драйвера на системный диск.

Драйвер может выполнить печать лишь тогда, когда он сделан активным с помощью команды:

```
SET HC ON
```

Для нормальной работы драйвера требуется, чтобы в момент нажатия клавиши ПЕЧАТЬ КАДРА программа графического протокола KeyGP была загружена в КЦГД и были разрешены прерывания от клавиатуры терминала.

Когда драйвер больше не нужен, то память, которую он занимает, можно освободить командой:

```
SET HC OFF
```

В зависимости от типа печатающего устройства драйвер может функционировать в различных режимах. С принтером EPSON FX-800 или FX-1000 он работает в режиме графической печати с плотностью 144 точки на дюйм. При этом обеспечивается наиболее качественная печать, особенно для 16 градаций яркости. Этот режим драйвера устанавливается командой:

```
SET HC MODE=0
```

Другие печатающие устройства не обеспечивают такой режим. Драйвер посылает им информацию для плотности пе-

чати 240 точек на дюйм, при этом для сохранения пропорций изображения программным способом моделируется печать с плотностью 144 точки на дюйм, что приводит к незначительному ухудшению качества. Этот режим устанавливается командой:

SET HC MODE=2

Перед печатью изображений в 16 градациях производится взвешивание цветов, как в совместимой системе цветного телевидения. При этом наиболее яркий цвет может печататься или черным (негатив) или белым (позитив). Нужный режим устанавливается командой:

SET HC NEGATI

или

SET HC POSITI

Установленный этими командами режим влияет также на печать в режиме четырех градаций, при этом взвешивание не производится, а цвета моделируются следующими количествами проходов:

цвета (4/16)	0/0	1/5	2/10	3/15
число проходов:				
негатив :	0	1	2	3
позитив :	2	0	1	3

Еще один режим, который использует плотность печати 72 точками на дюйм, позволяет получать пропорциональные изображения в четырех градациях серого, при этом разные яркости моделируются разным числом проходов. Он устанавливается командой:

SET HC MODE=1

В отличие от описанных выше режимов этот режим позволяет печатать только графические изображения низкого разрешения и только поперек листа, независимо от того, было ли нажато СУ/ПЕЧАТЬ КАДРА или ВР/ПЕЧАТЬ КАДРА.

В этом режиме каждому из 16 цветов посредством меню можно назначить одну из четырех градаций, для этого выбор режима следует производить также через меню.

В зависимости от типа интерфейса печатающего устройства для него может потребоваться инверсия сигналов данных. Для управления инверсией имеются команды:

SET HC DATA11

SET HC DATA01

Первая из них устанавливает режим передачи данных без инверсии (для интерфейса ИРПР устройства Роботрон

CM6329). С помощью второй устанавливается режим передачи данных с инверсией для устройств с интерфейсом ИРПР—М—EPSON и др. Инверсия данных может быть задана переключателем на плате ЭВМ, поэтому может потребоваться экспериментально подобрать нужный режим.

Установленные режимы работы драйвера можно посмотреть с помощью команды:

SET HC LIST

При этом будет выдано сообщение вида: HC is set OFF,NEGATI,DATA01.MODE=2, это соответствует режимам, установленным при поставке.

Перечисленные выше команды изменяют либо загруженную в память копию драйвера, либо находящуюся на системном диске, либо обе. Для этого в конце команды через запятую нужно указать или параметр MEMORY (или NODISK) — для модификации копии только в памяти, или DISK (NOMEMORY) — для модификации копии только на диске. Если ни один из этих параметров не указан, то изменяются обе копии.

Параметры работы драйвера можно изменять в режиме меню. Для этого необходимо подать команду:

SET HC MENU

По этой команде на экран выводится меню, по полям которого можно передвигаться вверх и вниз с помощью клавиш управления курсором, изменяя режимы нажатием клавиши ВВОД или ВК. Выход из меню и сохранение установок производится выбором поля «Выход». С правой стороны экрана отображается SET-команда, эквивалентная выбранному режиму.

При выборе поля установки режима печати и установленном режиме 1 выводится дополнительная таблица назначений количества проходов цветам. Для входа в эту таблицу необходимо нажать клавишу «курсор влево» или «курсор вправо». Значение числа проходов изменяется нажатием клавиши ВК или ВВОД. Движение по таблице производится с помощью клавиш управления курсором «вправо», «влево», выход в основное меню — клавишами «вверх» или «вниз».

Нажатие клавиши СУ/С приводит к выходу из меню без изменения ранее установленных режимов.

ОТЛИЧИЯ ПЗУ КР1801РЕ2-182 ОТ КР1801РЕ2-181

В ПЗУ-182 по сравнению с ПЗУ-181 введены новые функции, а также исправлены имевшиеся ошибки. Загружаемая в КЦГД программа графического протокола KeyGP дополняет управляющую программу ПЗУ-182 возможностью полной эмуляции дисплея VT220, а вариант ее для ПЗУ-181 также исправляет некоторые ошибки, имеющиеся в управляющей программе.

В ПЗУ-182 введены следующие новые функции:

- работа в режиме дисплея VT200;
- возможность послышки и приема 8-битовых эквивалентов ESC-последовательностей в режиме VT200;
- прием 8-битовых символов и послышка 8-битовых символов с клавиатуры;
- вставка и удаление строк под курсором;
- режим вставки символов при вводе;
- сдвигка и раздвигка символов в строке;
- стирание заданного числа символов в строке;
- выборочное стирание текста;
- режим локального эха, обеспечивающий работу в полудуплексном режиме;
- буферизация при выводе в ЭВМ символов с клавиатуры и сообщений дисплея;
- установка цвета символов и фона клавишами ПРЕРЫВ и ПРОДОЛЖ;
- в режиме совместности клавиши ПМ, ИСП, Ф17, Ф18, Ф19, Ф20, НТ, ВСТ, УДАЛ, ВЫБР, ПРЕД КАДР, СЛЕД КАДР посылают такие же последовательности, что и контроллер КСМ ПЭВМ ДВК-3;
- обрабатываются клавиши [{}], [~], [ъ].

ПЗУ-181 совместно с программой графического протокола KeyGP также обеспечивает буферизацию клавиатуры, установку цвета символов и фона, и обработку клавиш верхнего ряда подобно контроллеру КСМ.

В ПЗУ-182 по сравнению с ПЗУ-181 исправлены следующие ошибки:

- при приеме команд перехода в режим 80 или 132 символа экран очищался лишь тогда, когда режим действительно изменялся, что соответствует дисплею VT100. Управляющая программа ПЗУ-182 всегда очищает экран дисплея при

приеме такой последовательности, даже если режим не изменяется. Это соответствует работе дисплея VT240;

- при стирании в режиме 132 символа командой ESC [1 К первого символа строки или командой ESC [К последнего символа строки происходил сбой управляющей программы, приводящий к полному сбросу дисплея. Программа графического протокола KeyGP исправляет эту ошибку;

- при переходе дисплея из режима 132 символа в режим 80 символов или обратно, если курсор находился в строке с символами двойной ширины, адресация курсора выполнялась неправильно;

- вертикальные линии в прорисовках символов выводятся толщиной в одну точку, из-за чего они в режиме 80 символов изображаются недостаточно контрастными. В ПЗУ-182 вертикальные линии выводятся удвоенными по толщине, что делает их более удобочитаемыми на цветном мониторе;

- при нажатии некоторых клавиш, как ЗБ, ТАБ, ФКС, КМП, в режиме автоповтора выдавался код не этой клавиши, а код алфавитно-цифровой клавиши или клавиши управления курсором, нажатой последней. Программа графического протокола KeyGP исправляет эту ошибку;

- при приеме дисплеем кода РУС или ЛАТ тут же изменялось состояние индикатора на клавиатуре — путем послышки на клавиатуру специальной команды. Если в этот момент клавиатура передавала код нажатой клавиши, то эта послышка искажалась. Программа графического протокола KeyGP исправляет этот недостаток. ПЗУ-182 посылает команду на клавиатуру лишь по включению курсора, которое происходит через 1/30 секунды после приема последнего символа в непрерывной последовательности. При работе же ПЗУ-182 в режиме 8 бит коды РУС и ЛАТ вообще не изменяют состояние индикатора клавиатуры;

- при нажатии совместно с клавишей СУ клавиши ТАБ или ВК вместо ее кода посылался ее код плюс 100. Программа графического протокола KeyGP исправляет эту ошибку;

- ПЗУ-181 не обрабатывает нажатие клавиш [{} и {}] в четвертом ряду клавиатуры, а также клавиши [~] слева от ВК и [ъ];

- из-за отсутствия внутренней буферизации клавиатуры КЦГД зависал при нажатии нескольких клавиш в то время, когда центральный процессор не мог принять информацию. В автономном режиме это приводило к тому, что при нажатии клавиш, посылающих несколько символов, на экране отображался последний символ последовательности. Про-

грамма графического протокола KeyGP исправляет эту ошибку;

— при указании в команде установки зоны рулона в качестве нижней границы зоны 24-й строки, зона рулона устанавливалась на полный экран. Программа графического протокола KeyGP исправляет эту ошибку;

— команды ESC F и ESC G (включение и выключение вывода псевдографики в режиме VT52) обрабатывались как назначение G3 в GL и отмена этого назначения, при этом символы РУС и ЛАТ прерывали вывод псевдографики. Программа графического протокола KeyGP исправляет эту ошибку. В ПЗУ-182 эти команды обрабатываются как назначение набора псевдографики на G0, то есть как SCS-последовательности;

— изменения в назначении наборов G0—G3 отражались на выводе символов лишь после выполнения команды назначения измененного набора на GL, например, после приема символа РУС или ЛАТ. Программа графического протокола KeyGP исправляет эту ошибку;

— в наборе дополнительных знаков европейских алфавитов символы, на которые не назначены прорисовки, изображались в виде пробелов. ПЗУ-182 эти символы выводит в виде обращенного вопросительного знака (символ ошибки);

— в режиме автоперевода курсор переходит на следующую строку сразу же после вывода символа в последнюю позицию строки. ПЗУ-182 выполняет автоперевод при попытке повторного вывода символа в последнюю позицию, а до того курсор остается в этой же позиции.

СОДЕРЖАНИЕ

ФУНКЦИИ СИМВОЛЬНОГО ДИСПЛЕЯ НА БАЗЕ КЦГД. РУКОВОДСТВО ОПЕРАТОРА

	Стр.
1. Введение	3
2. Кодирование входной информации	4
3. Алфавиты	6
4. Сдвиг информации по экрану (рулон)	9
5. Управляющие символы кодовой таблицы СО	10
6. Управляющие символы кодовой таблицы С1	11
7. ESC — последовательности	12
8. Режим отображения управляющих символов	12
9. Режим совместимости с VT52 и KCM	13
10. Режим ANSI	15
10.1. Управляющие последовательности режима ANSI	16
10.2. ESC — последовательности	18
10.3. Переключение алфавитов	19
10.4. Изменение атрибутов строк	21
10.5. Уровни управления режима ANSI	22
10.6. 7- или 8-битовая передача управляющих символов	23
10.7. Установка и сброс режимов работы дисплея	23
10.7.1. Блокирование клавиатуры	24
10.7.2. Режим вставки/замещения	24
10.7.3. Локальное эхо	24
10.7.4. Перевод строки — новая строка	25
10.7.5. Режим клавиш управления курсором	25
10.7.6. Переход в режим совместимости	26
10.7.7. 80 или 132 символа в строке	26
10.7.8. Управление рулоном	26
10.7.9. Установка негативного/позитивного экрана	26
10.7.10. Режим относительной адресации курсора в области рулона	27
10.7.11. Автоперевод курсора в новую строку	27
10.7.12. Автоповтор клавиатуры	27
10.7.13. Изменение режима вывода алфавита	28
10.7.14. Включение/выключение индикации курсора	28
10.7.15. Специальный режим клавиш верхнего ряда	28
10.8. Установка области рулона	29
10.9. Вставка и удаление строк	29
10.10. Задание атрибутов символов	30
10.11. Управление курсором	30
10.12. Установка и сброс позиций табуляции	32
10.13. Стирание текста в пределах строки и экрана	32
10.14. Выборочное стирание	33
10.15. Раздвижка и сдвижка символов в строке	34
10.16. Сброс дисплея (начальная установка)	35
10.17. Сообщения от дисплея в ЭВМ	36
10.18. Заполнение экрана для настройки ЭЛТ	37
11. Работа клавиатуры	37
11.1. Дополнительная цифровая клавиатура	41
11.2. Клавиши управления курсором	43
11.3. Прочие функциональные клавиши	43
12. Установка цвета экрана и символов	46

13.	Загрузка дополнительного ПО в ОЗУ КЦГД	47
14.	Дополнительные функции текстового режима, обеспечиваемые программой KeyGP	47
14.1.	Клавиши, программируемые пользователем	48
14.2.	Загрузка новых прорисовок символов пользователем	50
14.3.	Сиксельный формат для графики	52
15.	Установка параметров дисплея в режиме меню	54
15.1.	Главная таблица установки режимов	56
15.2.	Таблица установки режимов дисплея	58
15.3.	Таблица установки общих функций	60
15.4.	Таблица установки режимов клавиатуры	61
15.5.	Таблица установки позиций табуляторов	63
15.6.	Таблица установки цветов	63

**ГРАФИЧЕСКИЙ ПРОТОКОЛ KeyGP ДЛЯ КЦГД.
РУКОВОДСТВО ПРОГРАММИСТА**

1.	Введение	65
2.	Общие понятия и термины	68
3.	Команды графического протокола	70
4.	Атрибуты графических примитивов	73
5.	Установка режима разрешения	73
6.	Команды задания окна и области вывода, управления масштабированием и отсечением	74
7.	Очистка области вывода	79
8.	Установка цвета и режима вывода	80
9.	Построение точек, маркеров и отрезков прямых, установка типа линии	84
10.	Выбор и определение штриховки и узора	90
11.	Заполнение (раскраска) фигуры узором или штриховкой	93
12.	Построение прямоугольников, многоугольников, ломаных линий, полимаркеров	99
13.	Рисование плавных кривых	106
14.	Рисование окружностей, эллипсов, их дуг, сегментов, секторов	109
15.	Вывод символов и текста, и установка их параметров	116
16.	Рисование графического курсора и локатора, запрос координат курсора и локатора	124
17.	Загрузка изображений, чтение изображений и кода одного пиксела	132
18.	Вывод твердой копии	136
19.	Подпрограммы установки режимов дисплея	138
20.	Начальная установка режимов графического протокола	139

**ИСПОЛЬЗОВАНИЕ ПРОГРАММЫ
ГРАФИЧЕСКОГО ПРОТОКОЛА KeyGP.
РУКОВОДСТВО ПРОГРАММИСТА**

1.	Введение	141
2.	Комплект поставки	142
3.	Копирование программы KeyGP на жесткий диск	143
4.	Загрузка программы графического протокола	145
5.	Копирование файлов с поставочной дискеты	146
6.	Связь с пользовательскими программами	147


7.	Программа GX	149
8.	Драйвер печати копии экрана HC.SYS	105
ПРИЛОЖЕНИЕ. Отличия ПЗУ KP1801PE2—182 от KP1801PE2—181		154

Ответственный за выпуск **М. Г. Бойкова**
Редактор **Т. А. Савельева**
Корректор **В. Н. Лыткина**

Изд. № 30. Сдано в набор 11.09.90 г. Подписано в печать 28.11.90 г.
Формат 60×84¹/₁₆. Бум. тип. № 1. Гарнитура литературная.
Печать высокая. Объем 9,3 усл. печ. л. Тираж 20 000 экз. Заказ 2397.

Ленинградское отделение РППО «Союзбланкоиздат».

Великолукская городская типография управления издательств,
полиграфии и книжной торговли Псковского облисполкома,
182100, г. Великие Луки, ул. Полиграфистов, 78/12.



(C) EKOLLO.SPE.FU