

.TITLE ROM

.ASECT

. =100000 ; Начальный адрес ПЗУ

```

; *****
; *
; *      ПОДПРОГРАММЫ ОБСЛУЖИВАНИЯ МЕНЮ УСТАНОВКА И ЗАГРУЗКА      *
; *
; *      Адреса 100000 - 103553      *
; *
; *****

```

; П/п установки начальных значений в меню УСТАНОВКА и ЗАГРУЗКА

```

100000$:JSR      R5,110712$      ; П/п заносит данные из области
      .WORD      100012$,7152      ; по адресу 100012 в 7152
      RETURN
100012$: .WORD      10.      ; Количество слов
      .WORD      177701,0,0      ; Пункт УСТ не выбран, VT-52, 80x24
      .WORD      7,1,1,7      ; Цвета символа, знакоместа, экрана и курсора
      .WORD      0,0      ; Рулон плавный, таймер включен
      .WORD      177701      ; Пункт ЗАГРУЗКА не выбран

```

; П/п обслуживания главного меню УСТАНОВКА

```

100040$:TST      @#23170      ; Ожидание завершения скроллинга
      BNE      100040$
      MOV      @#2476, -(SP)      ; Сохранить адрес пользовательского экрана
      MOV      #4672,@#2476      ; Включить экран установки
      MOV      #22754,R5      ; Сделать текущей таблицу видеопараметров
      CALL     110210$      ; экрана установки
      JSR      R5,105346$      ; Перевести клавиатуру в системный режим
      .WORD      107310$,11054      ; Данные для системного режима
      CALL     104054$      ; Очистить буфер клавиатуры
      CLR      @#7232      ; Разрешение алфавитно-цифровой клавиатуры
      MOV      SP,@#7142
      EMT      52      ; Вывод УСТ в верхней строке
      .WORD      107533$
100120$:EMT      44      ; Вывод заголовка
      .WORD      102133$
100124$:MOV      #4,R0
      JSR      R4,101202$      ; Вывод меню и запрос пункта
      .WORD      101644$      ; Адрес списка для меню УСТАНОВКА
      BMI      100172$      ; Нажата <Enter> или УПР-@ ?
      CALL     @100150$(R2)      ; Обслуживание выбранного пункта
      BMI      100172$      ; Нажата <Enter> или УПР-@ ?
      BR       100124$

```

; Адреса п/п обслуживания пунктов меню УСТАНОВКА

```

100150$: .WORD      100172$,100334$,100372$,100432$,100476$
      .WORD      100536$,100576$,100630$,100730$

```

; Если нажаты <Enter> или УПР-@

```

100172$:CMP      R0,#17772      ; Нажата УПР-@ (выход в монитор ПП) ?
      BEQ      100302$
      MOV      @#7142,SP
      EMT      52      ; Стирание в верхней строке надписи УСТ
      .WORD      107524$
      EMT      52
      .WORD      107542$
      TST      @#7146      ; Предыдущий режим клавиатуры был системным ?
      BNE      100232$      ; Да
      JSR      R5,105400$      ; Восстановить пользовательский режим
      .WORD      107310$,11054      ; Данные для пользовательского режима
100232$:CALL     110204$      ; Включение пользовательского экрана
      MOV      (SP)+,@#2476

```

```

CLR      @#7070      ; Очистить признак вызова
TST      @#7072      ; Вызов происходил из меню ЗАГРУЗКА ?
BNE      100300$     ; Да
100254$: MOV    #2, @#7230 ; Установить признак выполнения TRAP 2
MOV      #2, @#22546 ; Установить признак выполнения TRAP 4
MOV      #2, @#7042 ; Адресовать ASCII-коды клавиатуры каналу 0
TRAP     4           ; Прочитать символ с канала 0 (если есть)
100300$: RETURN
100302$: MOV    #14, R0 ; Очистить экран
EMT      42
MOV      #100120$, -(SP) ; Адрес возврата в меню УСТАНОВКА
MOV      SP, @#7144
JMP      @#160576 ; Переход к программе монитора ПП

```

; П/п обслуживания подменю СИСТЕМА КОМАНД

```

100324$: CALL   @100346(R2) ; Вызов п/п по текущему пункту меню
MOV      #4, R0
100334$: JSR    R4, 101202$ ; Вывод подменю и запрос пункта
        .WORD   101672$     ; Адрес списка подменю СИСТЕМА КОМАНД
BEQ      100324$
RETURN

```

; Адреса п/п для обслуживания подменю СИСТЕМА КОМАНД

```

100346$: .WORD   162572$, 113670$, 113712$

```

; П/п обслуживания подменю ФОРМАТ ЭКРАНА

```

100354$: MOV    R2, R0 ; Получение номера формата
        ASR     R0     ; 0 - 80x24, 1 - 40x24, 2 - 20x24, 3 - 10x24
        DEC     R0
        CALL    116502$ ; Обращение к п/п установки формата экрана
        MOV     #2, R0
100372$: JSR    R4, 101202$ ; Вывод меню и запрос пункта
        .WORD   101704$     ; Адрес списка подменю ФОРМАТ ЭКРАНА
BEQ      100354$
RETURN

```

; П/п для обслуживания подменю ЦВЕТ СИМВОЛА

```

100404$: MOV    R2, R0 ; Получение номера цвета
        ASR     R0
        DEC     R0
        MOV     R5, -(SP) ; Сохранить адрес текущей таблицы видеопар.
        MOV     #22656, R5 ; Загрузить адрес таблицы экрана пользователя
        CALL    116052$ ; Обращение к п/п установки цвета символа
        MOV     (SP)+, R5
        MOV     #2, R0
100432$: JSR    R4, 101202$ ; Вывод меню и запрос пункта
        .WORD   101722$     ; Адрес списка подменю ЦВЕТ СИМВОЛА
BEQ      100404$
RETURN

```

; П/п для обслуживания подменю ЦВЕТ ЗНАКОМЕСТА

```

100444$: MOV    R2, R0 ; Получение номера цвета
        SUB     #2, R0
        MOV     113512$(R0), R0
        MOV     R5, -(SP)
        MOV     #22656, R5
        CALL    116130$ ; Обращение к п/п установки цвета знакоместа
        MOV     (SP)+, R5
        MOV     #2, R0
100476$: JSR    R4, 101202$ ; Вывод меню и запрос пункта
        .WORD   101750$     ; Адрес списка подменю ЦВЕТ ЗНАКОМЕСТА
BEQ      100444$
RETURN

```

; П/п для обслуживания подменю ЦВЕТ ЭКРАНА

```
100510$:MOV    R2,R0          ; Получение номера цвета
        ASR     R0
        DEC     R0
        MOV     R5, -(SP)
        MOV     #22656,R5
        CALL    116170$      ; Обращение к п/п установки цвета экрана
        MOV     (SP)+,R5
        MOV     #2,R0
100536$:JSR     R4,101202$
        .WORD   101776$      ; Адрес списка подменю ЦВЕТ ЭКРАНА
        BEQ     100510$
        RETURN
```

; П/п для обслуживания подменю ЦВЕТ КУРСОРА

```
100550$:MOV    R2,R0          ; Получение номера цвета
        ASR     R0
        DEC     R0
        MOV     R5, -(SP)
        MOV     #22656,R5
        CALL    116226$      ; Обращение к п/п установки цвета курсора
        MOV     (SP)+,R5
        MOV     #2,R0
100576$:JSR     R4,101202$
        .WORD   102024$      ; Адрес списка подменю ЦВЕТ КУРСОРА
        BEQ     100550$
        RETURN
```

; П/п для обслуживания подменю РУЛОН

```
100610$:MOV    R5, -(SP)
        MOV     #22656,R5
        CALL    @<100642$-2>(R2); Обращение к п/п-ам установки рулона
        MOV     (SP)+,R5
        MOV     #2,R0
100630$:JSR     R4,101202$
        .WORD   102052$      ; Адрес списка подменю РУЛОН
        BEQ     100610$
        RETURN
```

; Адреса п/п-ам для установки рулона

```
100642$: .WORD  100650$,100656$,100674$
100650$:CLR     @#7136        ; Установить плавный рулон
        BR      100664$
100656$:MOV     #2,@#7136      ; Установить дискретный рулон
100664$:MOV     #2,74(R5)      ; Установить присутствие рулона
        RETURN      ; 74 - смещение в таблице видеопараметров
100674$:CLR     74(R5)        ; Установить отсутствие рулона
        RETURN
```

; П/п для обслуживания подменю ТАЙМЕР

```
100702$:BIS     #1000,@#177054 ; Выключить таймер в канале ЦП
        SUB     #2,R2          ; Текущий пункт - включен ?
        BNE     100724$        ; Нет
        BIC     #1000,@#177054 ; Включить таймер в канале ЦП
100724$:MOV     #2,R0
100730$:JSR     R4,101202$
        .WORD   102066$      ; Адреса списка подменю ТАЙМЕР
        BEQ     100702$
        RETURN
```

; П/п обслуживания главного меню ЗАГРУЗКА

```
100742$:CLR     @#7042        ; Направлять ASCII-коды запросам EMT 22
        CLR     -(SP)          ; Номер устройства = 0
100750$:MOV     #14,R0        ; Очистить экран
        EMT     42
```

```

100756$:CLR      @SP
          JSR      R4,101616$      ; Позиционирование курсора
          .BYTE    2,5              ; Y=2, X=5
          MOV      #-6,@#7150      ; Запрет вывода в верхнюю инф. строку
          CLR      @#23164         ; Разрешение использования курсора
          EMT      44               ; Вывод заголовка
          .WORD    103114$
101004$:MOV      #4,R0
          JSR      R4,101202$      ; Вывод меню и запрос пункта
          .WORD    102100$         ; Адрес списка меню ЗАГРУЗКА
          BEQ      100756$
          BMI      101146$         ; Нажаты <Enter>,<ИСП> или <0> ?
          ADD      R2,PC           ; Переход по номеру пункта (нажата вправо)
          BR       100756$
          BR       101044$         ; Диск
          BR       101102$         ; Кассета ПЗУ
          BR       101004$         ; Сеть
          BR       101004$         ; Стык С2
          BR       101004$         ; Магнитофон
          BR       101004$         ; Отладка
          BR       101004$         ; Тестирование
; Запрос номера диска
101044$:JSR      R4,101616$      ; Позиционирование курсора
          .BYTE    4,34
          EMT      22              ; Ввод символа с ожиданием
          CMP      R0,#33          ; Код Escape ?
          BEQ      101142$         ; Да
          EMT      42              ; Вывод символа
          SUB      #60,R0          ; Проверка нажатия <0> - <3>
          BMI      100756$
          CMP      R0,#3
          BHI      100756$
          BR       101136$
; Запрос номера кассеты ПЗУ
101102$:JSR      R4,101616$      ; Позиционирование курсора
          .BYTE    5,34
          EMT      22              ; Ввод символа с ожиданием
          CMP      R0,#33          ; Код Escape ?
          BEQ      101142$         ; Да
          EMT      42              ; Вывод символа
          SUB      #61,R0          ; Проверка нажатия <1>,<2>
          BMI      100756$
          CMP      R0,#1
          BHI      100756$
101136$:MOV      R0,@SP           ; Сохранение номера устройства
          BR       101004$
101142$:EMT      22
          BR       100750$
101146$:MOV      #14,R0           ; Очистка экрана
          EMT      42
          CALL     100254$         ; Восстановить пользовательский режим
          MOV      R2,R0           ; Занести номер пункта в буфер клавиатуры
          CALL     104160$
          MOV      (SP)+,R0        ; Занести номер устройства в буфер клавиатуры
          CALL     104160$
          CLR      @#7072         ; Очистить признак вызова
          RETURN
; П/п вывода пунктов меню и запроса номера пункта
101202$:MOV      #-6,@#7150      ; Запрет вывода в верхнюю инф. строку
          MOV      R0,-(SP)
          CALL     101446$         ; Вывод пунктов меню
          ADD      (SP)+,PC
          BR       101432$

```

	BR	101224\$	
101224\$:	MOV	@R4, R0	; Позиционирование курсора
	MOV	2(R0), R3	; R3 = координаты
	DEC	R3	
	CALL	101564\$	
	MOV	@R4, R3	; R3 = адрес списка
101242\$:	EMT	22	; Ввод символа с ожиданием
	TST	R0	; Нажата УПР-@ ?
	BEQ	101436\$	
	CMP	R0, #33	; Символ Escape ?
	BEQ	101306\$	
	SUB	#60, R0	; Символы <0> - <9> ?
	BMI	101276\$	
	BEQ	101442\$; Символ <0> ?
	CMP	R0, @R3	; Не превышает номер последнего пункта ?
	BHI	101242\$	
	MOV	R0, R2	; R2 = номер выбранного пункта
	BR	101404\$	
101276\$:	ADD	#43, R0	; Нажата <Enter> (15(8)-60(8)+43(8)=0) ?
	BEQ	101442\$	
	BR	101404\$	
101306\$:	EMT	22	
	SUB	#176, R0	; Нажаты <ПОМ>, <ИСП> или стрелки
	BMI	101242\$	
	CMP	R0, #6	
	BHIS	101242\$	
	ASL	R0	
	ADD	R0, PC	
	BR	101344\$; <ПОМ>
	BR	101436\$; <ИСП>
	BR	101362\$; <Вверх>
	BR	101372\$; <Вниз>
	BR	101436\$; <Влево>
	BR	101436\$; <Вправо>
101344\$:	MOV	#20, R3	; Позиционирование курсора
	CALL	101564\$	
	EMT	44	; Вывод краткого хелпинга
	.WORD	103406\$; Адрес начала хелпинга
	BR	101224\$	
101362\$:	DEC	R2	; Уменьшить номер пункта
	BGT	101404\$	
	MOV	@R3, R2	; R2 = номер последнего пункта
	BR	101404\$	
101372\$:	INC	R2	; Увеличить номер пункта
	CMP	R2, @R3	; Превысил номер последнего пункта ?
	BLOS	101404\$	
	MOV	#1, R2	
101404\$:	MOV	R2, @4(R3)	; Запомнить номер текущего пункта
	DEC	@4(R3)	
	CLR	R0	
101416\$:	TST	(R4)+	
	CLR	@#7150	; Разрешение вывода в верхнюю инф. строку
	ASL	R2	
	TST	R0	
	RTS	R4	
101432\$:	MOV	#10, R0	
101436\$:	SUB	#5, R0	
101442\$:	DEC	R0	
	BR	101416\$	
; П/п вывода пунктов меню			
101446\$:	CLR	@#23164	; Запрет использования курсора
	MOV	@R4, R2	
	MOV	(R2)+, -(SP)	; Число пунктов меню

```

MOV      (R2)+, R3          ; Координаты курсора
ADD      R3, @SP
MOV      @(R2)+, -(SP)      ; Номер текущего пункта меню
ADD      R3, @SP
TST      R0
BNE      101502$
CALL     101564$            ; Курсор в начало экрана
EMT      44                 ; Стирание правой части экрана
.WORD    103364$
101502$: CALL 101564$
CMP      R3, @SP            ; Номер выводимого пункта совпадает с текущим ?
BNE      101516$
EMT      44                 ; Включить режим инверсии
.WORD    102124$
101516$: MOV (R2)+, R1      ; Вывод очередного пункта
EMT      46
CMP      R3, @SP            ; Номер выводимого пункта совпадает с текущим ?
BNE      101532$
EMT      44                 ; Выключить режим инверсии
.WORD    102127$
101532$: INC R3              ; Переход к очередному пункту
CMP      R3, 2(SP)          ; Вывод завершен ?
BLO      101502$            ; Нет
CMP      (SP)+, (SP)+
MOV      @R4, R0
MOV      @4(R0), R2          ; R2 = номер текущего пункта
INC      R2
MOV      #2, @#23164        ; Разрешить использование курсора
RETURN

```

; Следующие две п/п осуществляют позиционирование курсора

; В первой п/п координаты задаются в R3, во второй - в слове,

; которое следует за командой вызова п/п (JSR R4, 101616\$).

```

101564$: MOV    @#23150, R5    ; R5 = адрес таблицы видеопараметров
MOVVB     R3, R0              ; R0 = Y (строка)
CALL      115720$            ; Позиционирование по строке
SWAB      R3
MOVVB     R3, R0              ; R0 = X (столбец)
CALL      115672$            ; Позиционирование по столбцу
SWAB      R3
INC       @#7066              ; Установить признак перемещения курсора
RETURN
101616$: MOV    @#23150, R5    ; R5 = адрес таблицы видеопараметров
MOVVB     (R4)+, R0           ; R0 = Y (строка)
CALL      115720$            ; Позиционирование по строке
MOVVB     (R4)+, R0           ; R0 = X (столбец)
CALL      115672$            ; Позиционирование по столбцу
INC       @#7066              ; Установить признак перемещения курсора
RTS       R4

```

; СПИСКИ ПУНКТОВ МЕНЮ

; Структура списка:

; первое слово - число пунктов меню

; второе слово - координаты курсора

; третье слово - адрес ячейки, в которой сохраняется номер тек. пункта

; далее - адреса надписей пунктов меню

; Список главного меню УСТАНОВКА

```

101644$: .WORD    10
        .BYTE     4, 0
        .WORD     7152, 102165$, 102211$, 102235$, 102261$
        .WORD     102305$, 102331$, 102355$, 102401$

```

; Список меню СИСТЕМА КОМАНД

```

101672$: .WORD      2
          .BYTE      4, 30
          .WORD      7154, 102425$, 102445$

; Список меню ФОРМАТ ЭКРАНА
101704$: .WORD      4
          .BYTE      4, 30
          .WORD      7156, 102465$, 102477$, 102511$, 102523$

; Список меню ЦВЕТ СИМВОЛА
101722$: .WORD      10
          .BYTE      4, 30
          .WORD      7160, 102535$, 102562$, 102607$, 102634$
          .WORD      102661$, 102706$, 102733$, 102760$

; Список меню ЦВЕТ ЗНАКОМЕСТА
101750$: .WORD      10
          .BYTE      4, 30
          .WORD      7162, 102535$, 102562$, 102607$, 102634$
          .WORD      102661$, 102706$, 102733$, 102760$

; Список меню ЦВЕТ ЭКРАНА
101776$: .WORD      10
          .BYTE      4, 30
          .WORD      7164, 102535$, 102562$, 102607$, 102634$
          .WORD      102661$, 102706$, 102733$, 102760$

; Список меню ЦВЕТ КУРСОРА
102024$: .WORD      10
          .BYTE      4, 30
          .WORD      7166, 102535$, 102562$, 102607$, 102634$
          .WORD      102661$, 102706$, 102733$, 102760$

; Список меню РУЛОН
102052$: .WORD      3
          .BYTE      4, 30
          .WORD      7170, 103005$, 103024$, 103043$

; Список меню ТАЙМЕР
102066$: .WORD      2
          .BYTE      4, 30
          .WORD      7172, 103062$, 103077$

; Список меню ЗАГРУЗКА
102100$: .WORD      7
          .BYTE      4, 2
          .WORD      7174, 103175$, 103216$, 103237$, 103260$
          .WORD      103301$, 103322$, 103343$

; Esc-последовательность включения инверсии
102124$: .BYTE      33, 243, 0
; Esc-последовательность выключения инверсии
102127$: .BYTE      33, 277, 243, 0
; Заголовок меню УСТАНОВКА
102133$: .BYTE      14, 12, 40, 33, 244
          .ASCII     /УСТАНОВКА РЕЖИМОВ/
          .BYTE      33, 277, 244, 0

; Надписи меню УСТАНОВКА
102165$: .ASCII     /1 - система команд /
102211$: .ASCII     /2 - формат экрана /
102235$: .ASCII     /3 - цвет символа /
102261$: .ASCII     /4 - цвет знакоместа/
102305$: .ASCII     /5 - цвет экрана /

```

102331\$: .ASCIZ /6 - цвет курсора /
102355\$: .ASCIZ /7 - рулон /
102401\$: .ASCIZ /8 - таймер /

; Надписи меню СИСТЕМА КОМАНД

102425\$: .ASCIZ /1 - VT-52 /
102445\$: .ASCIZ /2 - 15ИЭ-00-013/

; Надписи меню ФОРМАТ ЭКРАНА

102465\$: .ASCIZ /1 - 80/<220>/24/
102477\$: .ASCIZ /2 - 40/<220>/24/
102511\$: .ASCIZ /3 - 20/<220>/24/
102523\$: .ASCIZ /4 - 10/<220>/24/

; Надписи меню ЦВЕТ СИМВОЛА, ЦВЕТ ЗНАКОМЕСТА, ЦВЕТ ЭКРАНА, ЦВЕТ КУРСОРА

102535\$: .BYTE '1,40,'-',40,33,240,'0,33,241,'0
.BYTE 237,237,237,237,33,277,240,33,277,241,0
102562\$: .BYTE '2,40,'-',40,33,240,'1,33,241,'1
.BYTE 237,237,237,237,33,277,240,33,277,241,0
102607\$: .BYTE '3,40,'-',40,33,240,'2,33,241,'2
.BYTE 237,237,237,237,33,277,240,33,277,241,0
102634\$: .BYTE '4,40,'-',40,33,240,'3,33,241,'3
.BYTE 237,237,237,237,33,277,240,33,277,241,0
102661\$: .BYTE '5,40,'-',40,33,240,'4,33,241,'4
.BYTE 237,237,237,237,33,277,240,33,277,241,0
102706\$: .BYTE '6,40,'-',40,33,240,'5,33,241,'5
.BYTE 237,237,237,237,33,277,240,33,277,241,0
102733\$: .BYTE '7,40,'-',40,33,240,'6,33,241,'6
.BYTE 237,237,237,237,33,277,240,33,277,241,0
102760\$: .BYTE '8,40,'-',40,33,240,'7,33,241,'7
.BYTE 237,237,237,237,33,277,240,33,277,241,0

; Надписи меню РУЛОН

103005\$: .ASCIZ /1 - плавный /
103024\$: .ASCIZ /2 - дискретный/
103043\$: .ASCIZ /3 - выключен /

; Надписи меню ТАЙМЕР

103062\$: .ASCIZ /1 - включен /
103077\$: .ASCIZ /2 - выключен/

; Заголовок меню ЗАГРУЗКА

103114\$: .ASCII <33><244>/ЗАГРУЗКА/<33><277><244>
.BYTE 12,12
.REPT 10
.BYTE 31
.ENDR
.ASCII /(0/<221>/3): 0/<12>
.REPT 10
.BYTE 32
.ENDR
.ASCIZ /(1,2): 1/

; Надписи меню ЗАГРУЗКА

103175\$: .ASCIZ /1 - диск /
103216\$: .ASCIZ /2 - кассета ПЗУ /
103237\$: .ASCIZ /3 - сеть /
103260\$: .ASCIZ /4 - стык С2 /
103301\$: .ASCIZ /5 - магнитофон /
103322\$: .ASCIZ /6 - отладка /
103343\$: .ASCIZ /7 - тестирование/

; Стирание правой части экрана

103364\$: .REPT 10


```
.BYTE 37,35
.ENDR
.BYTE 37,0
```

; Краткий хелпинг

```
103406$: .BYTE 61,221,71,54,200,54,201
        .ASCII / - выбор элемента/<15><12>
        .BYTE 40,40,203
        .ASCII / - следующий уровень/<15><12>
        .BYTE 40,40,202
        .ASCII / - предыдущий уровень/<15><12>
        .ASCIZ /0,/<224>/,ИСП - выход/
        .EVEN
```

```
; *****
; *
; *
; *          ДРАЙВЕР КЛАВИАТУРЫ          *
; *
; *          Адреса 103554 - 107743        *
; *
; *****
```

; П/п начальной инициализации

```
103554$: JSR      R5,173730$      ; Очистка ОЗУ
        .WORD    7176,1762      ; Начальный адрес - 7176, число слов - 1762
103564$: TST      @#177702      ; Сбросить готовность рег.сост. клавиатуры
        MOV      #175412$,@#300 ; Вектор прерываний от клавиатуры
        MOV      #200,@#302
        MOV      #100,@#177700 ; Разрешить прерывания от клавиатуры
        MOV      #175540$,@#324 ; Вектор прерываний от канала 0 (источник)
        MOV      #200,@#326
        BIS      #1,@#177076    ; Разрешить прерывания от канала 0 (источник)
        CALL     104064$        ; Настройка начальных адресов буферов
        MOV      #11140,@#13140 ; Нач. адрес буфера клавиш K1 - K25
        MOV      #7234,@#7214   ; Текущая таблица - табл. основного регистра
        MOV      #7234,@#7216
        MOV      #40,@#7221     ; Заглавные буквы
        MOV      #16,@#7222     ; Латинский регистр
        MOV      #17,@#7223
```

; Заполнение таблиц раскладки клавиатуры

```
        MOV      #105520$,R1    ; R1 = нач. адрес данных для таблиц раскладки
        MOV      #7234,R0      ; R0 = нач. адрес таблиц раскладки
        MOV      #5,R4         ; R4 = число таблиц раскладки
103720$: MOV      #130,R2       ; R2 = число элементов таблицы
        MOV      R0,R3         ; R3 = адрес текущей заполняемой таблицы
103726$: MOVB     (R1)+,(R3)+   ; Пересылка
        CLRB     (R3)+
        SOB      R2,103726$    ; R2 = номер первой регистровой клавиши
103740$: MOVB     (R1)+,R3
        BEQ      103762$      ; Это строка или конец данных для тек. табл.?
        BMI      103756$      ; Переход к следующей регистровой комбинации ?
        DEC      R3           ; Регистровая клавиша (в R3 - смещение)
        ASL      R3
        ADD      R0,R3
        MOV      R2,@R3
103756$: INC      R2
        BR       103740$
103762$: MOVB     (R1)+,R3
        BEQ      104004$      ; Это конец таблицы ?
        DEC      R3           ; Строка (адрес в R1, смещение в табл. в R3)
        ASL      R3
```

```

        ADD      R0,R3
        MOV      R1,@R3
103776$:TSTB    (R1)+          ; Прогон строки
        BNE      103776$
        BR       103762$
104004$:ADD     #260,R0        ; Переход к следующей таблице
        SOB      R4,103720$
        CALL     113670$      ; Установить систему команд VT-52
        MOVB     #2,@#7226    ; Интервал автоповтора
        RETURN
; П/п ПАУЗА (ожидание нажатия любой клавиши)
104026$:CLR     @#177700      ; Запретить прерывания от клавиатуры
104032$:TSTB    @#177700      ; Ожидание нажатия или отжатия клавиши
        BPL      104032$
        TSTB     @#177702      ; Клавиша отжата ?
        BMI      104032$      ; Да
        MOV      #100,@#177700 ; Разрешить прерывания от клавиатуры
104054$:JSR     R5,173730$    ; Очистка буферов клавиатуры
        .WORD    2240,14
; Установка указателей на начало буферов
104064$:MOV     #2260,@#7176
        MOV      #2260,@#7200
        MOV      #2240,@#7202
        MOV      #2240,@#7204
        MOV      #2200,@#7206
        MOV      #2200,@#7210
        RETURN
; П/п обслуживания запросов ЕМТ 22 и ЕМТ 24 (ввод символа с ожиданием)
104132$:TRAP    2             ; Считывание символа с буфера в регистр R0
        ADD      @#7230,PC     ; Символ есть в буфере ?
        BR       104152$      ; Да
        CLR      @#7230
        TRAP     0             ; Запуск диспетчера процессов
        BR       104132$
104152$:MOV     R0,2(SP)
        RETURN
; П/п заноса ASCII-символов и адресов строк в буфер
104160$:MOV     R0,@7210      ; Занести символ в буфер
        ADD      #2,@#7210      ; Продвинуть указатель
        BIC      #40,@#7210    ; Корректировка указателя (циклический буфер)
        ADD      @#7230,PC
        RETURN
104206$:TRAP    2
        RETURN
; *** ОСНОВНАЯ П/п ОБСЛУЖИВАНИЯ КЛАВИАТУРЫ ***
; Точка выхода из п/п обслуживания клавиатуры
104212$:CLR     @#7060        ; Очистить флаг вызова п/п
        CLRB     @#7227        ; Очистить ячейку значения автоповтора
104222$:ADD     @#7230,PC
        RETURN
104230$:TRAP    2
        RETURN
104234$:TSTB    @7202          ; Есть скэн-коды в буфере автоповтора ?
        BEQ      104212$      ; Нет
        TSTB     @#7227        ; Истекло время автоповтора ?
        BNE      104222$      ; Нет
        MOV      @#7202,R3     ; R3 = указатель на буфер автоповтора(чтение)
104254$:MOVB    (R3)+,R0      ; Прочитать очередной скэн-код
        BNE      104336$      ; Он существует
104260$:BIC     #20,R3        ; Подправить значение указателя
        CMP      R3,@#7204      ; Буфер просмотрен до конца ?
        BNE      104254$      ; Нет

```

```

MOV      @#7226, @#7227
DEC      @#7060
BR       104222$
; Точка входа в п/п обслуживания клавиатуры из диспетчера процессов
104306$: MOV      @#7176, R1      ; R1 = указатель на буфер скэн-кодов
MOV      @R1, R0      ; Прочесть скэн-код
BEQ      104234$      ; Скэн-кода нет
DEC      @#7060      ; Уменьшить счетчик (флаг вызова)
CLRB     (R1)+      ; Вычеркнуть скэн-код из буфера
BIC      #10, R1      ; Подправить значение указателя
MOV      R1, @#7176
CLR      R3      ; В данном случае R3 является индикатором
; того, что скэн-код был прочитан из основ-
; ного буфера (R3=0) или из буфера авто-
; повтора (R3<>0).
104336$: MOV      R0, R2      ; R2 = скэн-код
BMI      104516$      ; Это скэн-код отжатия
ASHC     #74, R0      ; Извлечение ASCII-кода
MOVB     105510$(R0), R1
MOV      R2, R0
SUB      R1, R0
ASL      R0
ADD      @#7214, R0
MOV      @R0, R0
MOV      R0, R1
SWAB     R1
BEQ      104410$      ; Это одиночный ASCII-код
ASRB     R1
BNE      104434$      ; Это адрес ASCIIZ-строки
MOVB     R0, R0      ; Переход на обслуживание регистровых клавиш
ASL      R0
JMP      @104714$(R0)
104410$: TST      @#7232      ; Разрешена передача одиночных символов ?
BNE      104510$      ; Нет
TSTB     R0      ; Символ в диапазоне 128-255 ?
BMI      104434$      ; Да
CMP      R0, #100      ; Это символы @, A-Z, [, \, ], _ ?
BLO      104434$      ; Нет
BISB     @#7220, R0      ; Установка заглавные/строчные буквы
104434$: TST      R3      ; Это символ из буфера автоповтора ?
BNE      104472$      ; Да
TSTB     @7202      ; Буфер автоповтора пуст ?
BNE      104454$      ; Нет
MOVB     #31, @#7227      ; Задать время между нажатием и автоповтором
104454$: MOVB     R2, @7204      ; Положить скэн-код в буфер автоповтора
INC      @#7204      ; Подправить значение указателя
BIC      #20, @#7204
104472$: MOV      @#7210, R1      ; R1 = указатель на буфер ASCII-символов
MOV      R0, (R1)+      ; Положить символ(адрес строки) в буфер
BIC      #40, R1
MOV      R1, @#7210
104510$: TST      R3      ; Это был символ из буфера автоповтора ?
BNE      104260$      ; Да
104514$: BR       104306$
104516$: MOV      @#7212, R0      ; R0 = ожидаемый код отжатия рег. клавиши
BEQ      104614$      ; Регистровая клавиша не была нажата
CMPB     R2, R0      ; Коды отжатия совпадают ?
BEQ      104540$      ; Да
104530$: SWAB     R0
BEQ      104530$
CMPB     R2, R0
BNE      104614$
; Обработка кода отжатия регистровой клавиши
104540$: CLRB     R0      ; Очистить байт кода отжатия

```

```

SWAB      R0
MOV       R0, @7212
CLR       R0
CMPB      R2, #205      ; Была отжата клавиша HP (Shift) ?
BEQ       104574$      ; Да
MOVB      @7224, R0      ; R0 = ASCII-код, посылаемый при отжати АЛФ
BEQ       104600$      ; Его нет
CLRB      @7224
BR        104600$
104574$: SWAB      @7220      ; Переменить состояние заглавные/строчные
104600$: MOV      @7216, @7214 ; Восстановить пред. табл. раскл. клавиатуры
TST       R0            ; Есть данные для передачи при отжати АЛФ ?
BNE       104472$      ; Да
BR        104306$

; Обработка кода отжатия алфавитно-цифровой клавиши
104614$: BIC      #177760, R2 ; Оставить биты кода отжатия
MOV       @7202, R1      ; R1 = указатель на буфер автоповтора
104624$: MOVB     (R1)+, R0 ; Прочитать скэн-код из буфера автоповтора
BEQ       104644$      ; Он отсутствует
BIC       #177760, R0    ; Оставить биты кода отжатия
CMP       R0, R2        ; Эти коды совпадают ?
BEQ       104700$      ; Да
MOV       R2, R3
BEQ       104306$
104644$: BIC      #20, R1
TST       R3
BNE       104660$
MOV       R1, @7202
104660$: CMP      R1, @7204 ; Буфер просмотрен до конца ?
BNE       104624$      ; Нет
MOVB      #2, @7226      ; Установить нормальную скорость автоповтора
104674$: JMP      104306$
104700$: CLRB     -1(R1)   ; Выбросить скэн-код из буфера автоповтора
CLR       R2            ; R2 = 0 (признак того, что скэн-код обнаружен)
TST       R3
BNE       104674$
BR        104644$

; Адреса п/п обслуживания регистровых комбинаций
104714$: .WORD    104766$   ; Нажатие HP (Shift)
        .WORD    105044$   ; Нажатие АЛФ
        .WORD    105116$   ; Нажатие ГРАФ (Graph)
        .WORD    105150$   ; Нажатие УПР (Ctrl)
        .WORD    105160$   ; Нажатие ФИКС (Lock)
        .WORD    104734$   ; Установка режимов (УСТ)
        .WORD    105234$   ; Режим ИСУ (УПР+ПОМ)
        .WORD    104746$   ; Печать экрана (УПР+ИСП)

; П/п обслуживания нажатия УСТ
104734$: CLR      @7042     ; Адресовать ASCII-коды клавиатуры запросу ЕМТ
INC       @7070           ; Установить флаг вызова меню УСТАНОВКА
BR        104514$

; П/п обслуживания нажатия УПР+ИСП (печать экрана)
104746$: CALL     135704$   ; Печать экрана
MOV       @7216, @7214
CLR       @7212
BR        104514$

; П/п обслуживания нажатия HP
104766$: TSTB     @7202     ; Нажата какая-нибудь алф.-циф. клавиша ?
BEQ       105004$      ; Нет
MOV       #401, 7226      ; Установить повышенную скорость автоповтора
BR        104514$
105004$: MOV      #205, R1   ; R1 = код отжатия HP
MOV       #7514, @7214     ; Текущая табл.-таблица нижнего регистра
105016$: SWAB     @7220     ; Переключить режим заглавные/строчные буквы
105022$: SWAB     @7212

```

```

        MOVB     R1, @7212
        CMPB     @7213, #207      ; Была нажата клавиша ФИКС ?
        BEQ      105202$          ; Да
        BR       104514$

; П/п обслуживания нажатия АЛФ
105044$: MOVB     @7222, R0        ; R0 = 16(8) или 17(8)
        MOVB     @7223, @7224
        SWAB     @7212
        MOVB     #206, @7212      ; Код отжатия клавиши АЛФ
        CMPB     @7213, #207      ; Была нажата клавиша ФИКС ?
        BNE      105112$          ; Нет
        CLRB     @7224            ; При отжатии АЛФ не передавать ASCII-кода
        MOV      @7216, @7214
105112$: JMP     104472$

; П/п обслуживания нажатия ГРАФ
105116$: MOV      #7774, @7214    ; Текущая табл.-таблица ГРАФ-регистра
105124$: MOV      #206, R1        ; R1 = код отжатия ГРАФ и УПР
105130$: CMP      @7214, @7216    ; Адрес текущей табл. равен адресу предыдущей ?
        BNE      105022$          ; Нет
        MOV      #7234, @7214    ; Текущая таблица-таблица основного регистра
        BR       105022$

; П/п обслуживания нажатия УПР
105150$: MOV      #10254, @7214   ; Текущая таблица-таблица УПР-регистра
        BR       105124$

; П/п обслуживания нажатия ФИКС
105160$: MOV      #207, R1        ; R1 = код отжатия ФИКС
        TST      @7212            ; Нажата регистровая клавиша ?
        BNE      105202$          ; Да
        MOV      #10534, @7214   ; Текущая таблица-таблица ФИКС-регистра
        BR       105130$
105202$: CMP      @7214, #7514    ; Нажата НР ?
        BEQ      105016$          ; Да
        TSTB     @7224            ; Нажата АЛФ ?
        BEQ      105224$          ; Нет
        CLRB     @7224            ; При отжатии АЛФ не передавать ASCII-кода
105224$: MOV      @7214, @7216
        BR       105022$

; П/п обслуживания нажатия УПР+ПОМ (ИСУ on/off)
105234$: COMB     @722552         ; Изменить режим ИСУ/не ИСУ
        BEQ      105250$          ; Режим ИСУ выключен
        EMT      52               ; Высветить надпись "ИСУ"
        .WORD    107641$
        BR       105254$
105250$: EMT      52               ; Стереть надпись "ИСУ"
        .WORD    107632$
105254$: JMP     104306$

; П/п обработки управляющего символа Ctrl+N (переключение на русский алфавит)
105260$: MOV      @722556, R0     ; R0 = текущее состояние алфавита
        MOV      #200, @722556   ; Установить русский алфавит
        MOV      #16*400+17, @7222 ; Последовательность посылки кодов
        ; при нажатии и отжатии АЛФ
        EMT      52               ; Высветить надпись "РУС"
        .WORD    107731$
        BR       105330$

; П/п обработки управляющего символа Ctrl+O (переключение на латинский алфавит)
105306$: MOV      @722556, R0
        CLR      @722556         ; Установить латинский алфавит
        MOV      #17*400+16, @7222
        EMT      52               ; Высветить надпись "ЛАТ"
        .WORD    107736$
105330$: CMP      R0, @722556     ; Предыдущий и текущий режимы алфавита равны ?
        BEQ      105342$          ; Да
        SWAB     @7220            ; Переменить верхний/нижний регистр
105342$: RETURN

```

```

105344$:RETURN
; П/п устанавливает специальные режимы клавиатуры (системный, ДКЛ)
105346$:MOV      (R5)+,R0      ; R0 = адрес новых кодов клавиатуры
      MOV      (R5)+,R1      ; R1 = адрес буфера сохранения
      TST      @R1           ; Режим уже был установлен ?
      BNE      105376$       ; Да
      INC      (R1)+         ; Установить флаг спецрежима
      BR       105372$
105362$:MOV      7232(R2), (R1)+ ; Сохранить значение клавиши
      MOV      (R0)+, 7232(R2) ; Занести новое значение
105372$:MOV      (R0)+, R2      ; R2 = адрес клавиши
      BNE      105362$       ; Еще не конец таблицы
105376$:RTS      R5
; П/п восстанавливает предыдущий режим клавиатуры
105400$:MOV      (R5)+,R0      ; R0 = адрес новых кодов клавиатуры
      MOV      (R5)+,R1      ; R1 = адрес буфера сохранения
      TST      @R1           ; Спецрежим установлен ?
      BEQ      105426$       ; Нет
      CLR      (R1)+         ; Сбросить флаг спецрежима
      BR       105422$
105414$:MOV      (R1)+, 7232(R2) ; Восстановить значение клавиши
      TST      (R0)+
105422$:MOV      (R0)+, R2      ; R2 = адрес клавиши
      BNE      105414$       ; Еще не конец таблицы
105426$:RTS      R5
; П/п переключения клавиатуры в новый режим. Если текущий режим является
; не системным, то переключение производится сразу, иначе (системный режим)
; производится проверка адресов клавиш и для клавиш, задействованных
; в системном режиме, информация сохраняется в буфере, откуда перекачивается
; в таблицу раскладки при выходе из системного режима.
105430$:MOV      (R5)+,R0      ; R0 = адрес новой таблицы
      BR       105502$
105434$:TST      @#11054       ; Установлен системный режим ?
      BEQ      105476$       ; Нет
      MOV      #107310$-4,R2   ; R2 = адрес списка системных клавиш
105446$:ADD      #4,R2         ; Перейти к следующему адресу
      MOV      @R2,R3         ; R3 = адрес клавиши
      BEQ      105476$       ; Конец списка системных клавиш
      CMP      R3,R1          ; Адреса совпадают ?
      BNE      105446$       ; Нет
      SUB      #107310$,R2     ; Получение адреса клавиши в буфере сохранения
      ASR      R2
      MOV      (R0)+,11056(R2) ; Занести код в буфер сохранения
      BR       105502$
105476$:MOV      (R0)+, 7232(R1) ; Занести код в таблицу раскладки клавиатуры
105502$:MOV      (R0)+,R1      ; R1 = адрес клавиши
      BNE      105434$       ; Еще не конец таблицы
      RTS      R5
; Корректирующие коды, использующиеся при вычислении адреса в таблице раскладки
105510$: .BYTE    5,12,17,24,31,36,43,50

; НАЧАЛЬНЫЕ ДАННЫЕ ДЛЯ ТАБЛИЦ РАСКЛАДКИ КЛАВИАТУРЫ
; Структура :
; первые 88 байт - ASCII-коды для клавиш
; далее - смещения для регистровых клавиш (конец - байт 0)
; строки символов : - смещение в таблице (байт 0 - конец списка)
;                  - ASCIIZ-строка
; байт 0 - конец описания раскладки

; Раскладка основного регистра
105520$: .BYTE    54,33,73,0,0,64,0,0,67,70,55,11,112,61,62,63,105,65,66
      .BYTE    133,135,0,0,106,103,125,113,120,116,107,114,104,0,0,121
      .BYTE    131,127,101,111,122,117,102,100,0,0,0,136,123,115,40,124
      .BYTE    130,0,54,67,60,61,64,53,177,0,0,56,134,126,70,56,62,65,0

```

```

        .BYTE    0,15,0,137,110,132,71,15,63,66,14,12,57,72,55,60,71
; - список регистровых клавиш
        .BYTE    55,56,43,30,57,110,0
; - строки символов
        .BYTE    107
        .ASCIZ   <33><177>
        .BYTE    123
        .ASCIZ   <33>/~/
        .BYTE    4,0,5,0,6,0,10,0,11,0,0

; Раскладка HP-регистра
105702$: .BYTE    54,33,53,0,0,0,44,0,0,47,50,57,11,112,41,42,43,105,45,46
        .BYTE    133,135,0,0,106,103,125,113,120,116,107,114,104,0,0,121
        .BYTE    131,127,101,111,122,117,102,100,0,0,0,136,123,115,40,124
        .BYTE    130,0,74,67,60,61,64,52,177,0,0,76,134,126,70,56,62,65,0
        .BYTE    0,12,0,137,110,132,71,15,63,66,14,12,77,52,75,60,51
; - список регистровых клавиш
        .BYTE    55,56,43,30,57,110,0
; - строки символов
        .BYTE    107
        .ASCIZ   <33><177>
        .BYTE    123
        .ASCIZ   <33>/~/
        .BYTE    4,0,5,0,6,0,10,0,11,0,0

; Раскладка ГРАФ-регистра
106064$: .BYTE    54,33,73,0,0,0,64,0,0,67,70,55,11,252,61,62,63,245,65,66
        .BYTE    273,275,0,0,246,243,265,253,260,256,247,254,244,0,0,261
        .BYTE    271,267,241,251,262,257,242,240,0,0,0,276,263,255,40,264
        .BYTE    270,0,54,67,60,61,64,53,177,0,0,56,274,266,70,56,62,65,0
        .BYTE    0,0,0,277,250,272,71,15,63,66,14,12,57,72,55,60,71
; - список регистровых клавиш
        .BYTE    55,56,43,30,57,110,0
; - строки символов
        .BYTE    66
        .ASCIZ   <33>/D/
        .BYTE    76
        .ASCIZ   <33>/C/
        .BYTE    112
        .ASCIZ   <33>/A/
        .BYTE    77
        .ASCIZ   <33>/B/
        .BYTE    111
        .ASCIZ   <15><12>
        .BYTE    4,0,5,0,6,0,10,0,11,0,0

; Раскладка УПР-регистра
106262$: .BYTE    0,33,73,0,0,0,64,0,0,67,70,0,11,12,61,62,63,5,65,66,33,35
        .BYTE    0,0,6,3,25,13,20,16,7,14,4,0,0,21,31,27,1,11,22,17,2,0,0,0,0
        .BYTE    36,23,15,40,24,30,0,54,67,60,61,64,0,177,0,0,56,34,26,70,56
        .BYTE    62,65,0,0,0,0,37,10,32,71,15,63,66,14,12,57,72,55,60,71
; - список регистровых клавиш
        .BYTE    55,56,43,30,57,110,123,107,0
; - строки символов
        .BYTE    74
        .ASCIZ   <33>/P/
        .BYTE    14
        .ASCIZ   <33>/Q/
        .BYTE    1
        .ASCIZ   <33>/R/
        .BYTE    4,0,5,0,6,0,10,0,11,0,0

; Раскладка ФИКС-регистра
106452$: .BYTE    54,33,73,0,0,0,64,0,0,67,70,55,11,112,61,62,63,105,65,66

```

```

        .BYTE    133,135,0,0,106,103,125,113,120,116,107,114,104,0,0,121
        .BYTE    131,127,101,111,122,117,102,100,0,0,0,136,123,115,40,124
        .BYTE    130,0,54,67,60,61,64,53,177,0,0,56,134,126,70,56,62,65,0
        .BYTE    0,0,0,137,110,132,71,15,63,66,14,12,57,72,55,60,71
; - список регистровых клавиш
        .BYTE    55,56,43,30,57,110,0
; - строки символов
        .BYTE    107
        .ASCIZ   <33><177>
        .BYTE    123
        .ASCIZ   <33>/~/
        .BYTE    4,0,5,0,6,0,10,0,11,0,0

; Данные для пользовательского режима (VT-52)
106634$: .WORD    224,106752$,176,106755$,154,106760$,174,106763$
        .WORD    504,106752$,456,106755$,434,106760$,454,106763$
        .WORD    1244,106752$,1216,106771$,1174,106760$,1214,106766$
        .WORD    1524,106752$,1476,106755$,1454,106760$,1474,106763$
        .WORD    244,106774$,524,106771$,1242,106777$,0
106752$: .ASCIZ   <33>/A/
106755$: .ASCIZ   <33>/B/
106760$: .ASCIZ   <33>/D/
106763$: .ASCIZ   <33>/C/
106766$: .ASCIZ   <33>/K/
106771$: .ASCIZ   <33>/J/
106774$: .ASCIZ   <33>/H/
106777$: .ASCIZ   <15><12>
; Данные для пользовательского режима (15ИЭ-00-013)
107002$: .WORD    224,34,176,35,154,32,174,31,504,22,456,26,434,24,454,23
        .WORD    1244,34,1216,37,1174,32,1214,13,1524,34,1476,35,1454,32,1474,31
        .WORD    244,10,524,14,1242,25,0
; Данные для ДКЛ-режима
107120$: .WORD    236,107216$,210,107222$,162,107226$,164,107232$
        .WORD    212,107236$,240,107242$,166,107246$,214,107252$
        .WORD    242,107256$,160,107262$,206,107266$,234,107272$
        .WORD    170,107276$,30,107301$,2,107304$,0
107216$: .ASCIZ   <33>/?M/
107222$: .ASCIZ   <33>/?n/
107226$: .ASCIZ   <33>/?p/
107232$: .ASCIZ   <33>/?q/
107236$: .ASCIZ   <33>/?r/
107242$: .ASCIZ   <33>/?s/
107246$: .ASCIZ   <33>/?t/
107252$: .ASCIZ   <33>/?u/
107256$: .ASCIZ   <33>/?v/
107262$: .ASCIZ   <33>/?w/
107266$: .ASCIZ   <33>/?x/
107272$: .ASCIZ   <33>/?y/
107276$: .ASCIZ   <33>/P/
107301$: .ASCIZ   <33>/Q/
107304$: .ASCIZ   <33>/R/
        .EVEN
; Данные для системного режима
107310$: .WORD    162,60,164,61,212,62,240,63,166,64
        .WORD    214,65,242,66,160,67,206,70,234,71
        .WORD    222,107462$,236,107464$,246,107466$,216,107471$
        .WORD    224,107474$,176,107477$,154,107502$,174,107505$
        .WORD    504,107510$,456,107513$,434,107516$,454,107521$
        .WORD    146,107460$,172,107456$,244,14,0
107456$: .BYTE    177,0
107460$: .BYTE    40,0
107462$: .BYTE    15,0
107464$: .BYTE    15,0
107466$: .ASCIZ   <33>/~/

```



```

107471$: .ASCIZ <33><177>
107474$: .ASCIZ <33><200>
107477$: .ASCIZ <33><201>
107502$: .ASCIZ <33><202>
107505$: .ASCIZ <33><203>
107510$: .ASCIZ <33><204>
107513$: .ASCIZ <33><205>
107516$: .ASCIZ <33><206>
107521$: .ASCIZ <33><207>
; Надписи для верхней информационной строки
107524$: .ASCIZ <0>/ /
107533$: .ASCIZ <0>/ УСТ/
107542$: .ASCIZ <5>/ /
107551$: .ASCIZ <5>/ ПМ /
107560$: .ASCIZ <5>/ ЦМ /
107567$: .ASCIZ <12>/ /
107576$: .ASCIZ <12>/ ДКЛ /
107605$: .ASCIZ <17>/ /
107614$: .ASCIZ <17>/ ИНВ /
107623$: .ASCIZ <17>/ПОДЧ /
107632$: .ASCIZ <24>/ /
107641$: .ASCIZ <24>/ ИСУ /
107650$: .ASCIZ <31>/ /
107657$: .ASCIZ <31>/ НР /
107666$: .ASCIZ <31>/ СУ /
107675$: .ASCIZ <31>/ГРАФ /
107704$: .ASCIZ <31>/ФИКС /
107713$: .ASCIZ <36>/ЗАГЛ /
107722$: .ASCIZ <36>/СТР /
107731$: .ASCIZ <43>/РУС/
107736$: .ASCIZ <43>/ЛАТ/
.EVEN

```

```

; *****
; *
; *
; * ДРАЙВЕР ТЕКСТОВОГО ТЕРМИНАЛА *
; *
; *
; * Адреса 107744 - 125027 *
; *
; *****

```

```

; П/п начальной инициализации
107744$: JSR R5,173730$ ; Очистка ОЗУ
.WORD 13142,4017 ; Начальный адрес - 13142, число слов - 4017
107754$: MOV #0,@#177026 ; Разрешить все планы видеопамати
CLR @#177020 ; Очистка регистров цвета кода фона
CLR @#177022
CLR R0 ; Цикл очистки видеопамати
MOV #177010,R1
MOV #100000,@R1
MOV #100000,R2
MOV #177024,R3
110014$: MOV R0,@R3
INC @R1
SOB R2,110014$
MOV #175700$,@#320 ; Адрес п/п обработки прерывания канала 0, пр.
MOV #200,@#322
BIS #1,@#177066 ; Разрешить прерывания канала 0, приемник
CALL 110266$ ; Запрограммировать таблицу видеострок
MOV #117430$-13,R0 ; Настройка таблицы адресов рисунков символов
MOV #14142,R1 ; на область ПЗУ (адреса 117430-125027)
MOV #400,R2

```

```

MOV      #13, R3
110070$: ADD    R3, R0
MOV      R0, (R1)+
SOB      R2, 110070$
MOV      #112462$, R0      ; Настройка таблицы адресов п/п управляющих и
BR        110110$          ; Esc-последовательностей
110104$: MOV    (R0)+, 13142(R1)
110110$: MOV    (R0)+, R1
BNE      110104$
MOV      #2000, @#22542    ; Настройка указателей на буферы
MOV      #2000, @#22544
MOVB     #14, @#2000      ; Положить символ Ctrl+L в буфер (CLS)
JSR      R5, 110656$      ; Загрузка в ОЗУ ТВ для верхней строки
.WORD    112040$, 22560
JSR      R5, 110656$      ; Загрузка в ОЗУ ТВ для экрана пользователя
.WORD    112142$, 22656
JSR      R5, 110656$      ; Загрузка в ОЗУ ТВ для экрана установки
.WORD    112252$, 22754
JSR      R5, 110656$      ; Загрузка в ОЗУ ТВ для нижней строки
.WORD    112360$, 23052
MOV      #2270, @#272      ; Настройка первой видеостроки
110204$: MOV    #22656, R5   ; R5 = адрес ТВ для экрана пользователя
110210$: MOV    62(R5), @#23160 ; Указатель на регистр управления отображением
                                ; экраном пользователя
MOV      72(R5), @#23164    ; Режим использования курсора
INC      @#7066             ; Увел. счетчик симв. в буфере (для вывода CLS)
110230$: MOV    R5, @#23150 ; Сделать текущей ТВ, адрес которой в R5
MOV      34(R5), @#177016   ; Настройка регистра кода цвета точки
MOV      40(R5), @#177020   ; Настройка регистра кода цвета фона
MOV      40(R5), @#177022
MOVB     70(R5), @64(R5)    ; Установить цвет курсора
RETURN

; П/п программирования таблицы видеострок
110266$: MOV    #1131, R2    ; Число программируемых видеострок (BC)
MOV      #7034, R1          ; Адрес конца
MOV      R1, @#7036
110302$: MOV    R1, R0      ; Цикл начального заполнения таблицы
MOV      R0, -(R1)
TST      -(R1)
SOB      R2, 110302$
JSR      R5, 110550$        ; Заполнение зоны адресов для BC верхней строки
.WORD    175700, 2410, 12, 50
MOV      R0, @#2464
MOV      #177460, @#2474
JSR      R5, 110550$        ; Заполнение зоны адресов для BC польз. экрана
.WORD    100000, 2500, 436, 120
MOV      #177460, @#4674
JSR      R5, 110550$        ; Заполнение зоны адресов для BC УСТ экрана
.WORD    177460, 4704, 25, 0
JSR      R5, 110550$
.WORD    154540, 5030, 334, 50
JSR      R5, 110550$
.WORD    177460, 6610, 26, 0
MOV      #177460, @#6744
JSR      R5, 110550$        ; Заполнение зоны адресов для BC нижней строки
.WORD    176570, 6754, 13, 50
JSR      R5, 110550$
.WORD    177460, 7030, 2, 0
MOV      #2, R0             ; Выделение регистров управления отображением,
MOV      #6, R1             ; цветом и подправка структуры таблицы
BIS      R0, @#2366
BIS      R1, @#2376
BIS      R0, @#2456
BIS      R1, @#2466

```

```

MOV      #2500, @#4666
BIS      R1, @#4676
MOV      #6742, @#4536
MOV      #6742, @#6736
BIS      R1, @#6746
RETURN

; П/п заполнения зоны адресов для ВС
; 1 параметр - начальный видеоадрес
; 2 параметр - начальный адрес таблицы ВС
; 3 параметр - размер таблицы (число видеострок)
; 4 параметр - длина видеостроки
110550$:MOV      (R5)+, R0
MOV      (R5)+, R1
MOV      (R5)+, R2
MOV      (R5)+, R3
110560$:MOV      R0, (R1)+
ADD      R3, R0
TST      (R1)+
SOB      R2, 110560$
RTS      R5

; П/п установки таблицы ВС на начало видеопамати
110572$:ADD      74(R5), PC      ; Рулон присутствует ?
BR       110654$              ; Нет
CLR      @#23170              ; Число строк для скроллинга = 0
MOV      @#2476, R0           ; Корректировка таблицы
ADD      #2040, R0
CMP      R0, #4670
BLOS     110626$
SUB      #2170, R0
110626$:MOV      R0, -2(R0)
MOV      #6742, @#4536      ; Установка начальных значений
MOV      #2500, @#2476
MOV      #2500, 30(R5)
110654$:RETURN

; П/п формирования таблицы видеопараметров
110656$:MOV      (R5)+, R0      ; R0 = адрес таблицы в ПЗУ
MOV      @R5, R1              ; R1 = адрес таблицы в ОЗУ
JSR      R5, 110716$          ; Перекачка таблицы в ОЗУ
MOV      (R5)+, R1
MOV      (R0)+, @64(R1)      ; Заполнение регистра управления отображением
MOV      (R0)+, @54(R1)
MOV      (R0)+, @50(R1)      ; Заполнение регистра управления цветом
MOV      (R0)+, @52(R1)
RTS      R5

; П/п перекачки
110712$:MOV      (R5)+, R0
MOV      (R5)+, R1
110716$:MOV      (R0)+, R2      ; R2 = число слов
110720$:MOV      (R0)+, (R1)+
SOB      R2, 110720$
RTS      R5

; *** ОСНОВНАЯ ПОДПРОГРАММА УПРАВЛЕНИЯ ТЕКСТОВЫМ ТЕРМИНАЛОМ ***
110726$:MOV      #174164$, -(SP) ; Занести в стек адреса возвратов
MOV      #111126$, -(SP)
MOV      SP, @#23152
MOV      @#23154, SP
MOV      @#23150, R5          ; R5 = адрес текущей ТВ
INC      @#7066              ; Установить флаг вызова п/п перекл. курсора
110756$:INC      @#7064

; П/п чтения очередного символа, посланного по каналу 0
110762$:DEC      @#7064      ; Уменьшить счетчик символов
BEQ      111050$            ; Символов нет
MOVB     @22542, R0          ; Прочитать символ
INC      @#22542            ; Подкорректировать указатель

```

```

        BIC      #200, @#22542
        BIC      #177400, R0
        CMP      R0, #40          ; Это управляющие коды ?
        BHIS     111024$          ; Нет
        ADD      @#23174, PC      ; Разрешено чтение управляющих символов ?
111024$: RETURN          ; Да
111026$: CMP     R0, #33          ; Это символ Esc ?
        BEQ      111044$          ; Да
        ASL      R0              ; Вызов п/п обработки управляющего символа
        CALL     @#13142(R0)
        BR       110762$
111044$: JMP     113544$
111050$: ADD     @#22546, PC      ; Была попытка передачи символа ?
        BR       111062$          ; Нет
        TRAP     4
        BR       110756$
111062$: MOV     @#7056, R0       ; R0 = указатель текущей задачи
        MOV      #110726$, 22(R0) ; Запомнить адрес возврата
        MOV      SP, @#23154
        MOV      @#23152, SP
        CMP      (SP)+, (SP)+
        JMP      174164$          ; Переход на цикл вызова подпрограмм
111112$: ADD     @#22546, PC      ; Была попытка передачи символа ?
        RETURN          ; Нет
        TRAP     4
        INC      @#7064
111126$: DEC     @#7064          ; Уменьшить счетчик символов
        BEQ      111112$          ; Символов нет
        TST      @#7060          ; Были нажаты клавиши ?
        BEQ      111204$          ; Нет
        RETURN
; Точка входа в подпрограмму
111144$: INC     @#7066          ; Установить флаг вызова п/п перекл. курсора
        MOV      @#23150, R5      ; R5 = адрес текущей таблицы видеопараметров
        MOV      34(R5), @#177016 ; Загрузка регистров кода цвета и кода фона
        MOV      40(R5), @#177020
        MOV      40(R5), @#177022
        MOV      70(R5), @#64(R5) ; Установить цвет курсора
111204$: MOV     @#22542, R0      ; Прочитать символ
        INC      @#22542          ; Подкорректировать указатель
        BIC      #200, @#22542
        MOV      #11126$, -(SP)   ; Положить в стек адрес возврата
        BIC      #177400, R0
        CMP      R0, #40          ; Это управляющие коды ?
        BHIS     111260$          ; Нет
        TSTB     @#22552          ; Установлен режим ИСУ ?
        BEQ      111252$          ; Нет
        JMP      111730$
111252$: ASL     R0              ; Вызов п/п обработки управляющего символа
        JMP      @#13142(R0)
111260$: CMPB    R0, #77          ; Это символы "A-Z, [, \, ], ^, _, a-z, {, |, }, ~" ?
        BLOS     111272$          ; Нет
        BIS      @#22556, R0      ; Подстройка алфавита
111272$: ASL     R0
        MOV      14142(R0), R1     ; R1 = адрес рисунка символа
        MOV      10(R5), R0       ; R0 = длина видеостроки
        MOV      #177010, R2
        MOV      #177024, R3
        MOV      @R5, @R2          ; Уст. в регистре адреса планов нач. адрес
        ADD      56(R5), PC       ; Переход в зависимости от режима (инверсия...)
111322$: .REPT   12              ; Цикл рисования символа
        MOV      (R1)+, R4
        MOV      R4, @R3
        ADD      R0, @R2

```

```

        .ENDR
        MOV      (R1)+, R4
        MOV      R4, @R3
111422$: INC      @R5          ; Передвинуть указатель на следующий адрес
        DEC      16(R5)       ; Уменьшить счетчик оставшихся позиций
        BNE      111624$      ; Еще не конец строки
        MOV      22(R5), @R5  ; Формирование адреса начала следующей строки
        MOV      4(R5), 16(R5)
111444$: ADD      #54, 32(R5)
        CMP      32(R5), 26(R5)
        BNE      111470$
        MOV      24(R5), 32(R5)
111470$: ADD      @32(R5), @R5
        DEC      20(R5)       ; Уменьшить счетчик оставшихся строк
        BNE      111624$      ; Еще не конец экрана
        ADD      74(R5), PC    ; Рулон присутствует ?
        BR       111626$      ; Нет
        INC      20(R5)       ; Увеличить счетчик оставшихся строк
        MOV      #13, -(SP)
        ADD      #54, 30(R5)  ; Увелич. адрес списка ВС верхней части экрана
        CMP      30(R5), #4670
        BNE      111544$
        MOV      #2500, 30(R5)
111544$: MOV      @#23170, R0  ; R0 = кол-во оставшихся видеострок для скрол.
        BEQ      111574$      ; Их нет
        BPL      111556$
        NEG      R0
111556$: CMP      R0, #14      ; <= 12(10) ?
        BLOS     111614$      ; Да
        TRAP     0            ; Вызов цикла подпрограмм
        MOV      @#23150, R5   ; R5 = адрес текущей ТВ
        BR       111544$
111574$: CLR      @#23164      ; Запретить курсор
        MOV      #2, @#7134
        MOV      #177, @23160 ; Убрать курсор с экрана
111614$: CALL     113344$      ; Очистить освободившуюся строку
        ADD      (SP)+, @#23170 ; Осуществить скроллинг
111624$: RETURN
111626$: JMP      114316$      ; Перекинуть курсор на верх экрана
; П/п выключения курсора
111632$: CLR      @#23164      ; Запретить использование курсора
        MOV      60(R5), R0    ; Выключение курсора
        BIC      #1, 6(R0)
        BIC      #1, 52(R0)
        RETURN
; Обработка режима ПОДЧЕРКИВАНИЕ
111660$: MOV      #12, R0      ; R0 = число байт для посылки в видео
111664$: MOV      (R1)+, R4    ; Рисование символа
        MOV      R4, @R3
        ADD      10(R5), @R2
        SOB      R0, 111664$
        MOV      #377, @R3     ; Подчеркнуть символ
        BR       111422$
; Обработка режима ИНВЕРСИЯ
111704$: MOV      #13, R0
        BR       111716$
111712$: ADD      10(R5), @R2
111716$: MOV      (R1)+, R4
        COM      R4            ; Инвертировать байт
        MOV      R4, @R3
        SOB      R0, 111712$
        BR       111422$
; Обработка режима ИСУ
111730$: MOV      #3, -(SP)    ; Формирование начальных значений

```

```

MOV      R0, -(SP)
MOV      R0, -(SP)
MOV      #74, R0          ; Вывод "<"
CALL     111272$
CLR      R0                ; Цикл вывода трех цифр
BR       111762$
111754$: CLR      R0
        ROLB     @SP
        ROLB     R0
111762$: ROLB     @SP
        ROLB     R0
        ROLB     @SP
        ROLB     R0
        ADD      #60, R0
        CALL     111272$
        DEC      4(SP)
        BNE      111754$
        MOV      #76, R0          ; Вывод ">"
        CALL     111272$
        TST      (SP)+
        CMP      #12, (SP)+      ; Это был символ LF ?
        BNE      112034$        ; Нет
        CALL     114310$        ; Вывод CRLF
112034$: TST      (SP)+
        RETURN

```

; ТАБЛИЦЫ ВИДЕОПАРАМЕТРОВ

; Структура таблицы:

```

; - первое слово N - число элементов в таблице
; - N слов - содержимое таблицы
; - 4 слова - значения регистров управления отображением и цветом

```

; ТВ для верхней информационной строки

```

112040$: .WORD    34
        .WORD    175700, 175700, 50, 13, 50, 1, 13, 50, 1, 0, 2410, 12114, 2410, 2410
        .WORD    6, 6, 0, 0, 0, 0, 2400, 2402, 2372, 0, 2410, 2371, 2370, 2
        .WORD    0, 27, 21000, 63104

```

; ТВ для пользовательского экрана

```

112142$: .WORD    37
        .WORD    100000, 100000, 120, 410, 120, 30, 436, 120, 30, 0, 2500, 4670, 2500, 2500
        .WORD    7, 7, 10421, 10421, 10421, 10421, 2470, 2472, 2462, 0, 2500, 2461, 2460
        .WORD    1, 7, 2, 2
        .WORD    7, 7, 135230, 177334

```

; ТВ для экрана установки

```

112252$: .WORD    36
        .WORD    154540, 154540, 50, 334, 50, 24, 334, 50, 24, 0, 5030, 6610, 5030, 5030
        .WORD    7, 7, 10421, 10421, 10421, 10421, 4700, 4702, 4672, 0, 5030, 4671, 4670
        .WORD    2, 7, 2
        .WORD    7, 27, 135230, 177334

```

; ТВ для нижней информационной строки

```

112360$: .WORD    34
        .WORD    176570, 176570, 50, 13, 50, 1, 13, 50, 1, 0, 6754, 7030, 6754, 6754
        .WORD    6, 6, 0, 0, 0, 0, 6750, 6752, 6742, 0, 6754, 6741, 6740, 2
        .WORD    0, 27, 21000, 63104

```

; Таблица описания адресов п/п исполнения управляющих и Esc-последовательностей

```

112462$: .WORD    16, 113220$      ; Ctrl-G
        .WORD    20, 114226$      ; Ctrl-H
        .WORD    24, 114216$      ; Ctrl-J
        .WORD    26, 114772$      ; Ctrl-K
        .WORD    30, 113236$      ; Ctrl-L
        .WORD    32, 114336$      ; Ctrl-M
        .WORD    34, 105260$      ; Ctrl-N
        .WORD    36, 105306$      ; Ctrl-O

```

.WORD	44,114636\$; Ctrl-R
.WORD	46,114514\$; Ctrl-S
.WORD	50,114410\$; Ctrl-T
.WORD	52,114310\$; Ctrl-U
.WORD	54,114620\$; Ctrl-V
.WORD	56,113670\$; Ctrl-W
.WORD	62,114254\$; Ctrl-Y
.WORD	64,114226\$; Ctrl-Z
.WORD	66,113532\$; Ctrl-[
.WORD	70,114106\$; Ctrl-\
.WORD	72,114216\$; Ctrl-]
.WORD	76,114772\$; Ctrl-__
.WORD	112,113602\$; Esc %
.WORD	172,113750\$; Esc =
.WORD	174,113766\$; Esc >
.WORD	176,114004\$; Esc ?
.WORD	202,114106\$; Esc A
.WORD	204,114216\$; Esc B
.WORD	206,114254\$; Esc C
.WORD	210,114226\$; Esc D
.WORD	212,113712\$; Esc E
.WORD	220,114660\$; Esc H
.WORD	222,114106\$; Esc I
.WORD	224,114714\$; Esc J
.WORD	226,114772\$; Esc K
.WORD	240,115010\$; Esc P
.WORD	262,115626\$; Esc Y
.WORD	264,115776\$; Esc Z
.WORD	266,114006\$; Esc [
.WORD	400,114106\$; Esc 200
.WORD	402,114216\$; Esc 201
.WORD	404,114226\$; Esc 202
.WORD	406,114254\$; Esc 203
.WORD	410,114316\$; Esc 204
.WORD	412,114324\$; Esc 205
.WORD	414,114336\$; Esc 206
.WORD	416,114356\$; Esc 207
.WORD	420,114404\$; Esc 210
.WORD	422,114406\$; Esc 211
.WORD	424,114410\$; Esc 212
.WORD	426,114514\$; Esc 213
.WORD	430,114620\$; Esc 214
.WORD	432,114636\$; Esc 215
.WORD	434,114654\$; Esc 216
.WORD	436,114656\$; Esc 217
.WORD	440,114260\$; Esc 220
.WORD	442,114266\$; Esc 221
.WORD	444,114274\$; Esc 222
.WORD	446,114302\$; Esc 223
.WORD	450,114660\$; Esc 224
.WORD	452,113236\$; Esc 225
.WORD	454,114714\$; Esc 226
.WORD	456,114772\$; Esc 227
.WORD	460,114310\$; Esc 230
.WORD	462,115626\$; Esc 231
.WORD	464,116014\$; Esc 232
.WORD	466,116016\$; Esc 233
.WORD	470,116020\$; Esc 234
.WORD	472,0	; Esc 235
.WORD	474,0	; Esc 236
.WORD	476,0	; Esc 237
.WORD	500,116022\$; Esc Граф-@
.WORD	502,116066\$; Esc Граф-A
.WORD	504,116144\$; Esc Граф-B

```

.WORD 506,116234$ ; Esc Граф-C
.WORD 510,116274$ ; Esc Граф-D
.WORD 512,116374$ ; Esc Граф-E
.WORD 514,116462$ ; Esc Граф-F
.WORD 516,116212$ ; Esc Граф-G
.WORD 520,116542$ ; Esc Граф-H
.WORD 522,116544$ ; Esc Граф-I
.WORD 524,116546$ ; Esc Граф-J
.WORD 526,116550$ ; Esc Граф-K
.WORD 556,116552$ ; Esc Граф-W
.WORD 560,116600$ ; Esc Граф-X
.WORD 570,116624$ ; Esc Граф-\
.WORD 572,116632$ ; Esc Граф-]
.WORD 574,116742$ ; Esc Граф-^
.WORD 576,116750$ ; Esc Граф-_
.WORD 0

```

; ПОДПРОГРАММЫ ОБРАБОТКИ УПРАВЛЯЮЩИХ И ESC-ПОСЛЕДОВАТЕЛЬНОСТЕЙ

; П/п обработки Ctrl-G

```

113220$:BIS #600,@#177716 ; Включить звуковой сигнал (частота 500 Гц)
      MOV #20,@#7046 ; Установить время (0.32 сек)
      RETURN

```

; П/п обработки Ctrl-L, Esc 225

```

113236$:MOV @50(R5),-(SP) ; Сохранение содержимого рег. управления цветом
      MOV @52(R5),-(SP)
      MOV 44(R5),@50(R5) ; Засветка всего экрана цветом фона
      MOV 44(R5),@52(R5)
      CALL 113362$ ; Очистка экрана (заполнение цветом фона)
      CALL 110572$ ; Установка ТВ на начало видеопамати
      MOV 30(R5),R4 ; Установка курсора в начало экрана
      MOV R4,32(R5)
      MOV @R4,@R5
      ADD 22(R5),@R5
      MOV 4(R5),16(R5)
      MOV 12(R5),20(R5)
      MOV (SP)+,@52(R5) ; Восстановление регистров управления цветом
      MOV (SP)+,@50(R5)
      MOV 72(R5),@#23164
      RETURN

```

; П/п-мы очистки заданной области экрана (заполнение цветом фона)

; Параметры (для точки входа 113376\$):

; R1 - начальный адрес;

; R2 - длина в байтах;

; R3 - число видеострок;

```

113344$:MOV @32(R5),R1
      ADD 22(R5),R1
113354$:MOV #13,R3
      BR 113372$
113362$:MOV 2(R5),R1
      MOV 14(R5),R3
113372$:MOV 4(R5),R2
113376$:MOV 44(R5),@#177020 ; Загрузка в регистры кода фона цвета фона
      MOV 44(R5),@#177022
      MOV #177010,R4
      MOV R1,@R4 ; Загрузить в регистр адреса планов нач. адрес
      MOV #177024,R1
      CLR R0
      CMP R2,10(R5) ; Длина очищаемой строки равна истинной длине ?
      BEQ 113500$ ; Да
      MOV R2,-(SP) ; Сохранить длину строки
      BR 113450$
113440$:MOV @SP,R2 ; R2 = длина очищаемой строки
      SUB R2,@R4 ; Вычесть длину строки из адреса планов
      ADD 10(R5),@R4 ; Прибавить истинную длину

```



```

113450$:MOV      R0,@R1          ; Очистить байт строки
          INC      @R4           ; Переход к следующему адресу
          SOB      R2,113450$    ; Цикл по длине очищаемой части строки
          SOB      R3,113440$    ; Цикл по числу видеострок
          TST      (SP)+
113462$:MOV      40(R5),@#177020 ; Загр. в регистры кода фона цвета знакоместа
          MOV      40(R5),@#177022
          RETURN
113500$:MUL      R2,R3           ; R3 = длина строки * число видеострок
113502$:MOV      R0,@R1          ; Цикл очистки
          INC      @R4
          SOB      R3,113502$
          BR       113462$
; Значения для загрузки в регистры кода фона (177020 и 177022)
113512$: .WORD   0,10421,21042,31463,42104,52525,63146,73567
; П/п обработки Ctrl-[ (Esc)
113532$:MOV      SP,@#23152
          MOV      #2,@#23174    ; Активировать проверку на управляющие символы
113544$:MOV      #1000,SP
          CALL     110762$        ; Прочитать символ
          ASL      R0             ; Вызов п/п обработки Esc-последовательности
          CALL     @13142(R0)
          CLR      @#23174
          MOV      @#23152,SP
          RETURN
; Точка входа при исполнении Esc %!3 в графическом режиме
113574$:MOV      2(SP),R0
          BR       113620$
; П/п обработки Esc %
113602$:CALL     110762$          ; Прочитать символ
          CMPB     R0,#41         ; Это "!" ?
          BNE      113632$        ; Нет
          CALL     110762$
113620$:BIC      #177774,R0
          ASL      R0             ; Вызов соответствующей п/п
          CALL     @113634(R0)
113632$:RETURN
; Адреса п/п, вызываемые при обработке Esc %!
113634$: .WORD   113644$,113670$,113670$,113670$
; П/п обработки Esc %!0 (переход в графический режим)
113644$:CALL     111632$          ; Выключение курсора
          CLR      @#23174        ; Аннулировать проверку на управляющие символы
          CALL     @#135436$      ; Вызов п/п управления графикой
          MOV      72(R5),@#23164
          RETURN
; П/п обработки Ctrl-W, Esc %! (1-3)
113670$:JSR      R5,105430$       ; Перекодировать клавиши под VT-52
          .WORD    106634$
          MOV      #114226$,@#13162; При Ctrl-N выполнять шаг назад
          CLR      @#7154         ; Установить пункт VT-52 в меню УСТ
          RETURN
; П/п обработки Esc E
113712$:JSR      R5,105430$       ; Перекодировать клавиши под 15ИЭ-00-013
          .WORD    107002$
          MOV      #114660$,@#13162; При Ctrl-N выполнять операцию Home
          MOV      #1,@#7154      ; Установить пункт 15ИЭ-00-013 в меню УСТ
          RETURN
; П/п загрузки регистров кода цвета и фона
113736$:MOV      R5,-(SP)         ; Сохранить адрес текущей ТВ
          CALL     110204$        ; Загрузить регистры
          MOV      (SP)+,R5       ; Восстановить адрес ТВ
          RETURN
; П/п обработки Esc =
113750$:JSR      R5,105346$       ; Установить режим ДКЛ

```

```

        .WORD    107120$,11014
        EMT      52                ; Вывести надпись "ДКЛ"
        .WORD    107576$
        RETURN
; П/п обработки Esc >
113766$:JSR     R5,105400$        ; Сбросить режим ДКЛ
        .WORD    107120$,11014
        EMT      52                ; Стереть надпись "ДКЛ"
        .WORD    107567$
        RETURN
114004$:RETURN
; П/п обработки Esc [
114006$:CALL    110762$           ; Чтение символа
        CMP      R0,#62           ; Это "2" ?
        BEQ      114050$         ; Да
        CMP      R0,#77           ; Это "?" ?
        BEQ      114046$         ; Да
        BR       114034$
114030$:CALL    110762$
114034$:CMPB    R0,#162           ; Это "r" ?
        BNE      114030$         ; Нет
        CALL     113236$         ; Очистка экрана
114046$:RETURN
114050$:CALL    110762$
        CMP      R0,#150         ; Это "h" ?
        BNE      114072$         ; Нет
        MOV      #2,@#7232       ; Запрет передачи одиночных ASCII-кодов клав.
        BR       114104$
114072$:CMP     R0,#154           ; Это "l" ?
        BNE      114104$         ; Нет
        CLR      @#7232          ; Разрешение передачи одиночных ASCII-кодов
114104$:RETURN
; П/п обработки Ctrl-\, Esc A, Esc I, Esc 200
114106$:SUB     @32(R5),@R5       ; Вычесть адрес начала видеостроки
        SUB      #54,32(R5)       ; Перейти к предыд. символьной строке
        CMP      32(R5),24(R5)    ; Был переход за верх экрана ?
        BHIS     114144$         ; Нет
        MOV      26(R5),32(R5)    ; Переход на низ экрана
        SUB      #54,32(R5)
114144$:ADD     @32(R5),@R5       ; Прибавить к положению курсора тек. строку
        CMP      20(R5),12(R5)    ; Это была верхняя строка
        BNE      114204$         ; Нет
        ADD      74(R5),PC        ; Рулон присутствует ?
        BR       114212$         ; Нет
        MOV      32(R5),30(R5)
        MOV      #-13,-(SP)       ; Скроллинг 11(10) видеострок вниз
        JMP      111544$
114204$:INC     20(R5)            ; Количество оставшихся строк ++
        RETURN
114212$:JMP     114324$           ; Позиционирование на низ текущей колонки
; П/п обработки Ctrl-J, Ctrl-], Esc B, Esc 201
114216$:SUB     @32(R5),@R5       ; Вычесть адрес начала видеостроки
        JMP      111444$         ; Перемещение вниз
; П/п обработки Ctrl-H (в системе команд VT-52), Ctrl-Z, Esc D, Esc 202
114226$:CMP     16(R5),4(R5)      ; Курсор в начале строки ?
        BEQ      114246$         ; Да
        DEC      @R5             ; Уменьшить позицию в строке
        INC      16(R5)          ; Увеличить количество оставшихся позиций
        RETURN
114246$:CALL    114356$           ; Позиционирование в конец строки
        BR       114106$         ; На одну строку вверх
; П/п обработки Ctrl-Y, Esc C, Esc 203
114254$:JMP     111422$          ; Перемещение курсора на одну позицию вправо

```

```

; П/п обработки Esc 220
114260$:CALL 114254$ ; Вправо
BR 114106$ ; Вверх
; П/п обработки Esc 221
114266$:CALL 114226$ ; Влево
BR 114216$ ; Вниз
; П/п обработки Esc 222
114274$:CALL 114106$ ; Вверх
BR 114226$ ; Влево
; П/п обработки Esc 223
114302$:CALL 114216$ ; Вниз
BR 114254$ ; Вправо
; П/п обработки Ctrl-U, Esc 230
114310$:CALL 114336$ ; В начало строки
BR 114216$ ; Вниз
; П/п обработки Esc 204
114316$:CLR R0 ; Номер начальной строки
JMP 115720$ ; Позиционирование
; П/п обработки Esc 205
114324$:MOV 12(R5),R0 ; R0 = номер последней строки
DEC R0
JMP 115720$ ; Позиционирование
; П/п обработки Ctrl-M, Esc 206
114336$:MOV 22(R5),@R5 ; Позиционирование в начало строки
ADD @32(R5),@R5
MOV 4(R5),16(R5) ; Количество оставшихся позиций = длина строки
RETURN
; П/п обработки Esc 207
114356$:MOV 4(R5),@R5 ; Позиционирование в конец строки
ADD 22(R5),@R5
DEC @R5
ADD @32(R5),@R5
MOV #1,16(R5) ; Осталась одна позиция
RETURN
114404$:RETURN
114406$:RETURN
; П/п обработки Ctrl-T, Esc 212
114410$:MOV 16(R5),R0 ; R0 = количество оставшихся позиций
DEC R0
BEQ 114474$ ; Осталась одна позиция
MOV @R5,R1
INC R1 ; R1 = текущий видеоадрес + 1
MOV #13,-(SP) ; Число видеострок
MOV #177010,R3
MOV #177024,R4
114440$:MOV R0,R2 ; R2 = число перемещаемых символов
MOV R1,@R3
BR 114452$
114446$:ADD #2,@R3 ; Цикл перемещения символов
114452$:TST @R4
DEC @R3
CLR @R4
SOB R2,114446$
ADD 10(R5),R1 ; Переход к следующей видеостроке
DEC @SP
BNE 114440$
TST (SP)+
114474$:MOV @R5,R1
ADD R0,R1 ; R1 = адрес конечного символа
MOV #1,R2 ; R2 = 1 (длина очищаемой части строки)
MOV #13,R3 ; Число очищаемых видеострок
JMP 113376$ ; Очистка
; П/п обработки Ctrl-S, Esc 213
114514$:MOV 16(R5),R0 ; R0 = количество оставшихся позиций

```

```

DEC      R0
BEQ      114602$      ; Осталась одна позиция
MOV      @R5, R1
ADD      R0, R1      ; R1 = адрес конца строки
DEC      R1
MOV      #13, -(SP)   ; Число видеострок
MOV      #177010, R3
MOV      #177024, R4
114546$: MOV      R0, R2      ; R2 = число перемещаемых символов
MOV      R1, @R3
BR       114560$
114554$: SUB      #2, @R3      ; Цикл перемещения символов
114560$: TST      @R4
INC      @R3
CLR      @R4
SOB      R2, 114554$
ADD      10(R5), R1      ; Переход к следующей видеостроке
DEC      @SP
BNE      114546$
TST      (SP)+
114602$: MOV      @R5, R1      ; R1 = текущий видеоадрес
MOV      #1, R2      ; R2 = длина очищаемой части строки
MOV      #13, R3      ; R3 = число видеострок
JMP      113376$      ; Очистка
; П/п обработки Ctrl-V, Esc 214
114620$: DEC      20(R5)      ; Уменьшить количество оставшихся строк
SUB      #13, @#23170      ; Скроллинг вниз
JMP      114106$      ; Курсор вверх
; П/п обработки Ctrl-R, Esc 215
114636$: INC      20(R5)      ; Увеличить количество оставшихся строк
ADD      #13, @#23170      ; Скроллинг вверх
JMP      114216$      ; Курсор вниз
114654$: RETURN
114656$: RETURN
; П/п обработки Ctrl-H (в системе команд 15ИЭ-00-013), Esc H, Esc 224
114660$: MOV      30(R5), 32(R5) ; Текущий адрес списка ВС - верх экрана
MOV      22(R5), @R5      ; Позиционирование курсора в верх. угол экрана
ADD      @32(R5), @R5
MOV      4(R5), 16(R5)      ; Число ост. позиций = длина строки
MOV      12(R5), 20(R5)      ; Число ост. строк = число строк на экране
RETURN
; П/п обработки Esc J, Esc 226
114714$: CALL     114772$      ; Очистка конца строки
MOV      20(R5), -(SP)      ; Количество оставшихся строк
MOV      32(R5), R1      ; R1 = текущий адрес списка ВС
BR       114762$
114732$: ADD      #54, R1      ; Перейти к следующей строке
CMP      R1, 26(R5)      ; Дошли до конца экрана ?
BNE      114750$      ; Нет
MOV      24(R5), R1      ; R1 = начальный адрес списка ВС
114750$: MOV      R1, -(SP)
MOV      @R1, R1
CALL     113354$      ; Очистка
MOV      (SP)+, R1
114762$: DEC      @SP
BNE      114732$
TST      (SP)+
RETURN
; П/п обработки Ctrl-K, Esc K, Esc 227
114772$: MOV      @R5, R1      ; R1 = текущий видеоадрес
MOV      16(R5), R2      ; R2 = количество оставшихся позиций в строке
MOV      #13, R3      ; R3 = число видеострок
JMP      113376$      ; Очистка
; П/п обработки Esc P

```

```

115010$:CLR      -(SP)
115012$:CALL     110762$      ; Чтение символа
      MOV      R0,-(SP)      ; Сохранение его в стеке
      CMPB     R0,#73        ; Это ";" ?
      BNE      115012$
115026$:CALL     110762$
      MOV      R0,-(SP)
      CMPB     R0,#73        ; Это ";" ?
      BEQ      115424$        ; Да
      CMPB     R0,#174       ; Это "|" ?
      BNE      115026$
115050$:TST      (SP)+        ; Восстановление стека
      BNE      115050$
      ASR      -4(SP)        ; Была набрана "1" ?
      BCS      115112$        ; Да
      MOV      #115342$,R1    ; R1 = список смещ. клавиш K1-K25 в табл. раск.
      MOV      #31,R2        ; R2 = 25(10) число программир. клавиш
115072$:MOV      (R1)+,R0      ; Цикл очистки назначений клавиш
      MOV      #11140,7232(R0)
      SOB      R2,115072$
      MOV      #11142,@#13140 ; Установка указателя на начало буфера
115112$:CLR      -(SP)
115114$:CALL     110762$
      CMPB     R0,#33        ; Это Escape ?
      BEQ      115302$        ; Да
      CMPB     R0,#57        ; Это "/" ?
      BEQ      115154$        ; Да
      MOV      @SP,R1        ; @SP = @SP * 10 + R0
      MUL      #10.,R1
      BIC      #177760,R0
      ADD      R0,R1
      MOV      R1,@SP
      BR       115114$
115154$:MOV      (SP)+,R0
      CMP      R0,#25.        ; Номер клавиши в диапазоне 1 -- 25 ?
      BHI      115112$        ; Нет
      DEC      R0
      ASL      R0
      MOV      115342$(R0),R0 ; R0 = смещение клавиши в таблице раскладки
      MOV      @#13140,7232(R0); Запомнить адрес строки
115202$:CALL     110762$
      CMPB     R0,#33        ; Это Escape ?
      BEQ      115304$
      CMPB     R0,#73        ; Это ";" ?
      BEQ      115270$
      CALL     115322$        ; Преобразование HEX-цифры в двоичный код
      ASH      #4,R0          ; R0 = R0 * 16
      MOV      R0,-(SP)
      CALL     110762$
      CALL     115322$
      BIS      (SP)+,R0
      MOVB     R0,@13140      ; Запомнить символ в строке
      CMP      @#13140,#13130 ; Достигнут конец буфера
      BHI      115202$        ; Да
      INC      @#13140        ; Продвинуть указатель на следующую позицию
      BR       115202$
115270$:CLRB     @13140        ; Пометить конец строки
      INC      @#13140
      BR       115112$
115302$:TST      (SP)+
115304$:CLRB     @13140
      INC      @#13140
      CALL     110762$
      RETURN

```

; П/п преобразования HEX-цифры в двоичное представление

```
115322$:CMP      R0, #101          ; Цифра в диапазоне "A-F" ?
          BLO      115334$          ; Нет
          ADD      #11, R0
115334$:BIC      #177760, R0
          RETURN
; Смещения для клавиш K1-K25 в таблице раскладки клавиатуры
115342$: .WORD    10, 12, 14, 20, 22, 270, 272, 274, 300, 302, 550, 552, 554, 560, 562
          .WORD    1030, 1032, 1034, 1040, 1042, 1310, 1312, 1314, 1320, 1322
115424$:MOV      2(SP), R0          ; R0 = код симв., шрифт кот. перепрограммир.
115430$:TST      (SP)+              ; Восстановление стека
          BNE      115430$
          MOV      R0, R1
          ASL      R0
          ADD      #14142, R0        ; R0 = указатель на ячейку адреса рисунка симв.
          MOV      R0, -(SP)
          MUL      #11, R1          ; Вычисление местоположения рисунка символа
          ADD      #15142, R1        ; R1 = адрес местоположения рисунка символа
          MOV      R1, -(SP)
115460$:CALL     110762$            ; Чтение символа
          CMP      R0, #33          ; Это Escape ?
          BEQ      115550$          ; Да
          CMP      R0, #100         ; Это "@" ?
          BNE      115460$          ; Нет
115500$:MOV      @SP, @2(SP)        ; Сохранить адрес рисунка в таблице
          ADD      #2, 2(SP)        ; Переход на следующую позицию в табл. адресов
          JSR      R5, 115554$      ; Программирование первых пяти байт
          .WORD    5
          CALL     110762$
          JSR      R5, 115554$      ; Программирование следующих шести байт
          .WORD    6
          CALL     110762$
          CMP      R0, #33          ; Это Escape
          BNE      115500$
          CALL     110762$
115550$:CMP      (SP)+, (SP)+
          RETURN
```

; П/п программирования заданного количества байт в рисунке символа

```
115554$:MOV      #10, -(SP)         ; Длина байта
115560$:MOV      R5, -(SP)
          CALL     110762$
          MOV      (SP)+, R5
          CMP      R0, #33          ; Это Escape ?
          BEQ      115616$          ; Да
          MOV      4(SP), R3        ; R3 = адрес рисунка
          MOV      @R5, R4          ; R4 = число программируемых байт
115604$:ASR      R0                  ; Цикл программирования
          RORB     (R3)+
          SOB      R4, 115604$
          DEC      @SP
          BNE      115560$
115616$:ADD      (R5)+, 4(SP)        ; Перейти к следующей области рисунка
          TST      (SP)+
          RTS      R5
```

; П/п обработки Esc Y, Esc 231

```
115626$:CALL     110762$            ; Прочитать символ
          SUB      #40, R0
          BMI      115652$
          CMP      R0, 12(R5)        ; Координата Y превышает диапазон ?
          BHIS     115652$          ; Да
          CALL     115720$          ; Позиционирование по координате Y
115652$:CALL     110762$
          SUB      #40, R0
          BMI      115716$
```

```

        CMP      R0,4(R5)          ; Координата X превышает диапазон ?
        BHIS     115716$           ; Да
115672$:MOV     4(R5),16(R5)       ; Позиционирование по координате X
        SUB      R0,16(R5)        ; (16)R5 - кол-во оставшихся позиций
        ADD      @32(R5),R0        ; R0 = адрес видеопамати начала строки + X
        ADD      22(R5),R0        ; Плюс константа начального доступа
        MOV      R0,@R5           ; @R5 - адр.начала видеопамати для выв. симв.
115716$:RETURN
; П/п позиционирования по координате Y
115720$:SUB     @32(R5),@R5        ; @R5 - оставить координату X
        MOV      12(R5),20(R5)    ;
        SUB      R0,20(R5)        ; 20(R5) - кол-во ост. символьных строк
        MUL      #11.*4,R0
        ADD      30(R5),R1        ; R1 = адрес описания видеостроки
        ADD      74(R5),PC        ; Рулон присутствует ?
        BR       115766$          ; Нет
        CMP      R1,#4670         ; Переход за границу списка ?
        BLO      115766$          ; Нет
        SUB      #2170,R1         ; Подкорректировать значение
115766$:MOV     R1,32(R5)         ; 32(R5) - текущий адрес списка видеострок
        ADD      @R1,@R5          ; @R5 - адр.начала видеопамати для выв.симв.
        RETURN
; П/п обработки Esc Z
115776$:MOV     #116010$,R0       ; R0 = адрес символьной строки
        CALL     104160$          ; Передача адреса строки п/п облс. клавиатуры
        RETURN
116010$: .ASCIZ <33>"/Z"        ; Терминал VT-100 в режиме VT-52
        .EVEN
116014$:RETURN
116016$:RETURN
116020$:RETURN
; П/п обработки Esc Граф-@
116022$:ADD     @#23176,PC        ; Установлен режим отмены ?
        BR       116036$          ; Нет
        MOV      36(R5),R0        ; R0 = предыдущий цвет символа
        BR       116046$
116036$:CALL    110762$           ; Прочесть символ
        BIC      #177770,R0
116046$:MOV     R0,@#177016       ; Положить номер цвета в регистр кода цвета
116052$:MOV     34(R5),36(R5)    ; Запомнить предыдущий цвет символа
        MOV      R0,34(R5)        ; Сохранить номер цвета в ТВ
        RETURN
; П/п обработки Esc Граф-A
116066$:ADD     @#23176,PC        ; Установлен режим отмены ?
        BR       116102$          ; Нет
        MOV      42(R5),R0        ; R0 = предыдущий цвет знакоместа
        BR       116120$
116102$:CALL    110762$           ; Прочесть символ
        BIC      #177770,R0
        ASL      R0
        MOV      113512$(R0),R0   ; R0 = код фонового цвета
116120$:MOV     R0,@#177020       ; Положить номер фон. цвета в рег. кода фона
        MOV      R0,@#177022
116130$:MOV     40(R5),42(R5)    ; Запомнить предыдущий цвет знакоместа
        MOV      R0,40(R5)        ; Сохранить номер цвета знакоместа в ТВ
        RETURN
; П/п обработки Esc Граф-B
116144$:ADD     @#23176,PC        ; Установлен режим отмены ?
        BR       116160$          ; Нет
        MOV      46(R5),R0        ; R0 = предыдущий цвет фона
        BR       116046$
116160$:CALL    110762$           ; Прочесть символ
        BIC      #177770,R0
116170$:ASL     R0

```

```

MOV      113512$(R0),R0 ; R0 = код фонового цвета
MOV      44(R5),46(R5) ; Запомнить предыдущий цвет фона
MOV      R0,44(R5) ; Сохранить номер цвета фона в ТВ
RETURN
; П/п обработки Esc Граф-G
116212$:CALL 110762$ ; Прочесть символ
BIC      #177770,R0 ; R0 = номер цвета курсора
MOVB     R0,@64(R5) ; Изменить цвет курсора
116226$:MOV R0,70(R5) ; Сохранить цвет курсора в ТВ
RETURN
; П/п обработки Esc Граф-C
116234$:ADD @#23176,PC ; Установлен режим отмены ?
BR       116254$ ; Нет
CLR      @#22554 ; Сбросить индикатор режима инверсии(подч.)
EMT      52 ; Очистить надпись "ИНВ" ("ПОДЧ")
.WORD    107605$
BR       116334$
116254$:MOV #<111704$-111322$>,56(R5) ; Установить режим инверсии (362)
INCB     @#22554 ; Установить индикатор режима инверсии
EMT      52 ; Высветить надпись "ИНВ"
.WORD    107614$
RETURN
; П/п обработки Esc Граф-D
116274$:ADD @#23176,PC ; Установлен режим отмены ?
BR       116314$
CLR      @#22554 ; Сбросить индикатор режима подчеркивания(инв.)
EMT      52 ; Погасить надпись "ПОДЧ" ("ИНВ")
.WORD    107605$
BR       116334$
116314$:MOV #<111660$-111322$>,56(R5) ; Установить режим подч. (336)
INCB     @#22555 ; Установить индикатор режима подчеркивания
EMT      52 ; Высветить надпись "ПОДЧ"
.WORD    107623$
RETURN
116334$:CLR 56(R5) ; Установить нормальный режим
TSTB     @#22554 ; Ранее был установлен режим инверсии ?
BEQ      116356$ ; Нет
MOV      #364,56(R5) ; Установить режим инверсии
BR       116372$
116356$:TSTB @#22555 ; Ранее был установлен режим подчеркивания ?
BEQ      116372$ ; Нет
MOV      #340,56(R5) ; Установить режим подчеркивания
116372$:RETURN
; П/п обработки Esc Граф-E
116374$:CALL 110762$ ; Прочесть символ
SUB      #61,R0 ; Символ в диапазоне "1"-"4"
BMI      116422$ ; Нет
CMP      R0,#3
BHI      116422$ ; Нет
ASL      R0 ; Вызов соответствующей п/п
CALL     @116424$(R0)
116422$:RETURN
; Адреса п/п для установки длины строки
116424$: .WORD 116434$,116450$,116460$,116460$
; П/п установки длины строки 80
116434$:MOV #120,4(R5) ; Установить длину строки 80
116442$:CLR 22(R5) ; Константа начального отступа = 0
RETURN
; П/п установки длины строки 40
116450$:MOV #50,4(R5) ; Установить длину строки 40
BR       116442$
116460$:RETURN
; П/п обработки Esc Граф-F
116462$:CALL 110762$ ; Прочесть символ

```



```

SUB      #61,R0          ; Символ в диапазоне "1"-"4"
BMI      116524$         ; Нет
CMP      R0,#4
BHS      116524$         ; Нет
116502$:MOVB 116526$(R0),@22732; Установить соответствующий формат экрана
MOVB     116532$(R0),22744 ; Сохранить номер формата в ТВ
MOVB     116536$(R0),22662 ; Занести новую длину строки в ТВ
116524$:RETURN
; Биты формата экрана для загрузки в регистр управления отображением
116526$: .BYTE 0,20,40,60
; Номера форматов для загрузки в ТВ
116532$: .BYTE 1,2,4,10
; Длины строк
116536$: .BYTE 120,50,24,12
116542$:RETURN
116544$:RETURN
116546$:RETURN
116550$:RETURN
; П/п обработки Esc Граф-W
116552$:ADD @#23176,PC    ; Установлен режим отмены ?
BR        116570$        ; Нет
BIS       #1000,@#177054 ; Выключить сетевой таймер в канале ЦП
RETURN
116570$:BIC #1000,@#177054 ; Включить сетевой таймер в канале ЦП
RETURN
; П/п обработки Esc Граф-X
116600$:MTPS #200        ; Запретить прерывания
MOVB      #177,@23160    ; Убрать курсор с экрана
CALL      104026$        ; Вызов п/п ПАУЗА
MTPS      #0            ; Разрешить прерывания
RETURN
; П/п обработки Esc Граф-\
116624$:INC @#7072       ; Установить индикатор вызова меню ЗАГРУЗКА
RETURN
; П/п обработки Esc Граф-]
116632$:ADD @#23176,PC    ; Установлен режим отмены ?
BR        116706$        ; Нет
JSR       R5,105400$     ; Установить пользовательский режим клавиатуры
.WORD     107310$,11054
CLR       @#7146         ; Сбросить индикатор системного режима
ADD       @#22550,PC     ; Разрешена установка/снятие сигнала HALT ?
BR        116664$        ; Да
RETURN
116664$:BIC #20,@#177716 ; Снять сигнал HALT в ЦП
MOV       #70046,@#177010 ; Сброс ключа продолжения в области ОЗУ ЦП
CLR       @#177014
RETURN
116706$:JSR R5,105346$    ; Установить системный режим клавиатуры
.WORD     107310$,11054
INC       @#7146         ; Установить индикатор системного режима
ADD       @#22550,PC     ; Разрешена установка/снятие сигнала HALT ?
BR        116732$        ; Да
RETURN
116732$:BIS #20,@#177716 ; Установить сигнал HALT в ЦП
RETURN
; П/п обработки Esc Граф-^
116742$:INC @#7100       ; Установить индикатор вызова тест-программы
RETURN
; П/п обработки Esc Граф-_
116750$:MOV #2,@#23176    ; Установить режим отмены
116756$:CALL 110762$      ; Прочесть символ
CMP       R0,#33         ; Это Escape ?
BEQ       116756$        ; Да
ASL       R0            ; Вызов соответствующей п/п

```

```

CALL    @13142(R0)
CLR     @#23176          ; Сбросить режим отмены
RETURN

; П/п установки цвета курсора, вызываемая при исп. послед. Esc TD (графика)
117004$:MOV    2(SP),R0
        BIC     #177760,R0      ; R0 = номер цвета курсора
        MOV     R0,@#7166      ; Сохр. ном. цвета кур. в пункте УСТ/цв. кур.
        MOVB    R0,@22742      ; Изменить цвет курсора
        RETURN

; П/п установки цветов символа, знакоместа и фона, вызываемая при исполнении
; последовательности Esc LI (графика)
117026$:MOV    R5, -(SP)        ; Сохранить R5
        MOV     @#23150,R5      ; R5 = адрес текущей ТВ
        MOV     4(SP),R0        ; R0 = номер цвета символа
        BMI     117052$        ; Если отрицательный - нет установки
        MOV     R0,34(R5)       ; Сохранить номер цвета в ТВ
        MOV     R0,@#7160      ; Сохр. ном. цвета симв. в пункте УСТ/цв. симв.
117052$:MOV    6(SP),R0        ; R0 = номер цвета знакоместа
        BMI     117074$        ; Если отрицательный - нет установки
        MOV     R0,@#7162      ; Сохр. ном. цвета зн. в пункте УСТ/цв. зн.
        ASL     R0
        MOV     113512$(R0),40(R5); Сохранить номер фонового цвета в ТВ
117074$:MOV    10(SP),R0        ; R0 = номер цвета фона
        BMI     117116$        ; Если отрицательный - нет установки
        MOV     R0,@#7164      ; Сохр. ном. цвета фона в пункте УСТ/цвет фона
        ASL     R0
        MOV     113512$(R0),44(R5); Сохранить номер фонового цвета в ТВ
117116$:CALL   110230$          ; Заполнить регистры кода цвета и фона
        MOV     (SP)+,R5        ; Восстановить R5
        RETURN

; П/п исполнения запроса EMT 42 (вывод символа)
117126$:CALL   117136$          ; Занести символ в буфер
        TRAP    0              ; Запуск цикла задач
        RETURN

; П/п заноса символа в буфер
117136$:MOVB   R0,@22544        ; Занести символ в буфер
        INC     @#22544        ; Корректировка указателя
        BIC     #200,@#22544
        INC     @#7064         ; Увеличить счетчик символов
        RETURN

; П/п исполнения запроса EMT 44 (вывод строки символов)
117162$:MOV    @16(SP),R1      ; R1 = адрес строки символов
        ADD     #2,16(SP)      ; Коррекция адреса возврата
; П/п исполнения запроса EMT 46 (вывод строки символов)
117174$:JSR    R5,117206$      ; Занести строку в буфер
        TRAP    0              ; Запуск цикла задач
        RETURN

; П/п заноса строки символов в буфер
117204$:MOV    (R5)+,R1        ; R1 = адрес строки символов
117206$:MOV    @#22544,R0      ; R0 = значение указателя для записи в буфер
        BR     117224$
117214$:BIC     #200,R0        ; Коррекция указателя
        INC     @#7064         ; Увеличить счетчик символов
117224$:MOVB   (R1)+,(R0)+    ; Занести очередной символ в буфер
        BNE     117214$        ; Еще не конец строки
        DEC     R0
        MOV     R0,@#22544
        RTS     R5

; П/п исполнения запроса EMT 56 (вывод в нижнюю инф. строку)
117240$:MOV    #23052,R5      ; R5 = адрес ТВ для нижней инф. строки
        BR     117260$
117246$:BR     117376$
; П/п исполнения запроса EMT 52 (вывод в верхнюю инф. строку)
117250$:ADD    @#7150,PC      ; Разрешение/запрет вывода в верхнюю строку

```

```

MOV      #22560,R5      ; R5 = адрес ТВ для верхней инф. строки
117260$:MOV 34(R5),@#177016 ; Загрузка регистров кода цвета и фона
MOV      40(R5),@#177020
MOV      40(R5),@#177022
MOV      @32(R5),@R5    ; Вычисление видеоадреса
MOV      4(R5),16(R5)
MOV      @16(SP),R1     ; R1 = адрес строки
MOVB     (R1)+,R0       ; R0 = начальная позиция
ADD      R0,@R5
SUB      R0,16(R5)
BR       117342$
117332$:MOV R1,-(SP)
CALL     111272$        ; Рисование символа
MOV      (SP)+,R1
117342$:CLR R0
BISB     (R1)+,R0       ; R0 = очередной символ
BNE      117332$        ; Еще не конец строки
MOV      @23150,R5      ; R5 = адрес текущей ТВ
MOV      34(R5),@#177016 ; Восстановление регистров кода цвета и фона
MOV      40(R5),@#177020
MOV      40(R5),@#177022
117376$:ADD #2,16(SP)   ; Коррекция адреса возврата
RETURN
117406$:RETURN
117410$:RETURN
117412$:RETURN
117414$:RETURN
117416$:RETURN
117420$:RETURN
117422$:RETURN
117424$:RETURN
117426$:RETURN

```

; * ТАБЛИЦА РИСУНКОВ СИМВОЛОВ *****

```

117430$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-@
117443$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-A
117456$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-B
117471$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-C
117504$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-D
117517$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-E
117532$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-F
117545$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-G
117560$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-H
117573$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-I
117606$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-J
117621$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-K
117634$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-L
117647$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-M
117662$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-N
117675$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-O
117710$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-P
117723$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-Q
117736$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-R
117751$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-S
117764$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-T
117777$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-U
120012$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-V
120025$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-W
120040$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-X
120053$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-Y
120066$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-Z
120101$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-[
120114$: .BYTE 0,0,0,0,0,0,0,0,0,0,0,0 ; Ctrl-\

```

120127\$:. BYTE	0,0,0,0,0,0,0,0,0,0,0,0	; Ctrl-]
120142\$:. BYTE	0,0,0,0,0,0,0,0,0,0,0,0	; Ctrl-^
120155\$:. BYTE	0,0,0,0,0,0,0,0,0,0,0,0	; Ctrl-_
120170\$:. BYTE	0,0,0,0,0,0,0,0,0,0,0,0	; Space
120203\$:. BYTE	0,20,20,20,20,20,20,0,20,0,0	; !
120216\$:. BYTE	0,44,44,22,0,0,0,0,0,0,0	; "
120231\$:. BYTE	0,0,44,44,176,44,176,44,44,0,0	; #
120244\$:. BYTE	0,0,102,74,102,102,102,74,102,0,0	; \$
120257\$:. BYTE	0,0,206,106,40,20,10,304,302,0,0	; %
120272\$:. BYTE	0,30,44,44,30,24,242,102,274,0,0	; &
120305\$:. BYTE	0,60,20,10,0,0,0,0,0,0,0	; '
120320\$:. BYTE	0,40,20,10,10,10,10,20,40,0,0	; (
120333\$:. BYTE	0,10,20,40,40,40,40,20,10,0,0	;)
120346\$:. BYTE	0,0,0,20,124,70,124,20,0,0,0	; *
120361\$:. BYTE	0,0,20,20,20,376,20,20,20,0,0	; +
120374\$:. BYTE	0,0,0,0,0,0,0,30,30,20,10	; ,
120407\$:. BYTE	0,0,0,0,0,176,0,0,0,0,0	; -
120422\$:. BYTE	0,0,0,0,0,0,0,30,30,0,0	; .
120435\$:. BYTE	0,0,200,100,40,20,10,4,2,0,0	; /
120450\$:. BYTE	0,74,102,142,122,112,106,102,74,0,0	; 0
120463\$:. BYTE	0,20,30,24,20,20,20,20,174,0,0	; 1
120476\$:. BYTE	0,74,102,102,40,20,10,104,176,0,0	; 2
120511\$:. BYTE	0,176,40,20,70,100,100,102,74,0,0	; 3
120524\$:. BYTE	0,40,60,50,44,42,176,40,40,0,0	; 4
120537\$:. BYTE	0,176,2,2,76,100,100,102,74,0,0	; 5
120552\$:. BYTE	0,74,102,2,76,102,102,102,74,0,0	; 6
120565\$:. BYTE	0,176,102,40,20,10,4,2,2,0,0	; 7
120600\$:. BYTE	0,74,102,102,74,102,102,102,74,0,0	; 8
120613\$:. BYTE	0,74,102,102,102,174,100,102,74,0,0	; 9
120626\$:. BYTE	0,0,0,30,30,0,0,30,30,0,0	; :
120641\$:. BYTE	0,0,0,30,30,0,0,30,30,20,10	; ;
120654\$:. BYTE	0,0,40,20,10,4,10,20,40,0,0	; <
120667\$:. BYTE	0,0,0,0,176,0,176,0,0,0,0	; =
120702\$:. BYTE	0,0,4,10,20,40,20,10,4,0,0	; >
120715\$:. BYTE	0,70,104,100,40,20,20,0,20,0,0	; ?
120730\$:. BYTE	0,0,74,102,200,234,222,222,154,0,0	; @
120743\$:. BYTE	0,30,44,102,102,176,102,102,102,0,0	; A
120756\$:. BYTE	0,76,102,102,76,102,102,102,76,0,0	; B
120771\$:. BYTE	0,74,102,2,2,2,2,102,74,0,0	; C
121004\$:. BYTE	0,76,104,204,204,204,204,104,76,0,0	; D
121017\$:. BYTE	0,176,2,2,76,2,2,2,176,0,0	; E
121032\$:. BYTE	0,176,2,2,76,2,2,2,2,0,0	; F
121045\$:. BYTE	0,74,102,2,2,162,102,102,74,0,0	; G
121060\$:. BYTE	0,102,102,102,102,176,102,102,102,0,0	; H
121073\$:. BYTE	0,70,20,20,20,20,20,20,70,0,0	; I
121106\$:. BYTE	0,70,20,20,20,20,20,22,14,0,0	; J
121121\$:. BYTE	0,102,42,22,16,12,22,42,102,0,0	; K
121134\$:. BYTE	0,2,2,2,2,2,2,2,176,0,0	; L
121147\$:. BYTE	0,202,306,252,222,202,202,202,202,0,0	; M
121162\$:. BYTE	0,102,102,106,112,122,142,102,102,0,0	; N
121175\$:. BYTE	0,74,102,102,102,102,102,102,74,0,0	; O
121210\$:. BYTE	0,76,102,102,102,76,2,2,2,0,0	; P
121223\$:. BYTE	0,74,102,102,102,102,102,122,74,100,0	; Q
121236\$:. BYTE	0,76,102,102,102,76,22,42,102,0,0	; R
121251\$:. BYTE	0,74,102,2,74,100,100,102,74,0,0	; S
121264\$:. BYTE	0,376,20,20,20,20,20,20,20,0,0	; T
121277\$:. BYTE	0,102,102,102,102,102,102,102,74,0,0	; U
121312\$:. BYTE	0,202,202,202,202,202,104,50,20,0,0	; V
121325\$:. BYTE	0,202,202,202,202,222,252,306,202,0,0	; W
121340\$:. BYTE	0,102,102,44,30,30,44,102,102,0,0	; X
121353\$:. BYTE	0,202,202,104,50,20,20,20,20,0,0	; Y
121366\$:. BYTE	0,176,100,40,20,10,4,2,176,0,0	; Z
121401\$:. BYTE	0,70,10,10,10,10,10,10,70,0,0	; [
121414\$:. BYTE	0,0,2,4,10,20,40,100,200,0,0	; \

```

121427$:.BYTE      0,70,40,40,40,40,40,70,0,0      ; ]
121442$:.BYTE      0,0,0,0,176,100,100,0,0,0,0      ; ^
121455$:.BYTE      0,0,0,0,0,0,0,0,176,0,0          ; _
121470$:.BYTE      0,30,20,40,0,0,0,0,0,0,0          ; `
121503$:.BYTE      0,0,0,74,100,174,102,102,274,0,0  ; a
121516$:.BYTE      0,2,2,72,106,102,102,106,72,0,0  ; b
121531$:.BYTE      0,0,0,74,102,2,2,102,74,0,0      ; c
121544$:.BYTE      0,100,100,134,142,102,102,142,134,0,0 ; d
121557$:.BYTE      0,0,0,74,102,102,176,2,174,0,0  ; e
121572$:.BYTE      0,60,110,10,34,10,10,10,10,0,0  ; f
121605$:.BYTE      0,0,0,134,142,102,102,142,134,100,74 ; g
121620$:.BYTE      0,2,2,72,106,102,102,102,102,0,0 ; h
121633$:.BYTE      0,20,0,30,20,20,20,20,70,0,0     ; i
121646$:.BYTE      0,20,0,20,20,20,20,20,20,22,14   ; j
121661$:.BYTE      0,2,2,42,22,16,22,42,102,0,0     ; k
121674$:.BYTE      0,30,20,20,20,20,20,20,70,0,0     ; l
121707$:.BYTE      0,0,0,156,222,222,222,222,222,0,0 ; m
121722$:.BYTE      0,0,0,72,106,102,102,102,102,0,0 ; n
121735$:.BYTE      0,0,0,74,102,102,102,102,74,0,0  ; o
121750$:.BYTE      0,0,0,72,106,102,102,106,72,2,2  ; p
121763$:.BYTE      0,0,0,134,142,102,102,142,134,100,100 ; q
121776$:.BYTE      0,0,0,72,106,2,2,2,2,0,0         ; r
122011$:.BYTE      0,0,0,74,2,74,100,102,74,0,0     ; s
122024$:.BYTE      0,10,10,34,10,10,10,110,60,0,0   ; t
122037$:.BYTE      0,0,0,102,102,102,102,142,134,0,0 ; u
122052$:.BYTE      0,0,0,202,202,202,104,50,20,0,0  ; v
122065$:.BYTE      0,0,0,202,202,202,222,222,154,0,0 ; w
122100$:.BYTE      0,0,0,102,44,30,30,44,102,0,0    ; x
122113$:.BYTE      0,0,0,102,102,102,44,30,10,4,2   ; y
122126$:.BYTE      0,0,0,176,40,20,10,4,176,0,0    ; z
122141$:.BYTE      0,0,140,20,20,14,20,20,140,0,0   ; {
122154$:.BYTE      0,0,20,20,20,20,20,20,20,0,0     ; |
122167$:.BYTE      0,0,14,20,20,140,20,20,14,0,0    ; }
122202$:.BYTE      0,176,0,0,0,0,0,0,0,0,0          ; ~
122215$:.BYTE      0,176,176,176,176,176,176,176,176,0,0 ; Delete
122230$:.BYTE      0,10,34,52,10,10,10,10,10,0,0    ; 200
122243$:.BYTE      0,10,10,10,10,10,52,34,10,0,0    ; 201
122256$:.BYTE      0,0,0,4,2,377,2,4,0,0,0          ; 202
122271$:.BYTE      0,0,0,40,100,377,100,40,0,0,0    ; 203
122304$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 204
122317$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 205
122332$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 206
122345$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 207
122360$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 210
122373$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 211
122406$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 212
122421$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 213
122434$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 214
122447$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 215
122462$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 216
122475$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 217
122510$:.BYTE      0,0,0,104,50,20,50,104,0,0,0     ; 220
122523$:.BYTE      0,0,0,20,0,174,0,20,0,0,0        ; 221
122536$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 222
122551$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 223
122564$:.BYTE      0,100,100,100,110,104,176,4,10,0,0 ; 224
122577$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 225
122612$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 226
122625$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 227
122640$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 230
122653$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 231
122666$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 232
122701$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 233
122714$:.BYTE      0,0,0,0,0,0,0,0,0,0,0            ; 234

```

122727\$:. BYTE	0,0,0,0,0,0,0,0,0,0,0,0	; 235
122742\$:. BYTE	0,0,0,0,0,0,0,0,0,0,0,0	; 236
122755\$:. BYTE	0,0,0,0,0,0,0,0,0,0,0,0	; 237
122770\$:. BYTE	24,24,24,24,24,27,24,24,24,24,24,24	; Граф-@
123003\$:. BYTE	10,10,10,10,10,377,0,0,0,0,0,0	; Граф-A
123016\$:. BYTE	24,24,24,24,24,364,24,24,24,24,24,24	; Граф-B
123031\$:. BYTE	0,0,0,0,0,17,10,10,10,10,10,10	; Граф-C
123044\$:. BYTE	24,24,24,24,24,24,24,24,24,24,24,24	; Граф-D
123057\$:. BYTE	10,10,10,10,10,370,10,10,10,10,10,10	; Граф-E
123072\$:. BYTE	10,10,10,10,10,370,0,0,0,0,0,0	; Граф-F
123105\$:. BYTE	24,24,24,24,364,4,364,24,24,24,24,24	; Граф-G
123120\$:. BYTE	0,0,0,0,374,4,364,24,24,24,24,24	; Граф-H
123133\$:. BYTE	10,10,10,10,377,0,377,10,10,10,10,10	; Граф-I
123146\$:. BYTE	0,0,0,0,0,370,10,10,10,10,10,10	; Граф-J
123161\$:. BYTE	0,0,0,0,0,377,10,10,10,10,10,10	; Граф-K
123174\$:. BYTE	24,24,24,24,367,0,377,0,0,0,0,0,0	; Граф-L
123207\$:. BYTE	24,24,24,24,24,377,0,0,0,0,0,0	; Граф-M
123222\$:. BYTE	10,10,10,10,10,17,10,10,10,10,10,10	; Граф-N
123235\$:. BYTE	24,24,24,24,367,0,367,24,24,24,24,24	; Граф-O
123250\$:. BYTE	10,10,10,10,10,377,10,10,10,10,10,10	; Граф-P
123263\$:. BYTE	10,10,10,10,370,10,370,10,10,10,10,10	; Граф-Q
123276\$:. BYTE	24,24,24,24,24,367,24,24,24,24,24,24	; Граф-R
123311\$:. BYTE	0,0,0,0,0,377,24,24,24,24,24,24	; Граф-S
123324\$:. BYTE	10,10,10,10,10,377,0,377,0,0,0,0,0	; Граф-T
123337\$:. BYTE	0,0,0,0,0,377,0,0,0,0,0,0,0	; Граф-U
123352\$:. BYTE	24,24,24,24,364,4,374,0,0,0,0,0,0	; Граф-V
123365\$:. BYTE	10,10,10,10,10,10,10,10,10,10,10,10	; Граф-W
123400\$:. BYTE	0,0,0,0,377,0,377,10,10,10,10,10,10	; Граф-X
123413\$:. BYTE	10,10,10,10,10,17,0,0,0,0,0,0,0	; Граф-Y
123426\$:. BYTE	0,0,0,0,377,0,377,0,0,0,0,0,0	; Граф-Z
123441\$:. BYTE	24,24,24,24,27,20,27,24,24,24,24,24	; Граф-[
123454\$:. BYTE	24,24,24,24,27,20,37,0,0,0,0,0,0	; Граф-\
123467\$:. BYTE	0,0,0,0,377,0,367,24,24,24,24,24,24	; Граф-]
123502\$:. BYTE	10,10,10,10,17,10,17,10,10,10,10,10	; Граф-^
123515\$:. BYTE	0,0,0,0,37,20,27,24,24,24,24,24,24	; Граф-__
123530\$:. BYTE	0,0,0,142,222,222,236,222,142,0,0,0	; ю
123543\$:. BYTE	0,0,0,74,100,174,102,102,274,0,0,0	; а
123556\$:. BYTE	0,100,74,2,76,102,102,102,74,0,0,0	; б
123571\$:. BYTE	0,0,0,42,42,42,42,42,176,100,0,0	; ц
123604\$:. BYTE	0,0,0,160,110,104,104,104,376,202,0,0	; д
123617\$:. BYTE	0,0,0,74,102,102,176,2,174,0,0,0	; е
123632\$:. BYTE	0,0,0,20,174,222,222,222,174,20,20,0	; ф
123645\$:. BYTE	0,0,0,176,102,2,2,2,2,0,0,0	; г
123660\$:. BYTE	0,0,0,102,44,30,30,44,102,0,0,0	; х
123673\$:. BYTE	0,0,0,102,142,122,112,106,102,0,0,0	; и
123706\$:. BYTE	0,30,0,102,142,122,112,106,102,0,0,0	; й
123721\$:. BYTE	0,0,0,102,42,22,36,42,102,0,0,0	; к
123734\$:. BYTE	0,0,0,160,110,104,104,104,102,0,0,0	; л
123747\$:. BYTE	0,0,0,202,306,252,222,202,202,0,0,0	; м
123762\$:. BYTE	0,0,0,102,102,102,176,102,102,0,0,0	; н
123775\$:. BYTE	0,0,0,74,102,102,102,102,74,0,0,0	; о
124010\$:. BYTE	0,0,0,176,102,102,102,102,102,0,0,0	; п
124023\$:. BYTE	0,0,0,174,102,102,174,104,102,0,0,0	; я
124036\$:. BYTE	0,0,0,72,106,102,102,106,72,2,2,0	; р
124051\$:. BYTE	0,0,0,74,102,2,2,102,74,0,0,0	; с
124064\$:. BYTE	0,0,0,376,20,20,20,20,20,0,0,0	; т
124077\$:. BYTE	0,0,0,102,102,102,44,30,10,4,2,0	; у
124112\$:. BYTE	0,0,0,222,124,70,70,124,222,0,0,0	; ж
124125\$:. BYTE	0,0,0,76,102,76,102,102,76,0,0,0	; в
124140\$:. BYTE	0,0,0,2,2,76,102,102,76,0,0,0	; ь
124153\$:. BYTE	0,0,0,202,202,236,242,242,236,0,0,0	; ы
124166\$:. BYTE	0,0,0,74,100,70,100,100,74,0,0,0	; з
124201\$:. BYTE	0,0,0,222,222,222,222,222,376,0,0,0	; ш
124214\$:. BYTE	0,0,0,76,100,100,174,100,76,0,0,0	; э

124227\$:. BYTE	0, 0, 0, 222, 222, 222, 222, 222, 376, 200, 200	; Щ
124242\$:. BYTE	0, 0, 0, 102, 102, 102, 174, 100, 100, 0, 0	; Ч
124255\$:. BYTE	0, 0, 0, 7, 5, 174, 204, 204, 174, 0, 0	; Ъ
124270\$:. BYTE	0, 142, 222, 222, 222, 236, 222, 222, 142, 0, 0	; Ю
124303\$:. BYTE	0, 30, 44, 102, 102, 176, 102, 102, 102, 0, 0	; А
124316\$:. BYTE	0, 176, 2, 2, 76, 102, 102, 102, 76, 0, 0	; Б
124331\$:. BYTE	0, 102, 102, 102, 102, 102, 102, 376, 200, 0	; Ц
124344\$:. BYTE	0, 160, 110, 104, 104, 104, 104, 104, 376, 202, 0	; Д
124357\$:. BYTE	0, 176, 2, 2, 76, 2, 2, 2, 176, 0, 0	; Е
124372\$:. BYTE	0, 20, 174, 222, 222, 222, 222, 174, 20, 20, 0	; Ф
124405\$:. BYTE	0, 176, 102, 2, 2, 2, 2, 2, 0, 0	; Г
124420\$:. BYTE	0, 102, 102, 44, 30, 30, 44, 102, 102, 0, 0	; Х
124433\$:. BYTE	0, 102, 102, 142, 122, 112, 106, 102, 102, 0, 0	; И
124446\$:. BYTE	30, 102, 102, 142, 122, 112, 106, 102, 102, 0, 0	; Й
124461\$:. BYTE	0, 102, 42, 22, 16, 22, 42, 102, 202, 0, 0	; К
124474\$:. BYTE	0, 160, 110, 104, 104, 104, 104, 104, 102, 0, 0	; Л
124507\$:. BYTE	0, 202, 306, 252, 222, 202, 202, 202, 202, 0, 0	; М
124522\$:. BYTE	0, 102, 102, 102, 176, 102, 102, 102, 102, 0, 0	; Н
124535\$:. BYTE	0, 74, 102, 102, 102, 102, 102, 102, 74, 0, 0	; О
124550\$:. BYTE	0, 176, 102, 102, 102, 102, 102, 102, 102, 0, 0	; П
124563\$:. BYTE	0, 174, 102, 102, 102, 174, 110, 104, 102, 0, 0	; Я
124576\$:. BYTE	0, 76, 102, 102, 102, 76, 2, 2, 2, 0, 0	; Р
124611\$:. BYTE	0, 74, 102, 2, 2, 2, 2, 102, 74, 0, 0	; С
124624\$:. BYTE	0, 376, 222, 20, 20, 20, 20, 20, 20, 0, 0	; Т
124637\$:. BYTE	0, 102, 102, 102, 102, 174, 100, 102, 74, 0, 0	; У
124652\$:. BYTE	0, 222, 222, 124, 70, 124, 222, 222, 222, 0, 0	; Ж
124665\$:. BYTE	0, 76, 102, 102, 76, 102, 102, 102, 76, 0, 0	; В
124700\$:. BYTE	0, 2, 2, 2, 76, 102, 102, 102, 76, 0, 0	; Ь
124713\$:. BYTE	0, 202, 202, 202, 236, 242, 242, 242, 236, 0, 0	; Ы
124726\$:. BYTE	0, 74, 102, 100, 70, 100, 100, 102, 74, 0, 0	; З
124741\$:. BYTE	0, 222, 222, 222, 222, 222, 222, 222, 376, 0, 0	; Ш
124754\$:. BYTE	0, 74, 102, 200, 200, 370, 200, 102, 74, 0, 0	; Э
124767\$:. BYTE	0, 222, 222, 222, 222, 222, 222, 222, 376, 200, 200	; Щ
125002\$:. BYTE	0, 102, 102, 102, 102, 174, 100, 100, 100, 0, 0	; Ч
125015\$:. BYTE	0, 7, 5, 4, 174, 204, 204, 204, 174, 0, 0	; Ъ
. EVEN		

```

; *****
; *
; *
; *          ДРАЙВЕР КАНАЛА 2
; *
; *
; *          Адреса 125030 - 135435
; *
; *
; *****

```

```

; ***  ОСНОВНАЯ ПОДПРОГРАММА ОБСЛУЖИВАНИЯ КАНАЛА 2  ***
125030$:MOV    @#23200,R0      ; R0 = адрес массива параметров (МП) в ОЗУ ЦП
        CLC                      ; Преобр. для загрузки в рег. адреса планов
        ROR     R0
        MOV     R0,@#23200
        INC     R0              ; R0 = адрес типа устройства
        MOV     R0,@#177010
        MOVB    @#177014,R0    ; R0 = тип устройства
        MOV     R0,R5
        ASH     #-3,R0
        CMP     R0,#3          ; Тип устройства в пределах 0 - 37 ?
        BHI     125116$        ; Нет
        ASL     R0
        BNE     125104$        ; Обслуживается не дисковод
        TST     @#7044         ; Разрешено обращение к дисководу ?
        BMI     125116$        ; Нет
125104$:CALL    @125124$(R0)    ; Вызов соответствующей п/п обслуживания уст-ва

```

```

125110$:DEC      @#7062          ; Уменьшить значение флага вызова
        RETURN
125116$:CALL     125162$          ; Вызов п/п обслуживания кассеты ПЗУ
        BR       125110$
; Адреса п/п обслуживания устройств (дискковод, магнитофон, кассета ПЗУ, ОЗУ ПП)
125124$: .WORD   131176$,126236$,125162$,125644$
; П/п окончания обслуживания устройства
125134$:CLRB     @#23334          ; Сбросить код возврата (все нормально)
125140$:MOV      @#23200,@#177010
        MOV      @#23334,@#177014; Записать код возврата в ОЗУ ЦП
125154$:TST      @#177064          ; Восст. готовность канала 2 со стороны ЦП
        RETURN
; П/п обслуживания кассеты ПЗУ
125162$:JSR      R5,125560$        ; Чтение МП
        .WORD    5
        CMPB     #10,23335        ; Код функции = 10 (чтение) ?
        BEQ      125210$
125200$:MOVB     #2,23334          ; Код возврата = 2 (ошибочный МП)
        BR       125520$
125210$:MOV      23340,R1          ; R1 = адрес кассеты ПЗУ
        CMP      #60000,R1        ; Он превышает 60000 ?
        BLOS     125200$          ; Да
        BIC      #1,R1
        MOV      23342,R2          ; R2 = адрес буфера в ОЗУ ЦП
        CMP      #160000,R2       ; Он превышает 160000 ?
        BLOS     125200$          ; Да
        ASR      R2
        BIC      #100000,R2
        MOV      23344,R3          ; R3 = число считываемых слов
        BEQ      125514$          ; Оно равно нулю
        MOVB     23337,R0          ; R0 = номер кассеты ПЗУ
        MTPS     #200             ; Запретить прерывания
        MOV      @#4,R5           ; Сохранить значение 4 вектора в R5
        MOV      #125544$,@#4     ; Загрузить новое значение вектора
        BIC      #1,@#177054      ; Откл. сист. ПЗУ в диапазоне 100000-117777
        ASR      R0               ; Кассета N1 ?
        BCS      125320$          ; Нет
        BIC      #10,@#177054     ; Выбор кассеты N1
        BR       125326$
125320$:BIS      #10,@#177054      ; Выбор кассеты N2
125326$:BIC      #6,@#177054      ; Ни одного банка не выбрано
        CMP      R1,#17776        ; Адрес кассеты меньше 20000 ?
        BHI      125354$          ; Нет
        MOV      #2,R4
        BIS      R4,@#177054      ; Выбор банка N1
        BR       125406$
125354$:CMP      R1,#37776        ; Адрес кассеты меньше 40000 ?
        BHI      125376$          ; Нет
        BIS      #4,@#177054      ; Выбор банка N2
        MOV      #1,R4
        BR       125406$
125376$:BIS      #6,@#177054      ; Выбор банка N3
        CLR      R4
125406$:BIC      #60000,R1
        BIS      #100000,R1
        MOV      #120000,R0
        SUB      R1,R0            ; Вычисление граничных значений
        ASR      R0
125426$:CMP      R3,R0            ; Область перекачки содержится в одном банке ?
        BHI      125440$          ; Нет
        MOV      R3,R0
        CLR      R3
        BR       125442$
125440$:SUB      R0,R3            ; Вычесть из числа запрошенных слов

```



```

125442$:MOV      R2,@#177010      ; Цикл перекачки
        MOV      (R1)+,@#177014
        INC      R2
        CMP      #70000,R2      ; Произшел выход за диапазон ОЗУ ЦП (160000) ?
        BLOS     125200$        ; Да
        SOB      R0,125442$
        TST      R3              ; Перекачка закончена ?
        BEQ      125514$        ; Да
        DEC      R4              ; Переход к следующему банку
        BMI      125200$        ; Переход за границу кассеты ПЗУ
        ADD      #2,@#177054    ; Установить следующий банк
        MOV      #100000,R1
        MOV      #10000,R0
        BR       125426$
125514$:CLRB     23334           ; Очистить код возврата (все нормально)
125520$:BIS      #1,@#177054    ; Подключить системное ПЗУ
        MOV      R5,@#4         ; Восстановить 4 вектор
        MTPS     #0             ; Разрешить прерывания
        CALL     125140$        ; Завершить обслуживание
        RETURN
; П/п обработки прерывания 4 при чтении с кассеты ПЗУ
125544$:MOVB     #1,23334        ; Код возврата = 1 (отсутствует банк ПЗУ)
        TST      (SP)+          ; Вычеркнуть из стека значения PC и PS
        TST      (SP)+
        BR       125520$
; П/п чтения МП из ОЗУ ЦП
125560$:MOV      (R5)+,R0        ; R0 = число слов для чтения
        MOV      #23334,R1      ; R1 = адрес МП в ОЗУ ПП
        MOV      @#23200,@#177010
        BR       125602$
125576$:INC      @#177010        ; Цикл чтения
125602$:MOV      @#177014,(R1)+
        SOB      R0,125576$
        RTS      R5
; П/п записи МП в ОЗУ ЦП
125612$:MOV      (R5)+,R0        ; R0 = число слов для записи
        MOV      #23334,R1      ; R1 = адрес МП в ОЗУ ПП
        MOV      @#23200,@#177010
        BR       125634$
125630$:INC      @#177010        ; Цикл записи
125634$:MOV      (R1)+,@#177014
        SOB      R0,125630$
        RTS      R5
; П/п обслуживания функций работы с памятью ПП
125644$:JSR      R5,125560$      ; Чтение МП
        .WORD    5
        CLRB     @#23334        ; Код возврата = 0
        CMP      R5,#32         ; Тип устройства = 32 ?
        BNE      125720$        ; Нет
        MOVB     @#23335,R0      ; R0 = код функции
        MOV      R0,R5
        ASH      #-3,R0
        CMP      R0,#3          ; Код функции в пределах 0 - 37 ?
        BHI      125720$        ; Нет
        ASL      R0
        ADD      R0,PC          ; Переход по соответствующему адресу
        BR       125730$        ; Запрос и освобождение блоков памяти
        BR       126024$        ; Чтение памяти ПП
        BR       126050$        ; Запись в память ПП
        BR       126074$        ; Запуск п/п пользователя
125720$:JMP      125162$
125724$:JMP      125154$
; Запрос и освобождение блоков памяти
125730$:MOV      @#23340,R1      ; R1 = начальный адрес в памяти ПП

```

```

MOV    @#23342,R0      ; R0 = размер блока в словах
ASL    R0
ASR    R5              ; Функция запроса блока памяти ?
BCC    126012$         ; Нет
CALL   176132$         ; Запросить область памяти
ASR    R0
CMP    R0,@#23342      ; Полученное значение >= запрошенного ?
BHS    125770$         ; Да
MOVB   #377,@#23334    ; Код возврата = 377 (мало памяти)
125770$:MOV R0,@#23342  ; Возвратить объем полученной памяти
MOV    R1,@#23340      ; Возвратить адрес блока
JSR    R5,125612$      ; Записать МП в ОЗУ ЦП
.WORD  4
JMP    125154$
126012$:CALL 176300$    ; Освободить блок памяти
BMI    125720$
JMP    125140$
; Чтение памяти ПП
126024$:CALL 126110$    ; Получение адресов ЦП, ПП и числа слов
BEQ    125720$         ; Число слов равно нулю
BR     126040$
126034$:INC @#177010    ; Цикл чтения
126040$:MOV (R1)+,@#177014
SOB    R0,126034$
BR     125724$
; Запись в память ПП
126050$:CALL 126110$    ; Получение адресов ЦП, ПП и числа слов
BEQ    125720$         ; Число слов равно нулю
BR     126064$
126060$:INC @#177010    ; Цикл записи
126064$:MOV @#177014,(R1)+
SOB    R0,126060$
BR     125724$
; Запуск п/п пользователя
126074$:MOV @#23340,@#7140 ; Сохранение адреса запуска
INC    @#7076          ; Установить флаг запуска п/п пользователя
BR     125724$
; П/п получения адресов блоков памяти ЦП, ПП и числа слов для обмена
126110$:MOV @#23340,R1   ; R1 = адрес памяти ПП
MOV    @#23342,R0       ; R0 = адрес памяти ЦП
CLC
ROR    R0
MOV    R0,@#177010      ; Загрузка адреса в регистр адреса планов
MOV    @#23344,R0       ; R0 = число слов для обмена
RETURN
; *** МАГНИТОФОН ***
; П/п установки начальных значений
126136$:CLR R0
MOV    #23304,R1
MOV    #14,R2
126150$:MOV R0,(R1)+    ; Очистка области ОЗУ п/п работы с магнитофоном
SOB    R2,126150$
MOV    #320,23304      ; Длина бита "0"
MOV    #150,23306      ; Длина бита "1"
MOV    #17500,23310    ; Число битов "1" при записи пилот-тона загол.
MOV    #3720,23312     ; Число битов "1" при записи пилот-тона данных
MOV    #17540,23314    ; Число битов "1" при чтении пилот-тона загол.
MOV    #1750,23316     ; Число битов "1" при чтении пилот-тона данных
MOV    #1750,23320
MOV    #64,23322       ; Разброс времени между перепадами лог. уровня
RETURN
; Основная п/п обслуживания магнитофона
126236$:JSR R5,125560$  ; Чтение массива параметров
.WORD  27

```

```

CLR      R5
126246$: SOB      R5, 126246$      ; Задержка
MTPS     #200      ; Запретить прерывания
CALL     176376$    ; Останов программируемого таймера
CALL     130672$    ; Ожидание готовности программируемого таймера
MOV      #127734$, @#304      ; Настройка вектора прерывания прогр. таймера
MOV      #200, @#306
MOV      #130570$, @#310      ; Настройка вектора прерывания по внеш. событию
MOV      #200, @#312
CLR      @#177700      ; Запрет прерываний от клавиатуры
BIS      #400, @#177054      ; Запрет прерываний от сет. таймера в канале ПП
BIC      #200, @#177716      ; Выключить генератор звукового сигнала
BIC      #107, @#177066      ; Запретить прерывания от каналов связи с ЦП
MOVB     23335, R0      ; R0 = код функции
BIC      #177600, R0
ASL      R0
CALL     @126502$(R0)      ; Вызов соответствующей п/п
MOVB     R0, 23334      ; Код возврата
BIC      #14, @#177716      ; Запрет приема информации по линии вн. событие
CALL     130672$
MOV      23200, R4      ; R4 = адрес МП в ОЗУ ЦП
ADD      #14, R4
MOV      R4, @#177010
MOV      #23364, R4      ; R4 = адрес МП в ОЗУ ПП (имя прочит. файла)
MOV      #13, R2
126422$: MOV      (R4)+, @#177014      ; Пересылка
INC      @#177010
SOB      R2, 126422$
BIC      #400, @#177054      ; Разрешить прерывания от таймера в канале ПП
MOV      #100, @#177700      ; Разрешить прерывания от клавиатуры
BIS      #200, @#177716      ; Включить генератор звукового сигнала
BIS      #107, @#177066      ; Разрешить прерывания от каналов связи с ЦП
CALL     125140$      ; Переслать код возврата в ОЗУ ЦП
CALL     176422$      ; Запрет прерываний от прогр. таймера
MTPS     #0      ; Разрешить прерывания
RETURN

; Адреса п/п обслуживания работы с магнитофоном
126502$: .WORD     126546$, 126554$, 126632$, 126546$
        .WORD     126546$, 126546$, 126546$, 126546$
        .WORD     127102$, 127110$, 126546$, 126546$
        .WORD     127116$, 127132$, 127044$, 126546$
        .WORD     126714$, 126662$
126546$: MOV      #10, R0      ; R0 = код возврата (неверный код функции)
RETURN

; П/п установки параметров одинарной плотности записи
126554$: MOV      #320, 23304      ; Длина бита "0"
        MOV      #150, 23306      ; Длина бита "1"
        MOV      #17500, 23310      ; Число битов "1" перед заголовком
        MOV      #3720, 23312      ; Число битов "1" перед зоной данных
        MOV      #17540, 23314      ; Число перепадов уровня при чтении (загол.)
        MOV      #1750, 23316      ; Число перепадов уровня при чтении (данные)
        MOV      #1750, 23320
        CLR      R0
        RETURN

; П/п установки параметров двойной плотности записи
126632$: MOV      #150, 23304      ; Длина бита "0"
        MOV      #64, 23306      ; Длина бита "1"
        NOP
        MOV      #3720, 23320
        CLR      R0
        RETURN

; П/п записи блока данных на магнитную ленту
126662$: MOV      SP, 23332      ; Сохранить значение указателя стека
        MOV      23360, R4      ; R4 = адрес буфера в ОЗУ ЦП

```

```

CLC
ROR    R4
MOV    R4, @177010
CALL   127476$ ; Подготовка и пуск программируемого таймера
MOV    #200, R2 ; R2 = число записываемых слов
BR     126776$
; П/п записи файла с заголовком
126714$:MOV SP, 23332 ; Сохранить значение указателя стека
MOV     23360, R4 ; R4 = адрес буфера в ОЗУ ЦП
CLC
ROR     R4
MOV     R4, @177010
CALL    127476$ ; Подготовка и пуск программируемого таймера
CALL    127536$ ; Запись пилот-тона перед заголовком
CALL    127560$ ; Запись имени файла
MOV     23362, R0 ; R0 = число записываемых слов
MOV     R0, R2 ; R2 = число записываемых слов
TST     R2 ; Оно равно нулю ?
BEQ     127040$
CALL    127602$ ; Запись числа слов в файл
MOV     23360, R0 ; R0 = адрес буфера в ОЗУ ЦП
CALL    127602$ ; Запись адреса буфера в файл
126776$:CALL 127544$ ; Запись пилот-тона перед зоной данных
CLR     R4 ; Контрольная сумма = 0
127004$:MOV @177014, R0 ; R0 = слово данных
INC     @177010 ; Переход к следующему адресу
ADD     R0, R4 ; Подсчет контрольной суммы файла
ADC     R4
CALL    127602$ ; Запись слова данных
SOB     R2, 127004$ ; Цикл по R2 (число записываемых слов)
MOV     R4, R0 ; R0 = контрольная сумма
CALL    127602$ ; Запись контрольной суммы
CALL    127664$ ; Запись бита "1"
127040$:CLR R0
RETURN
; П/п чтения блока данных
127044$:MOV SP, 23332 ; Сохранить значение указателя стека
MOV     #200, R2 ; R2 = число считываемых слов
MOV     23360, R0 ; R0 = адрес буфера в ОЗУ ЦП
CLC
ROR     R0
MOV     R0, @177010
CALL    130020$ ; Подготовка и пуск программируемого таймера
CALL    130156$ ; Поиск пилот-тона перед зоной данных
BR     127372$
; Точка входа при чтении файла с заголовком
127102$:CLR 23330 ; Признак реального чтения файла
BR     127124$
; Точка входа при чтении файла с пропуском заголовка
127110$:CLR 23330 ; Признак реального чтения файла
BR     127140$
; Точка входа при фиктивном чтении файла с заголовком
127116$:MOV #4, 23330 ; Признак фиктивного чтения
127124$:CLR 23326 ; Признак проверки заголовка
BR     127146$
; Точка входа при фиктивном чтении файла с пропуском заголовка
127132$:MOV #4, 23330 ; Признак фиктивного чтения
127140$:MOV #22, 23326 ; Признак пропуска заголовка
127146$:MOV SP, 23332 ; Сохранить значение указателя стека
CALL    130020$ ; Подготовка и пуск программируемого таймера
CALL    130060$ ; Поиск пилот-тона перед заголовком
CALL    130254$ ; Автоподстройка по пилот-тону
CALL    130372$ ; Чтение заголовка
BIT     #1, R1 ; Тип устройства ?

```

	BEQ	127220\$; MSX
	CALL	130436\$; Чтение длины файла в словах
	MOV	R0, 23406	; Сохранить ее в МП
	CALL	130436\$; Чтение адреса загрузки файла
	MOV	R0, 23404	; Сохранить его в МП
127220\$:	MOV	23320, R3	
	ADD	23326, PC	; Пропустить проверку заголовка (на 127252\$)?
	MOV	#10, R2	; R2 = длина заголовка в словах
	MOV	#23364, R4	; R4 = адрес прочитанного заголовка
	MOV	#23340, R0	; R0 = адрес заданного заголовка
127244\$:	CMP	(R0)+, (R4)+	; Цикл сравнения
	BNE	127462\$; Не равны
	SOB	R2, 127244\$	
127252\$:	BIT	#1, R1	; Тип устройства ?
	BNE	127312	; УКНЦ
	MOV	#200, R2	; R2 = число записываемых слов
	MOV	23360, R0	; R0 = адрес буфера в ОЗУ ЦП
	CLC		
	ROR	R0	
	MOV	R0, @#177010	
	MOV	#10, R3	
	CALL	130156\$; Поиск пилот-тона перед зоной данных
	BR	127372\$	
127312\$:	MOV	23362, R0	; R0 = число считываемых слов
	TST	R0	; Оно превышает 32767 ?
	BMI	127330\$; Да
	CMP	23406, R0	; Число запрошенных слов превышает длину файла?
	BHI	127334\$; Нет
127330\$:	MOV	23406, R0	; Используем длину файла
127334\$:	MOV	R0, R2	; R2 = число считываемых слов
	MOV	23404, 23410	
	MOV	23360, R0	; R0 = адрес буфера в ОЗУ ЦП
	CLC		
	ROR	R0	
	BCC	127366\$; Четный адрес
	MOV	23404, R0	; Используется считанный адрес
	CLC		
	ROR	R0	
127366\$:	MOV	R0, @#177010	
127372\$:	TST	R2	; Число считываемых слов равно нулю ?
	BEQ	127452\$; Да
	CLR	R4	; Контрольная сумма = 0
	CALL	130336\$; Поиск и чтение первого слова данных
	BR	127412\$	
127406\$:	CALL	130436\$; Чтение слова
127412\$:	ADD	23330, PC	; Фиктивное чтение (на 127422\$) ?
	MOV	R0, @#177014	; Пересылка в ОЗУ ЦП
127422\$:	INC	@#177010	; Переход к следующему адресу
	ADD	R0, R4	; Подсчет контрольной суммы
	ADC	R4	
	SOB	R2, 127406\$	
	BIT	#1, R1	; Тип устройства
	BEQ	127452\$; MSX
	CALL	130436\$; Чтение контрольной суммы
	CMP	R0, R4	; Считанная КС равна подсчитанной ?
	BNE	127470\$	
127452\$:	CLR	R0	
127454\$:	MTPS	#200	
	RETURN		
127462\$:	MOV	#1, R0	; R0 = не совпадает заголовок
	BR	127454\$	
127470\$:	MOV	#12, R0	; R0 = не совпадает контрольная сумма
	BR	127454\$	

; П/п начальной подготовки перед записью

```
127476$:BIC      #14,@#177716 ; Блокировка приема информации с МЛ
      CLR      @#177712      ; Очистить счетчик прогр. таймера
127510$:TST      @#177714      ; Счетчик очищен ?
      BNE      127510$      ; Нет
      MOV      #101,@#177710 ; Пуск таймера и разрешение прерываний
      MOV      #4,R3        ; R3 = число записываемых импульсов
      MTPS     #0          ; Разрешить прерывания
      RETURN
```

; Точка входа при записи пилот-тона перед заголовком

```
127536$:MOV      23310,R4      ; R4 = число битов "1" в пилот-тоне
      BR       127550$
127544$:MOV      23312,R4      ; R4 = число "1" в пилот-тоне перед данными
127550$:CALL     127664$      ; Запись бита "1"
      SOB      R4,127550$    ; Цикл по R4
      RETURN
```

; П/п записи имени файла

```
127560$:MOV      #10,R2        ; R2 = размер имени файла в словах
      MOV      #23340,R4      ; R4 = адрес начала имени файла
127570$:MOV      (R4)+,R0      ; R0 = очередное слово из имени файла
      CALL     127602$      ; Запись слова
      SOB      R2,127570$    ; Цикл по R2
      RETURN
```

; П/п записи слова на МЛ

```
127602$:CALL     127616$      ; Запись младшего байта
      SWAB     R0
      CALL     127616$      ; Запись старшего байта
      RETURN
```

; П/п записи байта на МЛ

```
; /-----\
; !                СТРУКТУРА БАЙТА ПРИ ЗАПИСИ НА МЛ                !
; !                !                !                !                !
; !    бит "0", информационные биты (0) -- (7), бит "1", бит "1"    !
; \-----/
```

```
127616$:CALL     127660$      ; Запись бита "0"
      SEC      ; Признак окончания байта
127624$:RORB     R0          ; Выдвинуть очередной бит
      BEQ      127646$      ; Байт окончился
      BCC      127640$      ; Выдвинут бит "0"
      CALL     127664$      ; Запись бита "1"
      BR       127624$
127640$:CALL     127660$      ; Запись бита "0"
      BR       127624$
127646$:CALL     127664$      ; Запись концевика (два бита "1")
      CALL     127664$
      RETURN
```

; Точка входа при записи бита "0"

```
127660$:CLR      R5          ; Признак записи бита "0"
      BR       127670$
```

; Точка входа при записи бита "1"

```
127664$:MOV      #14,R5      ; Признак записи бита "1"
127670$:CALL     130636$      ; Проверка нажатия "СТОП"
      TST      R3          ; Запись бита окончена ?
      BNE      127670$      ; Нет
      ADD      R5,PC        ; Если запись "1", на 127716$
      MOV      23304,@#177712 ; Занести длительность "0" в буфер таймера
      MOV      #2,R3        ; R3 = число записываемых импульсов
      BR       127730$
127716$:MOV      23306,@#177712 ; Занести длительность "1" в буфер таймера
      MOV      #4,R3        ; R3 = число записываемых импульсов
127730$:CLC
      RETURN
```

; П/п обработки прерывания от программируемого таймера

```
127734$:BIS      #2,@#177716 ; Установить высокий уровень при записи на МЛ
```

```

MOV      #127752$,@#304 ; Новое значения вектора прерывания
BR       127766$
127752$:BIC      #2,@#177716 ; Установить низкий уровень при записи на МЛ
MOV      #127734$,@#304 ; Новое значение вектора прерывания
127766$:TST      @#177714 ; Сбросить готовность таймера
TSTB     @#177710 ; Готовность сброшена ?
BMI      127766$ ; Нет
DEC      R3 ; Уменьшить счетчик записываемых импульсов
BMI      130006$ ; Он отрицательный
RTI
130006$:MOV      #14,R0 ; R0 = ошибка выдержки времени при записи
MOV      23332,SP
RETURN
; П/п начальной подготовки перед чтением
130020$:CLR      @#177710 ; Остановить программируемый таймер
MOV      23336,R1 ; R1 = тип устройства (УКНЦ или MSX)
CLR      @#177712 ; Очистить буфер таймера
130034$:TST      @#177714 ; Буфер очищен ?
BNE      130034$ ; Нет
BIS      #14,@#177716 ; Разрешить прием информации с МЛ
MOV      #21,@#177710 ; Пуск таймера и разрешение прерывания при
; изменении информации на линии МЛ
RETURN
; П/п поиска пилот-тона перед заголовком
130060$:MTPS     #200 ; Запретить прерывания
130064$:MOV      23314,R2 ; R2 = число перепадов уровня при поиске загол.
130070$:MOV      R5,R4 ; R4 = длина предыдущего интервала
130072$:CALL     130636$ ; Проверка нажатия "СТОП"
BIT      #40,@#177710 ; Перепад произошел ?
BEQ      130072$ ; Нет
MOV      @#177714,R5 ; R5 = длина текущего интервала
130112$:TST      @#177714 ; Сбросить готовность таймера
BIT      #250,@#177710 ; Готовность сброшена ?
BNE      130112$ ; Нет
SUB      R5,R4 ; Вычислить разность между длинами интервалов
BPL      130134$
NEG      R4
130134$:CMP      R4,23322 ; Разброс длин в норме ?
BHI      130064$ ; Нет
SOB      R2,130070$ ; Цикл по R2
MOV      #10,R3
MTPS     #0
RETURN
; П/п поиска пилот-тона перед зоной данных
130156$:MTPS     #200 ; Запретить прерывания
130162$:MOV      23316,R0 ; R0 = число перепадов уровня при поиске загол.
130166$:MOV      R5,R4 ; R4 = длина предыдущего интервала
130170$:CALL     130636$ ; Проверка нажатия "СТОП"
BIT      #40,@#177710 ; Перепад произошел ?
BEQ      130170$ ; Нет
MOV      @#177714,R5 ; R5 = длина текущего интервала
130210$:TST      @#177714 ; Сбросить готовность таймера
BIT      #250,@#177710 ; Готовность сброшена ?
BNE      130210$ ; Нет
SUB      R5,R4 ; Вычислить разность между длинами интервалов
BPL      130232$
NEG      R4
130232$:CMP      R4,23322 ; Разброс длин в норме ?
BHI      130162$ ; Нет
SOB      R0,130166$ ; Цикл по R0
MOV      #10,R3
MTPS     #0
RETURN

```

; П/п вычисления усредненного значения длины бита (между "0" и "1")

```
130254$:MOV      R1,R4          ; Сохранить значение R1 в R4
        MOV      #10,R0        ; R0 = число читаемых импульсов
        CLR      R1
        CLR      R3
130266$:CALL     130530$        ; Чтение бита
        BIS      #170000,R5    ; Преобразование в отрицательное число
        SUB      R5,R1         ; Прибавить к общей длине
        SOB      R0,130266$    ; Цикл по R0
        MOV      #10,R3
        DIV      #10,R0        ; Вычисление среднего значения длины "1"
        MOV      R0,R1         ; R1 = длина "1"
        ASR      R0            ; R0 = длина "1" / 2
        ADD      R1,R0         ; R0 = среднее между длиной "1" и "0"
        NEG      R0            ; Преобразование в отрицательное число (12 бит)
        BIC      #170000,R0
        MOV      R0,23324      ; Сохранение его в памяти
        MOV      R4,R1         ; Восстановление R1
        RETURN
```

; П/п поиска данных (определяются по биту "0")

```
130336$:CLR      R5            ; Очередной бит не прочитан
130340$:CALL     130636$        ; Проверка нажатия "СТОП"
        TST      R5            ; Бит прочитан ?
        BEQ      130340$        ; Нет
        INC      R3            ; Увеличить счетчик импульсов
        CMP      R5,23324      ; Это бит "0" ?
        BHS      130336$        ; Нет
        INC      R3
        INC      R3
        CALL     130422$        ; Дочитать слово
        RETURN
```

; П/п чтения имени файла

```
130372$:MOV      #10,R2        ; R2 = длина имени файла в словах
        MOV      #23364,R4     ; R4 = адрес буфера
        CALL     130336$        ; Поиск и чтение первого слова
        BR       130414$
130410$:CALL     130436$        ; Чтение очередного слова
130414$:MOV      R0,(R4)+       ; Поместить в буфер
        SOB      R2,130410$    ; Цикл по R2
        RETURN
```

; П/п дочитывания слова

```
130422$:CLRB     R0
        BIS      #200,R0       ; Признак окончания чтения байта
        CALL     130470$        ; Дочитать байт
        BR       130442$
```

; П/п чтения слова

```
130436$:CALL     130454$        ; Прочитать младший байт
130442$:SWAB     R0
        CALL     130454$        ; Прочитать старший байт
        SWAB     R0
        RETURN
```

; П/п чтения байта

```
130454$:CLRB     R0
        BIS      #200,R0       ; Установить признак окончания
        CALL     130530$        ; Прочитать бит
        BCS      130516$        ; Это не "0" (ошибка)
130470$:CALL     130530$        ; Прочитать бит
        RORB     R0            ; "Задвинуть" бит
        BCC      130470$        ; Пока не конец
        CALL     130530$        ; Прочитать бит
        BCC      130516$        ; Это не "1" (ошибка)
        CALL     130530$        ; Прочитать бит
        BCC      130516$        ; Это не "1" (ошибка)
        RETURN
```



```

130516$:MOV      #11,R0          ; R0 = ошибка чтения
        MOV      23332,SP
        RETURN
; П/п чтения бита
130530$:CLR      R5              ; Бит еще не прочитан
130532$:CALL     130636$         ; Проверка нажатия "СТОП"
        TST      R5              ; Бит прочитан ?
        BEQ      130532$         ; Нет
        CMP      R5,23324        ; Это "0" ?
        BLO      130560$         ; Да
        MOV      #3,R3
        SEC                      ; Признак бита "1"
        BR       130566$
130560$:MOV      #1,R3
        CLC                      ; Признак бита "0"
130566$:RETURN
; П/п обработки прерывания по линии МЛ при изменении уровня сигнала
130570$:TST      R3              ; Счетчик равен нулю ?
        BNE      130604$         ; Нет
        TST      R5              ; Обработка предыдущего бита окончена ?
        BNE      130624$         ; Нет
        MOV      @#177714,R5     ; R5 = длина импульса
130604$:TST      @#177714        ; Сбросить готовность таймера
        BIT      #250,@#177710   ; Готовность сброшена ?
        BNE      130604$         ; Нет
        DEC      R3              ; Уменьшить счетчик импульсов
        RTI
130624$:MOV      #13,R0          ; R0 = ошибка выдержки времени при чтении
        MOV      23332,SP
        RETURN
; П/п проверки нажатия клавиши "СТОП"
130636$:TSTB     @#177700        ; Клавиша нажата ?
        BPL      130670$         ; Нет
        CMP      #4,@#177702     ; Это "СТОП" ?
        BNE      130670$         ; Нет
        MOV      23332,SP
        MOV      #377,R0         ; R0 = нажата клавиша "СТОП"
        MTPS     #200            ; Запретить прерывания
130670$:RETURN
; П/п выключения программируемого таймера
130672$:CLR      @#177710        ; Выключить таймер (режим "ОСТАНОВ")
        TST      @#177714        ; Сбросить готовность таймера
        TST      @#177710        ; Готовность сброшена ?
        BNE      130672$         ; Нет
        RETURN
; *** ДИСКОВОД ***
; П/п установки начальных значений
130712$:TST      @#177132        ; Проверка присутствия контроллера дисковода
        CLR      23204           ; Очистить управляющее слово
        MOV      23204,@#177130  ; Инициализация контроллера
        CLRB     23334
        CLR      23220
        CLR      23222
        BIT      #10000,@#177704 ; Подключен 80-дорожечный дисковод ?
        BEQ      131030$         ; Нет
; Параметры для 80-дорожечного дисковода
        MOV      #620,23216      ; Время разгона двигателя
        MOV      #23260,R1       ; R1 = адрес буфера времен шага
        MOV      #3,R0           ; R0 = время шага
        CALL     131164$         ; Установить для всех четырех накопителей
        MOV      #23264,R1       ; R1 = адрес буфера времен загрузки головки
        MOV      #44,R0          ; R0 = время загрузки головки
        CALL     131164$         ; Установить для всех четырех накопителей
        MOV      #50,23226      ; Номер дорожки, где вкл. прекооррекция 250 нс

```

```

        MOV    #120,23227      ; Номер дорожки, где вкл. прекоррекция 500 нс
        BR     131102$
; Параметры для 40-дорожечного дисковогода
131030$:MOV    #1750,23216      ; Время разгона двигателя
        MOV    #23260,R1       ; R1 = адрес буфера времен шага
        MOV    #24,R0          ; R0 = время шага
        CALL   131164$         ; Установить для всех четырех накопителей
        MOV    #23264,R1       ; R1 = адрес буфера времен загрузки головки
        MOV    #50,R0          ; R0 = время загрузки головки
        CALL   131164$         ; Установить для всех четырех накопителей
        MOV    #10,23226       ; Номер дорожки, где вкл. прекоррекция 250 нс
        MOV    #36,23227       ; Номер дорожки, где вкл. прекоррекция 500 нс
131102$:MOV    #23270,R1       ; R1 = адрес буфера числа секторов в дорожке
        MOV    #12,R0          ; R0 = число секторов в дорожке
        CALL   131164$         ; Установить для всех четырех накопителей
        MOV    #23274,R1       ; R1 = адрес буфера параметров SKIP (REGIM)
        MOV    #1,R0           ; R0 = параметр SKIP
        CALL   131164$         ; Установить для всех четырех накопителей
        MOV    #2,23250        ; Текущий формат сектора - 512 байт
        MOV    #-1,23337
        MOV    #-1,23254       ; Неопределенный номер текущих дорожек
        MOV    #-1,23256
        RETURN
131164$:MOV    #4,R2           ; R2 = максимальное число накопителей
131170$:MOVB   R0,(R1)+        ; Установить параметр
        SOB    R2,131170$
        RETURN
; Основная п/п управления накопителем
131176$:JSR    R5,125560$      ; Чтение МП
        .WORD  5
        CALL   176376$         ; Останов программируемого таймера
        JSR    R5,134210$      ; Запуск таймера с последующим вызовом п/п
        .WORD  10,131224$,107 ; по адресу 131224$
        RETURN
131224$:CALL   134242$         ; Сброс готовности программируемого таймера
        CLRB   @#177710        ; Перевести таймер в режим "ОСТАНОВ"
        CALL   134316$         ; Сохранить значения РОН
        BIC    #175754,23204    ; Сброс в УС прекоррекции,номера стороны ...
        BICB   #300,23335       ; Очистка старших битов кода функции
        BEQ    131516$         ; Код функции = 0 (ошибка)
        CMPB   #10,23335        ; Код функции >= 10(8) ?
        BLOS   131272$         ; Да
        JMP    135214$         ; Код функции в пределе 1 -- 7
131272$:TSTB   23337           ; Какая сторона запрошена ?
        BPL    131316$         ; Низ
        BIS    #40,23204        ; Установить "ВЕРХ" в УС
        MOVB   #1,23212        ; Текущая сторона - верх
        BR     131330$
131316$:BIC    #40,23204        ; Установить "НИЗ" в УС
        CLRB   23212           ; Текущая сторона - низ
131330$:BICB   #374,23337       ; Оставить биты номера устройства
        MOVB   23340,23213      ; Переслать номер запрашиваемой дорожки
        CLR    R1
        MOVB   23337,R1         ; R1 = номер устройства
        MOVB   23260(R1),23220 ; Текущее время шага
        MOVB   23264(R1),23222 ; Текущее время загрузки головки
        MOVB   23274(R1),23300 ; Текущий параметр SKIP
        MOVB   23270(R1),23301 ; Текущее число секторов в дорожке
        TSTB   23341           ; Номер запрашиваемого сектора отрицательный ?
        BMI    131516$         ; Да
        CMPB   23341,23301      ; Номер сектора превышает лимит ?
        BHI    131516$         ; Да
        CMPB   #5,23301        ; В дорожке 5 секторов ?
        BEQ    131440$         ; Да

```

```

MOV      #2,23250      ; Текущий формат сектора - 512 байт
BR       131446$
131440$:MOV      #3,23250      ; Текущий формат сектора - 1024 байт
131446$:MOV      23250,R0      ; R0 = текущий формат сектора
MOV      135426$(R0),23302; (23302) = размер межсекторного промежутка
ASL      R0
MOV      135412$(R0),23252; (23252) = число слов в секторе
CMPB     #30,23335      ; Код функции < 30
BHI      131502$        ; Да
BR       131542$
131502$:CLC           ; Преобразование адреса для загрузки в 177010
ROR      23342
MOV      23344,R0      ; R0 = число слов для чтения
BPL      131530$        ; Оно < 32768 (в норме)
131516$:MOVB     #11,23334    ; Ошибочный массив параметров
JMP      134422$
131530$:ADD      23342,R0      ; R0 = число слов + начальный адрес
CMP      #70000,R0      ; Конец буфера выходит за границу 56 КВ ?
BLO      131516$        ; Да
131542$:MOVB     23337,R1      ; R1 = номер устройства
MOV      R1,23206
ADD      #23254,23206      ; (23206) = указатель на текущую дорожку
JSR      R5,134210$      ; Запуск таймера с последующим вызовом п/п
.WORD    101,131710$,107    ; по адресу 131710$
BIT      #2000,23204      ; Двигатель накопителя еще работает ?
BNE      131620$        ; Да
BIS      #20,23204        ; Установить бит запуска мотора в УС
MOV      23216,23224      ; (23224) = время разгона двигателя
BR       131652$
131620$:MOVB     23204,R0      ; R0 = номер выбранного устройства
BICB     #374,R0          ; R0 = номер выбранного устройства
CMPB     135406$(R1),R0    ; Номера выбранного и запрошенного уст-в равны?
BNE      131644$        ; Нет
CALL     134356$          ; Восстановить PОН
BR       131726$
131644$:MOV      23222,23224    ; (23224) = время загрузки головки
131652$:BIC      #3,23204
BISB     135406$(R1),23204; Установить биты номера устройства в УС
BIS      #2000,23204      ; Установить бит выбора накопителя
MOV      23204,@#177130    ; Запуск мотора
CALL     134356$          ; Восстановить PОН
RTI
131710$:CALL     134242$      ; Сброс готовности программируемого таймера
DEC      23224            ; Уменьшить значение счетчика (разгон мотора)
BNE      131724$        ; Он еще не равен нулю
BR       131726$
131724$:RTI
131726$:JSR      R5,134210$    ; Запуск таймера с последующим вызовом п/п
.WORD    101,132074$,107    ; по адресу 132074$
TSTB     @23206           ; Текущий номер дорожки определен ?
BPL      132112$        ; Да
MOVB     #200,@23206      ; Примем в качестве номера дорожки - 128
BIC      #100,23204      ; Установить в УС перемещение от центра
MOV      23204,@#177130    ; Сообщить об этом контроллеру
BIS      #200,23204      ; Установить в УС бит шага
131776$:BIT      #1,@#177130 ; Головка находится на нулевой дорожке ?
BEQ      132022$        ; Нет
BIC      #200,23204      ; Сбросить в УС бит шага
CLRB     @23206          ; Текущая дорожка - нулевая
BR       132112$
132022$:DECB     @23206      ; Уменьшить номер текущей дорожки
BNE      132056$        ; Пока не 0
BIC      #200,23204      ; Сбросить в УС бит шага
MOVB     #377,@23206      ; Номер дорожки - неопределенный

```

	MOVB	#7,23334	; Ошибка - нулевая дорожка не обнаружена
	JMP	134422\$	
132056\$:	MOV	23204,@#177130	; Сделать шаг назад
	MOV	23220,23224	; (23224) = время шага
	RTI		
132074\$:	CALL	134242\$; Сбросить готовность программируемого таймера
	DEC	23224	; Уменьшить значение счетчика (время шага)
	BNE	132110\$; Он не равен нулю
	BR	131776\$	
132110\$:	RTI		
132112\$:	CMPB	@23206,23340	; Сравнить номера текущей и запрошенной дорожек
	BEQ	132346\$; Равны
	BHI	132164\$; Больше
	INCB	@23206	; Увеличить номер текущей дорожки
	BMI	132152\$; Он > 127 (ошибка)
	BIT	#100,23204	; В УС установлен бит перемещения к центру
	BNE	132246\$; Да
	BIS	#100,23204	; Установить в УС бит перемещения к центру
	BR	132224\$	
132152\$:	MOVB	#10,23334	; Ошибка - дорожка не обнаружена
	JMP	134422\$	
132164\$:	DECB	@23206	; Уменьшить номер текущей дорожки
	BIT	#1,@#177130	; Головка находится на нулевой дорожке ?
	BEQ	132206\$; Нет
	CLRB	@23206	; Текущая дорожка - нулевая
	BR	132112\$	
132206\$:	BIT	#100,23204	; В УС сброшен бит перемещения
	BEQ	132246\$; Да
	BIC	#100,23204	; Сбросить в УС бит перемещения
132224\$:	MOV	23204,@#177130	; Извещение контроллера о направлении перемещ.
	MOV	#132242\$,@#304	; Адрес новой п/п обработки прерывания
	RTI		
132242\$:	CALL	134242\$; Сбросить готовность программируемого таймера
132246\$:	MOVB	23300,23246	; (23246) = текущий параметр SKIP
132254\$:	MOV	23220,23224	; (23224) = время шага
	MOV	#132322\$,@#304	; Адрес новой п/п обработки прерывания
	CALL	132276\$; Выполнить шаг в выбранном направлении
	RTI		
132276\$:	BIS	#200,23204	; Установить в УС бит шага
	MOV	23204,@#177130	; Выполнить шаг
	BIC	#200,23204	; Сбросить в УС бит шага
	RETURN		
132322\$:	CALL	134242\$; Сбросить готовность программируемого таймера
	DEC	23224	; Уменьшить значение счетчика (время шага)
	BNE	132344\$; Он не равен нулю
	DECB	23246	; Уменьшить параметр пропуска дорожки
	BNE	132254\$; Не равен нулю
	BR	132112\$	
132344\$:	RTI		
132346\$:	BIC	#14,23204	; Очистить в УС биты прекооррекции
	MOV	23226,23246	; (23246) = номера дорожек, где вкл. прекоорр.
	CMPB	#1,23300	; Параметр SKIP равен 1 ?
	BEQ	132402\$; Да
	ASRB	23246	; Уменьшить номера дорожек на 2
	ASRB	23247	
132402\$:	CMPB	23246,@23206	; Надо включать прекооррекцию ?
	BHI	132440\$; Нет
	CMPB	23247,@23206	; Включать прекооррекцию 500 нс ?
	BHI	132432\$; Да
	BIS	#4,23204	; Установить в УС бит вкл. прекооррекции 250 нс
	BR	132440\$	
132432\$:	BIS	#10,23204	; Установить в УС бит вкл. прекооррекции 500 нс
132440\$:	MOV	23204,@#177130	; Извещение контроллера о вкл. прекооррекции
	MOV	#132456\$,@#304	; Адрес новой п/п обработки прерывания

	RTI		
132456\$:	CALL	134242\$; Сбросить готовность программируемого таймера
	CLRB	@#177710	; Остановить таймер
	CALL	134316\$; Сохранить значения PОН
	MOV	#177130, R4	; R4 = адрес PCY накопителя
	MOV	R4, R5	
	TST	(R5)+	; R5 = адрес РД накопителя
	CMPB	#10, 23335	; Запрошено чтение ?
	BEQ	132570\$; Да
	BIT	#4, @R4	; Дискета защищена от записи ?
	BEQ	132532\$; Нет
	MOVB	#6, 23334	; Ошибка - защита от записи
	JMP	134422\$	
132532\$:	CMPB	#20, 23335	; Запрошена запись ?
	BEQ	132570\$	
	CMPB	#30, 23335	; Запрошено форматирование дорожки ?
	BNE	132556\$; Нет
	JMP	134474\$	
132556\$:	MOVB	#11, 23334	; Ошибка - неверный массив параметров
	JMP	134422\$	
132570\$:	MOV	#36, 23224	
	MOV	23224, R2	
	MOV	23204, R3	; R3 = УС
	MOV	#177130, R4	; R4 = адрес PCY накопителя
	MOV	R4, R5	
	TST	(R5)+	; R5 = адрес РД накопителя
	JSR	R5, 134210\$; Запуск таймера
	.WORD	2465, 132630\$, 7	
132630\$:	MOV	@R5, R0	; Прочсть слово с дискеты
	BEQ	132640\$; Это нуль
	INC	R0	
	BNE	132656\$; Это не минус один
132640\$:	TSTB	@#177710	; Таймер окончил счет ?
	BPL	132630\$; Нет
	CALL	134242\$; Сбросить готовность таймера
	SOB	R2, 132630\$	
	BR	132772\$; Ошибка (не форматирована дискета или ее нет)
132656\$:	MOV	#4, R1	
	CMPB	#5, 23301	; Формат 1024 байта в секторе ?
	BNE	132702\$; Нет
	TST	@R4	; Присутствует линия индекс ?
	BPL	132702\$; Нет
	MOV	#36, R1	
132702\$:	MOV	@R5, R0	; Прочсть слово с дискеты
	BEQ	132730\$; Это нуль
	INC	R0	
	BEQ	132730\$; Это минус один
	TSTB	@#177710	; Таймер окончил счет
	BPL	132656\$; Нет
	CALL	134242\$; Сбросить готовность таймера
	SOB	R2, 132656\$	
	BR	133002\$; Ошибка (не найден межсекторный промежуток)
132730\$:	MOV	@R5, R0	; Прочсть слово с дискеты
	BEQ	132740\$; Это нуль
	INC	R0	
	BNE	132640\$; Это не минус один
132740\$:	SOB	R1, 132730\$	
	BIS	#400, R3	; Установить в УС бит поиска заголовка
	MOV	R3, @R4	; Начать поиск
	BIC	#400, R3	; Сбросить в УС бит поиска заголовка
	MOV	#226, R0	; Время ожидания поиска
	MOV	R3, @R4	
132762\$:	TSTB	@R4	; Маркер заголовка обнаружен ?
	BMI	133034\$; Да

	S0B	R0,132762\$	
	BR	132630\$; Заголовок сектора не обнаружен
132772\$:	MOVB	#100,23334	; Ошибка - не форматирована дискета
	BR	133030\$	
133002\$:	MOVB	#101,23334	; Ошибка - не обнаружен межсекторный промежуток
	BR	133030\$	
133012\$:	MOVB	#102,23334	; Ошибка - не найден сектор с заданным номером
	BR	133030\$	
133022\$:	MOVB	#3,23334	
133030\$:	JMP	134422\$	
133034\$:	TST	@R5	; Сбросить готовность контроллера дисковогода
133036\$:	TSTB	@R4	; Следующее слово прочитано ?
	BPL	133036\$; Нет
	CMP	#120776,@R5	; Это адресный маркер заголовка сектора?
	BEQ	133212\$; Да
	MOV	#14,R0	
133054\$:	ADD	23252,R0	
	ASL	R0	
	ASL	R0	
133064\$:	CLRB	@#177710	
	MOV	R0,@#177712	; Загрузить буфер таймера
	CMP	R0,@#177714	; Буфер загружен ?
	BNE	133064\$; Нет
	MOV	#133134\$,@#304	; Адрес новой п/п обработки прерываний
	MOVB	#107,@#177710	; Запустить таймер
	MOV	R3,23204	
	MOV	R2,23224	
	CALL	134356\$; Восстановить P0H
	RTI		
133134\$:	CALL	134242\$; Сбросить готовность таймера
	CALL	134316\$; Сохранить содержимое P0H
	DEC	23224	; Уменьшить счетчик (число оборотов диска)
	BLE	133012\$; Он равен нулю (ошибка)
	MOV	#2465,@#177712	; Загрузить буфер таймера временем оборота
	MOV	#7,@#177710	; Перевести таймер в режим "ПУСК" (без прерыв.)
	MOV	23224,R2	; R2 = значение счетчика
	MOV	23204,R3	; R3 = УС
	MOV	#177130,R4	; R4 = адрес РСУ накопителя
	MOV	R4,R5	
	TST	(R5)+	; R5 = адрес РД накопителя
	JMP	134100\$; На поиск заголовка сектора
133212\$:	TSTB	@R4	; Очередное слово сформировано ?
	BPL	133212\$; Нет
	MOV	@R5,23246	; Прочсть его
	CMP	23212,23246	; Это идентификатор стороны и дорожки ?
	BEQ	133276\$; Да
133232\$:	TSTB	@R4	; Очередное слово сформировано ?
	BPL	133232\$; Нет
	TST	@R5	; Игнорировать прочитанное слово(номер сектора)
	CALL	134300\$; Контрольная сумма в порядке ?
	BEQ	133270\$; Нет
	MOV	R3,23204	; Сохранить УС
	CALL	134356\$; Восстановить P0H
	MOVB	#377,@23206	; Поставить номер неопределенной дорожки
	JMP	131726\$	
133270\$:	MOV	#40,R0	
	BR	133054\$	
133276\$:	TSTB	@R4	; Очередное слово сформировано ?
	BPL	133276\$; Нет
	MOV	@R5,R0	; R0 = прочитанное слово
	SWAB	R0	
	CMPB	23341,R0	; Номер сектора совпадает ?
	BNE	133270\$; Нет
	MOV	23344,R2	; R2 = число слов для обмена

	MOV	23342, @#177010	; Адрес буфера ОЗУ ЦП
	CALL	134300\$; Контрольная сумма в порядке ?
	BNE	133344\$; Да
	MOVB	#2, 23334	; Ошибка в контрольной сумме заголовка
	BR	133030\$	
133344\$:	CMPB	#10, 23335	; Задана операция чтения ?
	BNE	133506\$; Нет
	TST	@R5	; Сбросить готовность контроллера
	MOV	#21, R1	
	CALL	134132\$; Поиск пропуска синхроимпульсов
	BEQ	133476\$; Не найдено
133370\$:	TSTB	@R4	; Очередное слово сформировано ?
	BPL	133370\$; Нет
	CMP	#120773, @R5	; Это маркер данных ?
	BNE	133476\$; Нет
	MOV	23252, R1	; R1 = число слов в секторе
	TST	R2	; Надо прочитать 0 слов ?
	BEQ	133442\$; Да (сделать проверку КС сектора)
133412\$:	TSTB	@R4	; Очередное слово сформировано ?
	BPL	133412\$; Нет
	MOV	@R5, R0	; Прочитать его
	SWAB	R0	
	MOV	R0, @#177014	; Переслать в ОЗУ ЦП
	INC	@#177010	; Переход к следующему адресу
	DEC	R2	; Уменьшить счетчик считываемых слов
	BEQ	133450\$; Все прочитали
	SOB	R1, 133412\$; Цикл по числу слов в секторе
	BR	133452\$	
133442\$:	TSTB	@R4	; Очередное слово сформировано ?
	BPL	133442\$	
	TST	@R5	; Пропуск конца сектора
133450\$:	SOB	R1, 133442\$; Цикл по числу слов в секторе
133452\$:	MOV	R2, 23344	; Сохранить число оставшихся слов
	CALL	134300\$; Контрольная сумма в порядке ?
	BNE	133706\$; Да
	MOVB	#1, 23334	; Ошибка в контрольной сумме зоны данных
133472\$:	JMP	133030\$	
133476\$:	MOVB	#4, 23334	; Не найден маркер данных
	BR	133472\$	
133506\$:	TST	R2	; Счетчик записываемых слов равен нулю ?
	BNE	133516\$; Нет
	JMP	134416\$; Да
133516\$:	MOV	#47116, @R5	; Запись межсекторного промежутка
	MOV	#13, R0	; R0 = размер межсекторного пром. перед данными
133526\$:	TSTB	@R4	; Предыдущее слово записано ?
	BPL	133526\$; Нет
	MOV	#47116, @R5	; Запись межсекторного промежутка
	SOB	R0, 133526\$	
	MOV	#6, R0	; R0 = размер нулевой зоны
133544\$:	TSTB	@R4	; Очередное слово записано ?
	BPL	133544\$; Нет
	MOV	#0, @R5	; Запись нулевой зоны
	SOB	R0, 133544\$	
	BIS	#1000, R3	; Установить бит пропуска синхроимпульсов
133562\$:	TSTB	@R4	; Очередное слово записано ?
	BPL	133562\$; Нет
	MOV	#120641, @R5	; Запись первого слова маркера данных
	MOV	R3, @R4	; Установить пропуск синхроимпульсов
	BIC	#1000, R3	; Сбросить бит пропуска синхроимпульсов
133600\$:	TSTB	@R4	; Очередное слово записано ?
	BPL	133600\$; Нет
	MOV	#175641, @R5	; Запись второго слова маркера данных
	MOV	23252, R1	; R1 = размер сектора в словах
	DEC	R1	

133616\$:	TSTB	@R4	; Очередное слово записано ?
	BPL	133616\$	
	MOV	@#177014, @R5	; Запись первого слова данных
	MOV	R3, @R4	; Сброс пропуска синхроимпульсов
	INC	@#177010	; Переход к следующему адресу
	DEC	R2	; Уменьшить счетчик записываемых слов
	BEQ	133664\$	
133640\$:	TSTB	@R4	; Очередное слово записано ?
	BPL	133640\$; Нет
	MOV	@#177014, @R5	; Запись данных
	INC	@#177010	; Переход к следующему адресу
	DEC	R2	; Уменьшить счетчик записываемых слов
	BEQ	133674\$	
	SOB	R1, 133640\$	
	BR	133676\$	
133664\$:	TSTB	@R4	; Очередное слово записано ?
	BPL	133664\$; Нет
	MOV	#0, @R5	; Запись нуля
133674\$:	SOB	R1, 133664\$	
133676\$:	MOV	R2, 23344	; Сохранить счетчик записываемых слов
	CALL	134256\$; Запись контрольной суммы
133706\$:	TST	23344	; Счетчик записываемых слов равен нулю
	BNE	133720\$; Нет
	JMP	134416\$; Да
133720\$:	MOV	@#177010, 23342	; Сохранение регистра адреса
	INCB	23341	; Увеличить номер текущего сектора
	CMPB	23341, 23301	; Он превышает номер последнего сектора ?
	BLOS	134064\$; Нет
	CMPB	#1, 23336	; Запрошена односторонняя операция ?
	BEQ	133766\$; Да
	BIT	#40, R3	; Текущая сторона - верх ?
	BNE	133766\$; Да
	INC	23212	; Увеличить номер текущей стороны
	BR	134010\$	
133766\$:	INCB	23340	; Увеличить номер запрашиваемой дорожки
	MOVB	23340, 23212	
	SWAB	23212	
	CLRB	23212	; Текущая сторона - низ
134010\$:	CLRB	@#177710	; Остановить программируемый таймер
	TST	@#177714	; Сбросить его готовность
	TST	@#177714	
	MOVB	#1, 23341	; Текущий номер сектора - 1
	CMPB	#1, 23336	; Запрошена односторонняя операция
	BEQ	134050\$; Да
	MOV	#40, R0	
	XOR	R0, R3	; Переменить сторону
134050\$:	MOV	R3, 23204	
	CALL	134356\$; Восстановить PОН
	JMP	131726\$	
134064\$:	TST	@#177714	; Сбросить готовность программируемого таймера
	TST	@#177714	
	MOV	#36, R2	
134100\$:	TST	@R5	; Перевод контроллера в режим чтения
	CLR	R1	
	MOVB	23302, R1	; R1 = текущий размер межсекторного промежутка
	ADD	#6, R1	
	CALL	134132\$; Поиск нулевой зоны и начала след. сектора
	BEQ	134126\$; Она не найдена
	JMP	133036\$	
134126\$:	JMP	132630\$	
; П/п поиска нулевой зоны			
134132\$:	TSTB	@R4	; Очередное слово считано ?
	BPL	134132\$; Нет
	MOV	@R5, R0	


```

        BEQ      134152$          ; Это нулевая зона
        INC      R0
        BEQ      134152$
        SOB      R1,134132$
        BR       134200$
134152$:BIS      #400,R3          ; Уст. бит поиска пропуска синхроимпульсов
        MOV      R3,@R4          ; Начать поиск
        BIC      #400,R3          ; Сбросить бит поиска пропуска синхроимпульсов
        MOV      #226,R0
        MOV      R3,@R4
134172$:TSTB     @R4              ; Поиск окончен
        BMI      134204$          ; Да
        SOB      R0,134172$
134200$:SEZ                      ; Операция окончилась неуспешно
        RETURN
134204$:TST      @R5
        RETURN
; П/п запуска таймера с вызовом п/п по его прерыванию
134210$:CLRB     @#177710         ; Остановить программируемый таймер
        MOV      @R5,@#177712     ; Занести в буфер временную задержку
        CMP      @R5,@#177714
        BNE      134210$
        TST      (R5)+
        MOV      (R5)+,@#304       ; Занести адрес п/п в вектор прерывания
        MOV      (R5)+,@#177710    ; Запустить таймер
        RTS      R5
; П/п сброса готовности программируемого таймера
134242$:TST      @#177714         ; Сбросить готовность таймера
        TSTB     @#177710         ; Она сброшена ?
        BMI      134242$          ; Нет
        RETURN
; П/п записи контрольной суммы
134256$:BIT      #40000,@R4       ; Запись КС
        BEQ      134256$
        MOV      #47116,@R5
134270$:TSTB     @R4
        BPL      134270$
        TST      @R5
        RETURN
; П/п проверки контрольной суммы
134300$:MOV      #36,R0
134304$:BIT      #40000,@R4       ; Проверка КС
        BNE      134314$
        SOB      R0,134304$
134314$:RETURN
; П/п сохранения РОН
134316$:MOV      R0,23232
        MOV      R1,23234
        MOV      R2,23236
        MOV      R3,23240
        MOV      R4,23242
        MOV      R5,23244
        MOV      @#177010,23230
        RETURN
; П/п восстановления РОН
134356$:MOV      23232,R0
        MOV      23234,R1
        MOV      23236,R2
        MOV      23240,R3
        MOV      23242,R4
        MOV      23244,R5
        MOV      23230,@#177010
        RETURN
; Завершающая процедура

```

```

134416$:CLRB      23334          ; Код ошибки = 0 (все нормально)
134422$:CALL     125140$        ; Переслать код возврата в ЦП
      JSR        R5,175400$     ; Установить адрес п/п выключения двигателя
      .WORD      372,134454$
      CALL      134356$        ; Восстановить PОН
      CLR       @#177710       ; Остановить таймер
      CALL      176422$
      RTI

; П/п остановка двигателя
134454$:BIC      #2037,23204    ; Сбросить биты выбора устройства,вкл. мотора
      MOV       23204,@#177130 ; Выключить двигатель
      RETURN

134472$:BR       134416$
; Форматирование дорожки
134474$:MOV      23204,R3        ; R3 = УС накопителя
      MOV      #177130,R4        ; R4 = адрес РУС накопителя
      MOV      R4,R5
      TST      (R5)+             ; R5 = адрес РД накопителя
      SWAB     23212
      MOV      23341,R2          ; R2 = формат сектора
      BIC      #177774,R2
      BNE      134540$
      MOV      #13,23334         ; Ошибка = неверный формат сектора
      JMP      134422$
134540$:MOVB     135422$(R2),23246; (23246) = число секторов в дорожке
      MOVB     135426$(R2),23302; (23302) = размер межсекторного промежутка
      MOV      R2,R1
      ASL      R1
      MOV      135412$(R1),R1    ; R1 = число слов в секторе
      SWAB     R2
      INC      R2                ; R2 = формат сектора : номер сектора
      MOV      #1750,R0
134574$:TST      @R4             ; Присутствует линия ИНДЕКС ?
      BPL      134616$          ; Нет
      SOB      R0,134574$
134602$:TST      @R5             ; Установить режим готовности чтения
      MOVB     #14,23334         ; Ошибка = ошибка линии ИНДЕКС
      JMP      134422$
134616$:MOV      #6200,R0        ; R0 = число слов в дорожке
      MOV      #47116,@R5        ; Заполнение дорожки
134626$:TST      @R4             ; Присутствует линия ИНДЕКС ?
      BMI      134646$
134632$:TSTB     @R4             ; Предыдущее слово записано ?
      BPL      134632$          ; Да
      MOV      #47116,@R5        ; Заполнение дорожки
      SOB      R0,134626$
      BR       134602$
134646$:MOV      #20,R0          ; R0 = размер межсект. пром. в начале дорожки
134652$:MOV      #47116,@R5      ; Заполнение межсект. пром. перед заголовком
134656$:TSTB     @R4             ; Очередное слово записано ?
      BPL      134656$          ; Нет
      MOV      #47116,@R5        ; Заполнение межсект. пром. перед заголовком
      SOB      R0,134656$
      MOV      #6,R0            ; R0 = размер нулевой зоны
134674$:TSTB     @R4             ; Очередное слово записано ?
      BPL      134674$          ; Нет
      MOV      #0,@R5            ; Заполнение нулевой зоны
      SOB      R0,134674$
      BIS      #1000,R3          ; Установить бит пропуска синхроимпульсов
134712$:TSTB     @R4
      BPL      134712$
      MOV      #120641,@R5       ; Запись первого слова адресного маркера
      MOV      R3,@R4            ; Установить режим пропуска синхроимпульсов
      BIC      #1000,R3          ; Сбросить бит пропуска синхроимпульсов

```

134730\$:	TSTB	@R4	
	BPL	134730\$	
	MOV	#177241,@R5	; Запись второго слова адресного маркера
134740\$:	TSTB	@R4	
	BPL	134740\$	
	MOV	23212,@R5	; Запись номера дорожки и стороны
	MOV	R3,@R4	; Сбросить режим пропуска синхроимпульсов
134752\$:	TSTB	@R4	
	BPL	134752\$	
	MOV	R2,@R5	; Запись номера сектора и формата сектора
	MOV	#13,R0	; R0 = размер межсект. пром. перед данными
134764\$:	BIT	#40000,@R4	; Запись контрольной суммы
	BEQ	134764\$	
	MOV	#47116,@R5	; Заполнение межсект. пром. перед данными
134776\$:	TSTB	@R4	
	BPL	134776\$	
	MOV	#47116,@R5	; Заполнение межсект. пром. перед данными
	SOB	R0,134776\$	
	MOV	#6,R0	; R0 = размер нулевой зоны
135014\$:	TSTB	@R4	
	BPL	135014\$	
	MOV	#0,@R5	; Заполнение нулевой зоны
	SOB	R0,135014\$	
	BIS	#1000,R3	; Установить бит пропуска синхроимпульсов
135032\$:	TSTB	@R4	
	BPL	135032\$	
	MOV	#120641,@R5	; Запись первого слова маркера данных
	MOV	R3,@R4	; Установить режим пропуска синхроимпульсов
	BIC	#1000,R3	; Сбросить бит пропуска синхроимпульсов
135050\$:	TSTB	@R4	
	BPL	135050\$	
	MOV	#175641,@R5	; Запись второго слова маркера данных
	MOV	R1,R0	; R0 = размер сектора в словах
135062\$:	TSTB	@R4	
	BPL	135062\$	
	MOV	23342,@R5	; Запись первого слова зоны данных
	MOV	R3,@R4	; Сбросить режим пропуска синхроимпульсов
	DEC	R0	
135076\$:	TSTB	@R4	
	BPL	135076\$	
	MOV	23342,@R5	; Заполнение зоны данных
	SOB	R0,135076\$	
	INC	R2	; Увеличить номер сектора
	MOVB	23302,R0	; R0 = размер межсект. пром. перед заголовком
135116\$:	BIT	#40000,@R4	; Запись контрольной суммы
	BEQ	135116\$	
	DECB	23246	; Уменьшить счетчик секторов
	BNE	134652\$	
	MOV	#1750,R0	
135136\$:	TST	@R4	; Присутствует линия ИНДЕКС ?
	BMI	135170\$; Да
135142\$:	TSTB	@R4	
	BPL	135142\$	
	MOV	#47116,@R5	; Заполнение конца дорожки
	SOB	R0,135136\$	
	TST	@R5	; Установить режим готовности чтения
	MOVB	#14,23334	; Ошибка - ошибка линии ИНДЕКС
	JMP	134422\$	
135170\$:	MOV	#5,R0	
135174\$:	TSTB	@R4	
	BPL	135174\$	
	MOV	#47116,@R5	; Заполнение в начале первых пяти слов
	SOB	R0,135174\$	
	TST	@R5	; Установить режим готовности чтения

```

        JMP      134416$
; П/п-мы установок параметров
135214$: CLR      R0
        MOV      23335, R0      ; R0 = номер устанавливаемого параметра (1 - 4)
        CMP      #4, R0        ; Он в диапазоне (1 - 4) ?
        BHIS     135242$        ; Да
        MOV      #11, 23334     ; Ошибка - неверный массив параметров
        JMP      134422$
135242$: MOV      23340, 23246    ; (23246) = значение устанавливаемого параметра
        DEC      R0
        MOV      #135272$, R1
        ADD      R0, R1
        ADD      R0, R1
        MOV      @R1, R1
        CALL     @R1            ; Вызов соответствующей п/п
        JMP      134416$
; Адреса начала п/п установки значений параметров
135272$: .WORD    135350$, 135356$, 135364$, 135372$
135302$: MOV      23337, R0      ; R0 = номер накопителя
        CMP      #377, R0       ; Устанавливается для всех накопителей ?
        BEQ      135326$        ; Да
        MOV      #1, R2         ; R2 = число накопителей, для кот. уст. парам.
        BICB     #374, R0
        BR       135334$
135326$: CLR      R0
        MOV      #4, R2         ; R2 = число накопителей, для кот. уст. парам.
135334$: ADD      R0, R1         ; Корректировка под номер накопителя
135336$: MOV      23246, @R1     ; Установка нового значения
        INC      R1
        SOB      R2, 135336$
        RETURN
135350$: MOV      #23260, R1     ; R1 = адрес буфера времен шага
        BR       135302$
135356$: MOV      #23264, R1     ; R1 = адрес буфера времен загрузки головки
        BR       135302$
135364$: MOV      #23270, R1     ; R1 = адрес буфера числа секторов в дорожке
        BR       135302$
135372$: MOV      #23274, R1     ; R1 = адрес буфера параметра SKIP
        MOV      #2, 23246       ; Принудительная установка пропуска дорожки
        BR       135302$
; Устанавливаемые биты в РУС накопителя при выборе соотв. устройства
135406$: .BYTE    3, 2, 1, 0
; Число слов в секторе (формат x, 256, 512, 1024 байт)
135412$: .WORD    0, 200, 400, 1000
; Число секторов в дорожке (формат x, 256, 512, 1024 байт)
135422$: .BYTE    0, 20, 12, 5
; Размер межсекторного промежутка перед заголовком (формат x, 256, 512, 1024 байт)
135426$: .BYTE    0, 33, 22, 72
135432$: .BYTE    0, 240, 124, 242

```

```

; *****
; *
; *      ГРАФИКА ПОЛЬЗОВАТЕЛЯ, ПРИНТЕР, ПЕЧАТЬ ЭКРАНА      *
; *
; *      Адреса 135436 - 157777                               *
; *
; *****

```

```

; ТОЧКА ВХОДА ПРИ ИСПОЛНЕНИИ ESC-ПОСЛЕДОВАТЕЛЬНОСТИ ESC %!0 (ВХОД В ГРАФИКУ)
135436$: MOV      R5, - (SP)
        MOV      SP, R5
135442$: CALL     136274$

```

```

TST      R0
BNE      135442$
JMP      136056$
; П/п установки начальных значений для п/п графического режима
135456$:MOV R5, -(SP)
MOV      SP, R5
MOV      22664, R0
ASL      R0
ASL      R0
ADD      #2500, R0
ADD      #-4, R0
MOV      R0, 23432
MOV      23432, 23434
MOV      #1200, 23412
CLR      23420
CLR      23422
MOV      #1, R0
MOV      23420, R1
BIC      #177770, R1
MOV      R1, 23440
ASH      23440, R0
MOV      R0, 23436
MOV      23432, R0
MOV      23422, R1
ASL      R1
ASL      R1
SUB      R1, R0
MOV      R0, 23434
MOV      @23434, R0
MOV      23420, R1
ASH      #-3, R1
MOV      R1, 23426
ADD      23426, R0
MOV      R0, 23430
CLR      23442
CLR      23444
CLR      23446
CLR      23450
CLR      23452
CLR      23454
CLR      23456
CLR      23460
CLR      23462
JMP      136056$
; Точка входа при вызове п/п печати экрана
135704$:MOV R0, -(SP)
MOV      R1, -(SP)
CALL     140326$
MOV      (SP)+, R1
MOV      (SP)+, R0
RETURN
; П/п вывода символа на принтер
135722$:MOV R5, -(SP)
MOV      SP, R5
BIT      #2, 177101 ; Принтер в состоянии ON-LINE ?
BEQ      136000$ ; Нет
135736$:BIT #200, 177101 ; Бит подтверждения сброшен ?
BNE      135736$ ; Нет
MOV      4(R5), 177100 ; Послать код символа на линию данных
BIC      #200, 177102 ; Установить сигнал строба
135762$:BIT #200, 177101 ; Получен сигнал подтверждения ?
BEQ      135762$ ; Нет
MOV      #200, 177102 ; Сбросить сигнал строба
136000$:JMP 136056$

```

; П/п инициализации параллельного порта

```
136004$:MOV    R5, -(SP)
          MOV    SP, R5
          MOV    #174, 177103
          MOV    #200, 177102
          JMP    136056$
```

```
136030$:MOV    R5, R0
          MOV    SP, R5
          MOV    R4, -(SP)
          MOV    R3, -(SP)
          MOV    R2, -(SP)
          CLR    -(SP)
          JMP    @R0
```

```
136046$:MOV    R5, R2
          MOV    -(R2), R4
          MOV    -(R2), R3
          MOV    -(R2), R2
```

```
136056$:MOV    R5, SP
          MOV    (SP)+, R5
          RETURN
```

; Блок описания Esc-последовательностей графического режима

```
136064$: .WORD 136414$, 111, 0, 136502$ ; LI
136074$: .WORD 136622$, 126, 136602$, 120 ; MV, MP
          .WORD 136562$, 115, 136542$, 114 ; MM, ML
          .WORD 0, 136642$
136120$: .WORD 137054$, 125, 136706$, 122 ; RU, RR
          .WORD 136700$, 104, 0, 137064$ ; RD
136140$: .WORD 137122$, 104, 0, 137164$ ; TD
136150$: .WORD 140140$, 111, 137722$, 110 ; UI, UH
          .WORD 137620$, 107, 137442$, 106 ; UG, UF
          .WORD 137432$, 105, 137424$, 104 ; UE, UD
          .WORD 137412$, 103, 137334$, 102 ; UC, UB
          .WORD 137222$, 101, 0, 140174$ ; UA
136220$: .WORD 140232$, 41, 0, 140302$ ; %!
136230$: .WORD 140202$, 45, 137172$, 125 ; %*, U*
          .WORD 137072$, 124, 136650$, 122 ; T*, R*
          .WORD 136512$, 115, 136364$, 114 ; M*, L*
          .WORD 0, 140310$
136264$: .WORD 136334$, 33, 0, 140316$ ; Esc
```

```
136274$:JSR    R5, 136030$
          ADD    #-6, SP
          CALL   145564$
          MOVB   R0, -7(R5)
          MOVB   -7(R5), R0
          BIC    #177400, R0
          MOV    #136264$, R1
          JMP    144650$
```

; П/п обработки символа Esc

```
136334$:CALL   145564$
          MOVB   R0, -7(R5)
          MOVB   -7(R5), R0
          BIC    #177400, R0
          MOV    #136230$, R1
          JMP    144650$
```

; П/п обработки Esc L

```
136364$:CALL   145564$
          MOVB   R0, -7(R5)
          MOVB   -7(R5), R0
          BIC    #177400, R0
          MOV    #136064$, R1
          JMP    144650$
```

; П/п обработки Esc LI

```

136414$:CALL    143756$
             MOV    23424,23456
             CALL   143756$
             MOV    23424,23460
             CALL   143756$
             MOV    23424,23462
             MOV    23462,@SP
             MOV    23460,-(SP)
             MOV    23456,-(SP)
             CALL   117026$
             CMP    (SP)+,(SP)+
             MOVB   #1,R0
             BR     136506$
136502$:MOVB   #1,R0
136506$:JMP     136046$
; П/п обработки Esc M
136512$:CALL   145564$
             MOVB   R0,-7(R5)
             MOVB   -7(R5),R0
             BIC    #177400,R0
             MOV    #136074$,R1
             JMP     144650$
; П/п обработки Esc ML
136542$:CALL   143756$
             MOV    23424,23442
             MOVB   #1,R0
             BR     136506$
; П/п обработки Esc MM
136562$:CALL   143756$
             MOV    23424,23446
             MOVB   #1,R0
             BR     136506$
; П/п обработки Esc MP
136602$:CALL   143756$
             MOV    23424,23450
             MOVB   #1,R0
             BR     136506$
; П/п обработки Esc MV
136622$:CALL   143756$
             MOV    23424,23444
             MOVB   #1,R0
             BR     136506$
136642$:MOVB   #1,R0
             BR     136506$
; П/п обработки Esc R
136650$:CALL   145564$
             MOVB   R0,-7(R5)
             MOVB   -7(R5),R0
             BIC    #177400,R0
             MOV    #136120$,R1
             JMP     144650$
; П/п обработки Esc RD
136700$:MOVB   #1,R0
             BR     136506$
; П/п обработки Esc RR
136706$:CALL   144104$
             CMP    23420,23412
             BGE    136732$
             CMP    23422,22664
             BLT    136740$
136732$:MOVB   #1,R0
             BR     136506$
136740$:MOV    23420,-12(R5)
             MOV    23422,-14(R5)

```

```

CALL    144104$
CMP     23420,23412
BGE     137000$
CMP     23422,22664
BLT     137006$
137000$:MOVB  #1,R0
BR      136506$
137006$:CALL 143756$
MOV     23424,@SP
MOV     23422,-(SP)
MOV     23420,-(SP)
MOV     -14(R5),-(SP)
MOV     -12(R5),-(SP)
CALL    144666$
ADD     #10,SP
MOVB    #1,R0
BR      137060$
; П/п обработки Esc RU
137054$:MOVB  #1,R0
137060$:JMP   136046$
137064$:MOVB  #1,R0
BR      137060$
; П/п обработки Esc T
137072$:CALL 145564$
MOVB    R0,-7(R5)
MOVB    -7(R5),R0
BIC     #177400,R0
MOV     #136140$,R1
JMP     144650$
; П/п обработки Esc TD
137122$:CALL 143756$
MOV     23424,23452
CALL    143756$
MOV     23424,23454
MOV     23452,@SP
CALL    117004$
MOVB    #1,R0
BR      137060$
137164$:MOVB  #1,R0
BR      137060$
; П/п обработки Esc U
137172$:CALL 145564$
MOVB    R0,-7(R5)
MOVB    -7(R5),R0
BIC     #177400,R0
MOV     #136150$,R1
JMP     144650$
; П/п обработки Esc UA
137222$:MOV   23430,177010
TST     177024
BNE     137236$
137236$:BIT   #4,23440
BEQ     137254$
MOV     177022,R0
BR      137260$
137254$:MOV   177020,R0
137260$:MOV   R0,R1
CLR     R0
MOV     23440,R4
BIC     #177774,R4
ASL     R4
ASL     R4
NEG     R4
ASHC    R4,R0

```



```

MOV      R1,@SP
BIC      #177760,@SP
CALL     144304$
MOV      #15,@SP
CALL     145612$
MOVB     #1,R0
BR       137060$
; П/п обработки Esc UB
137334$:MOV 23414,23420
MOV      23416,23422
MOV      23422,23414
MOV      23420,23416
CALL     151714$
MOV      23420,23414
MOV      23422,23416
MOVB     #1,R0
BR       137060$
; П/п обработки Esc UC
137412$:CALL 151650$
MOVB     #1,R0
BR       137060$
; П/п обработки Esc UD
137424$:MOVB #1,R0
BR       137436$
; П/п обработки Esc UE
137432$:MOVB #1,R0
137436$:JMP 136046$
; П/п обработки Esc UF
137442$:CALL 144104$
CMP      23420,23412
BGE      137466$
CMP      23422,22664
BLT      137474$
137466$:MOVB #1,R0
BR       137436$
137474$:MOV #1,R0
MOV      23420,R1
BIC      #177770,R1
MOV      R1,23440
ASH      23440,R0
MOV      R0,23436
MOV      23432,R0
MOV      23422,R1
ASL      R1
ASL      R1
SUB      R1,R0
MOV      R0,23434
MOV      @23434,R0
MOV      23420,R1
ASH      #-3,R1
MOV      R1,23426
ADD      23426,R0
MOV      R0,23430
MOV      23420,23414
MOV      23422,23416
MOVB     #1,R0
BR       137436$
; П/п обработки Esc UG
137620$:CALL 144104$
CMP      23420,23412
BGE      137644$
CMP      23422,22664
BLT      137652$
137644$:MOVB #1,R0

```

```

BR      137436$
137652$:MOV 23442,177016
MOV     23430,177010
TST     177024
BNE     137674$
137674$:MOV 23430,177010
MOV     23436,177024
CALL    141330$
MOVB    #1,R0
BR      137436$
; П/п обработки Esc UH
137722$:CALL 144104$
CMP     23420,23412
BGE     137746$
CMP     23422,22664
BLT     137754$
137746$:MOVB #1,R0
BR      140134$
137754$:MOV #1,R0
MOV     23420,R1
BIC     #177770,R1
MOV     R1,23440
ASH     23440,R0
MOV     R0,23436
MOV     23432,R0
MOV     23422,R1
ASL     R1
ASL     R1
SUB     R1,R0
MOV     R0,23434
MOV     @23434,R0
MOV     23420,R1
ASH     #-3,R1
MOV     R1,23426
ADD     23426,R0
MOV     R0,23430
MOV     23420,23414
MOV     23422,23416
MOV     23442,177016
MOV     23430,177010
TST     177024
BNE     140114$
140114$:MOV 23430,177010
MOV     23436,177024
MOVB    #1,R0
140134$:JMP 136046$
; П/п обработки Esc UI
140140$:MOV 23416,@SP
MOV     23414,-(SP)
CALL    144454$
TST     (SP)+
MOV     #15,@SP
CALL    145612$
MOVB    #1,R0
BR      140134$
140174$:MOVB #1,R0
BR      140134$
; П/п обработки Esc %
140202$:CALL 145564$
MOVB    R0,-7(R5)
MOVB    -7(R5),R0
BIC     #177400,R0
MOV     #136220$,R1
JMP     144650$

```

; П/п обработки Esc %!

```
140232$:CALL 145564$
        MOVB R0, -7(R5)
        CMPB -7(R5), #'0
        BEQ 140274$
        MOVB -7(R5), R0
        BIC #177400, R0
        MOV R0, @SP
        CALL 113574$
        CLR R0
        BR 140134$
140274$:MOVB #1, R0
        BR 140134$
140302$:MOVB #1, R0
        BR 140134$
140310$:MOVB #1, R0
        BR 140134$
140316$:MOVB #1, R0
        BR 140134$
140324$:BR 140134$
```

; П/п печати экрана

```
140326$:JSR R5, 136030$
        ADD #-22, SP
        BIT #2, 177101
        BEQ 140602$
        BIC #100, 177700
        MOV #15, @SP
        CALL 135722$
        MOV #12, @SP
        CALL 135722$
        CLR -10(R5)
140400$:CMP -10(R5), #79.
        BGT 140552$
        MOV #1, -20(R5)
140416$:CMP -20(R5), #2
        BGT 140606$
        MOV #33, @SP
        CALL 135722$
        MOV #52, @SP
        CALL 135722$
        MOV #1, @SP
        CALL 135722$
        MOV #30, @SP
        CALL 135722$
        MOV #3, @SP
        CALL 135722$
        CLR -12(R5)
140502$:CMP -12(R5), #263.
        BGT 140622$
        MOV 23432, R0
        MOV -12(R5), R1
        ASL R1
        ASL R1
        SUB R1, R0
        MOV @R0, R0
        ADD -10(R5), R0
        MOV R0, 177010
        TST 177024
        BEQ 141004$
        BR 141004$
140552$:MOV #33, @SP
        CALL 135722$
        MOV #100, @SP
        CALL 135722$
```

```

MOV      #14,@SP
CALL     135722$
140602$: JMP     136046$
140606$: INC     -10(R5)
BR       140400$
140614$: INC     -20(R5)
BR       140416$
140622$: MOV     #33,@SP
CALL     135722$
MOV      #63,@SP
CALL     135722$
MOV      -20(R5),R1
MUL      #22.,R1
MOV      R1,@SP
ADD      #-21.,@SP
CALL     135722$
MOV      #12,@SP
CALL     135722$
MOV      #15,@SP
CALL     135722$
BIT      #200,177700
BEQ      140614$
CMP      177702,#4
BNE      140614$
MOV      #33,@SP
CALL     135722$
MOV      #100,@SP
CALL     135722$
MOV      #15,@SP
CALL     135722$
MOV      #12,@SP
CALL     135722$
MTPS     #340
BIS      #100,177700
MTPS     #0
BR       140602$
141004$: MOV     177020,-30(R5)
MOV      #1,-14(R5)
141020$: CMP     -14(R5),#8.
BGT      141246$
MOV      -30(R5),R0
BIC      #177760,R0
MOV      R0,-16(R5)
MOV      -22(R5),R0
ASL      R0
MOV      #153540,R1
MOV      #1,R4
MOV      -16(R5),R2
ASL      R2
SUB      R2,R4
SUB      -20(R5),R4
ASH      R4,R1
BIC      #177776,R1
BIS      R1,R0
MOV      R0,-22(R5)
MOV      -24(R5),R0
ASL      R0
MOV      #177624,R1
MOV      #1,R4
MOV      -16(R5),R2
ASL      R2
SUB      R2,R4
SUB      -20(R5),R4
ASH      R4,R1

```

```

BIC      #177776, R1
BIS      R1, R0
MOV      R0, -24(R5)
MOV      -26(R5), R0
ASL      R0
MOV      #172100, R1
MOV      #1, R4
MOV      -16(R5), R2
ASL      R2
SUB      R2, R4
SUB      -20(R5), R4
ASH      R4, R1
BIC      #177776, R1
BIS      R1, R0
MOV      R0, -26(R5)
CMP      -14(R5), #4
BNE      141306$
MOV      177022, -30(R5)
BR        141322$
141246$: MOV      -22(R5), @SP
CALL     135722$
MOV      -24(R5), @SP
CALL     135722$
MOV      -26(R5), @SP
CALL     135722$
INC      -12(R5)
JMP      140502$
141306$: MOV      -30(R5), R0
ASH      #-4, R0
MOV      R0, -30(R5)
141322$: INC      -14(R5)
BR        141020$
; П/п рисования линии
141330$: MOV      R5, -(SP)
MOV      SP, R5
ADD      #-20, SP
MOV      23420, R0
SUB      23414, R0
MOV      R0, -12(R5)
MOV      23422, R0
SUB      23416, R0
MOV      R0, -14(R5)
TST      -14(R5)
BNE      141404$
TST      -12(R5)
BEQ      141560$
141404$: MOV      23420, 23414
MOV      23422, 23416
CLR      -16(R5)
TST      -14(R5)
BLT      141564$
TST      -12(R5)
BLT      141712$
CMP      -12(R5), -14(R5)
BGT      141454$
JMP      142106$
141454$: MOV      -12(R5), -20(R5)
MOV      -20(R5), -10(R5)
141470$: MOV      -10(R5), R0
DEC      -10(R5)
TST      R0
BEQ      141560$
SUB      -14(R5), -16(R5)
MOV      23436, R0

```

	ASL	R0
	BIC	#177400, R0
	MOV	R0, 23436
	BEQ	141536\$
	JMP	142260\$
141536\$:	MOV	#1, 23436
	INC	23430
	INC	23426
	JMP	142260\$
141560\$:	JMP	136056\$
141564\$:	TST	-12(R5)
	BLT	141576\$
	JMP	142722\$
141576\$:	CMP	-12(R5), -14(R5)
	BLT	141612\$
	JMP	143136\$
141612\$:	MOV	-12(R5), R0
	NEG	R0
	MOV	R0, -20(R5)
	MOV	-20(R5), -10(R5)
141632\$:	MOV	-10(R5), R0
	DEC	-10(R5)
	TST	R0
	BEQ	141560\$
	ADD	-14(R5), -16(R5)
	CLC	
	ROR	23436
	BEQ	141670\$
	JMP	143310\$
141670\$:	MOV	#200, 23436
	DEC	23430
	DEC	23426
	JMP	143310\$
141712\$:	MOV	-12(R5), R0
	NEG	R0
	CMP	R0, -14(R5)
	BLT	141732\$
141726\$:	JMP	142442\$
141732\$:	MOV	-14(R5), -20(R5)
	MOV	-20(R5), -10(R5)
141746\$:	MOV	-10(R5), R0
	DEC	-10(R5)
	TST	R0
	BEQ	141560\$
	ADD	-12(R5), -16(R5)
	SUB	#4, 23434
	MOV	@23434, R0
	ADD	23426, R0
	MOV	R0, 23430
	TST	-16(R5)
	BGE	142030\$
	MOV	-16(R5), R0
	NEG	R0
	BR	142034\$
142030\$:	MOV	-16(R5), R0
142034\$:	ASL	R0
	CMP	R0, -20(R5)
	BGE	142050\$
	JMP	142540\$
142050\$:	CLC	
	ROR	23436
	BEQ	142064\$
	JMP	142556\$
142064\$:	MOV	#200, 23436

	DEC	23430
	DEC	23426
	JMP	142556\$
142106\$:	MOV	-14(R5), -20(R5)
	MOV	-20(R5), -10(R5)
142122\$:	MOV	-10(R5), R0
	DEC	-10(R5)
	TST	R0
	BEQ	141560\$
	ADD	-12(R5), -16(R5)
	SUB	#4, 23434
	MOV	@23434, R0
	ADD	23426, R0
	MOV	R0, 23430
	TST	-16(R5)
	BGE	142204\$
	MOV	-16(R5), R0
	NEG	R0
	BR	142210\$
142204\$:	MOV	-16(R5), R0
142210\$:	ASL	R0
	CMP	R0, -20(R5)
	BLT	142376\$
	MOV	23436, R0
	ASL	R0
	BIC	#177400, R0
	MOV	R0, 23436
	BNE	142414\$
	MOV	#1, 23436
	INC	23430
	INC	23426
	BR	142414\$
142260\$:	TST	-16(R5)
	BGE	142276\$
	MOV	-16(R5), R0
	NEG	R0
	BR	142302\$
142276\$:	MOV	-16(R5), R0
142302\$:	ASL	R0
	CMP	R0, -20(R5)
	BLT	142342\$
	SUB	#4, 23434
	MOV	@23434, R0
	ADD	23426, R0
	MOV	R0, 23430
	ADD	-12(R5), -16(R5)
142342\$:	MOV	23430, 177010
	TST	177024
	BNE	142356\$
142356\$:	MOV	23430, 177010
	MOV	23436, 177024
	JMP	141470\$
142376\$:	MOV	23430, 177010
	TST	177024
	BEQ	142424\$
	BR	142424\$
142414\$:	SUB	-14(R5), -16(R5)
	BR	142376\$
142424\$:	MOV	23430, 177010
	MOV	23436, 177024
	BR	142122\$
142442\$:	MOV	-12(R5), R0
	NEG	R0
	MOV	R0, -20(R5)

	MOV	-20(R5), -10(R5)
142462\$:	MOV	-10(R5), R0
	DEC	-10(R5)
	TST	R0
	BNE	142502\$
	JMP	141560\$
142502\$:	ADD	-14(R5), -16(R5)
	CLC	
	ROR	23436
	BNE	142606\$
	MOV	#200, 23436
	DEC	23430
	DEC	23426
	BR	142606\$
142540\$:	MOV	23430, 177010
	TST	177024
	BEQ	142566\$
	BR	142566\$
142556\$:	ADD	-14(R5), -16(R5)
	BR	142540\$
142566\$:	MOV	23430, 177010
	MOV	23436, 177024
	JMP	141746\$
142606\$:	TST	-16(R5)
	BGE	142624\$
	MOV	-16(R5), R0
	NEG	R0
	BR	142630\$
142624\$:	MOV	-16(R5), R0
142630\$:	ASL	R0
	CMP	R0, -20(R5)
	BLT	142670\$
	SUB	#4, 23434
	MOV	@23434, R0
	ADD	23426, R0
	MOV	R0, 23430
	ADD	-12(R5), -16(R5)
142670\$:	MOV	23430, 177010
	TST	177024
	BNE	142704\$
142704\$:	MOV	23430, 177010
	MOV	23436, 177024
	BR	142462\$
142722\$:	MOV	-14(R5), R0
	NEG	R0
	CMP	-12(R5), R0
	BLE	142742\$
	JMP	143472\$
142742\$:	MOV	-14(R5), R0
	NEG	R0
	MOV	R0, -20(R5)
	MOV	-20(R5), -10(R5)
142762\$:	MOV	-10(R5), R0
	DEC	-10(R5)
	TST	R0
	BNE	143002\$
	JMP	141560\$
143002\$:	SUB	-12(R5), -16(R5)
	ADD	#4, 23434
	MOV	@23434, R0
	ADD	23426, R0
	MOV	R0, 23430
	TST	-16(R5)
	BGE	143050\$

	MOV	-16(R5), R0
	NEG	R0
	BR	143054\$
143050\$:	MOV	-16(R5), R0
143054\$:	ASL	R0
	CMP	R0, -20(R5)
	BGE	143070\$
	JMP	143574\$
143070\$:	MOV	23436, R0
	ASL	R0
	BIC	#177400, R0
	MOV	R0, 23436
	BEQ	143114\$
	JMP	143612\$
143114\$:	MOV	#1, 23436
	INC	23430
	INC	23426
	JMP	143612\$
143136\$:	MOV	-14(R5), R0
	NEG	R0
	MOV	R0, -20(R5)
	MOV	-20(R5), -10(R5)
143156\$:	MOV	-10(R5), R0
	DEC	-10(R5)
	TST	R0
	BNE	143176\$
	JMP	141560\$
143176\$:	SUB	-12(R5), -16(R5)
	ADD	#4, 23434
	MOV	@23434, R0
	ADD	23426, R0
	MOV	R0, 23430
	TST	-16(R5)
	BGE	143244\$
	MOV	-16(R5), R0
	NEG	R0
	BR	143250\$
143244\$:	MOV	-16(R5), R0
143250\$:	ASL	R0
	CMP	R0, -20(R5)
	BLT	143426\$
	CLC	
	ROR	23436
	BNE	143444\$
	MOV	#200, 23436
	DEC	23430
	DEC	23426
	BR	143444\$
143310\$:	TST	-16(R5)
	BGE	143326\$
	MOV	-16(R5), R0
	NEG	R0
	BR	143332\$
143326\$:	MOV	-16(R5), R0
143332\$:	ASL	R0
	CMP	R0, -20(R5)
	BLT	143372\$
	ADD	#4, 23434
	MOV	@23434, R0
	ADD	23426, R0
	MOV	R0, 23430
	SUB	-12(R5), -16(R5)
143372\$:	MOV	23430, 177010
	TST	177024

```

        BNE      143406$
143406$:MOV      23430,177010
        MOV      23436,177024
        JMP      141632$
143426$:MOV      23430,177010
        TST      177024
        BEQ      143454$
        BR       143454$
143444$:ADD      -14(R5),-16(R5)
        BR       143426$
143454$:MOV      23430,177010
        MOV      23436,177024
        BR       143156$
143472$:MOV      -12(R5),-20(R5)
        MOV      -20(R5),-10(R5)
143506$:MOV      -10(R5),R0
        DEC      -10(R5)
        TST      R0
        BNE      143526$
        JMP      141560$
143526$:SUB      -14(R5),-16(R5)
        MOV      23436,R0
        ASL      R0
        BIC      #177400,R0
        MOV      R0,23436
        BNE      143642$
        MOV      #1,23436
        INC      23430
        INC      23426
        BR       143642$
143574$:MOV      23430,177010
        TST      177024
        BEQ      143622$
        BR       143622$
143612$:SUB      -14(R5),-16(R5)
        BR       143574$
143622$:MOV      23430,177010
        MOV      23436,177024
        JMP      142762$
143642$:TST      -16(R5)
        BGE      143660$
        MOV      -16(R5),R0
        NEG      R0
        BR       143664$
143660$:MOV      -16(R5),R0
143664$:ASL      R0
        CMP      R0,-20(R5)
        BLT      143724$
        ADD      #4,23434
        MOV      @23434,R0
        ADD      23426,R0
        MOV      R0,23430
        SUB      -12(R5),-16(R5)
143724$:MOV      23430,177010
        TST      177024
        BNE      143740$
143740$:MOV      23430,177010
        MOV      23436,177024
        BR       143506$
; П/п приема трех символов и перевода их в целое число
143756$:MOV      R5,-(SP)
        MOV      SP,R5
        ADD      #-10,SP
        CALL     145564$

```

```

    ASH      #6, R0
    MOV      R0, 23424
    CALL     145564$
    BIC      #177700, R0
    BIS      23424, R0
    ASH      #4, R0
    MOV      R0, 23424
    CALL     145564$
    MOV      R0, -10(R5)
    MOV      -10(R5), R0
    BIC      #177760, R0
    BIS      23424, R0
    MOV      R0, 23424
    BIT      #20, -10(R5)
    BNE      144100$
    MOV      23424, R0
    NEG      R0
    MOV      R0, 23424
144100$: JMP 136056$
; П/п приема пяти символов и перевода их в два целых числа
144104$: MOV R5, -(SP)
    MOV      SP, R5
    ADD      #-10, SP
    CALL     145564$
    BIC      #177740, R0
    ASH      #5, R0
    MOV      R0, 23422
    CALL     145564$
    MOV      R0, -10(R5)
    CALL     145564$
    BIC      #177740, R0
    BIS      23422, R0
    ASL      R0
    ASL      R0
    MOV      R0, 23422
    CALL     145564$
    BIC      #177740, R0
    ASH      #5, R0
    MOV      R0, 23420
    CALL     145564$
    BIC      #177740, R0
    BIS      23420, R0
    ASL      R0
    ASL      R0
    MOV      R0, 23420
    MOV      -10(R5), R0
    ASR      R0
    ASR      R0
    BIC      #177774, R0
    BIS      23422, R0
    MOV      R0, 23422
    MOV      -10(R5), R0
    BIC      #177774, R0
    BIS      23420, R0
    MOV      R0, 23420
    JMP      136056$
; П/п преобразования целого числа в три символа и их вывод
144304$: MOV R5, -(SP)
    MOV      SP, R5
    ADD      #-12, SP
    TST      4(R5)
    BGE      144342$
    MOV      4(R5), R0
    NEG      R0

```

```

MOV      R0, 4(R5)
CLR      -10(R5)
BR       144350$
144342$:MOV  #20, -10(R5)
144350$:MOV  4(R5), R0
ASH      #-12, R0
MOV      R0, @SP
BIC      #177700, @SP
BIS      #100, @SP
CALL     145612$
MOV      4(R5), R0
ASH      #-4, R0
MOV      R0, @SP
BIC      #177700, @SP
BIS      #100, @SP
CALL     145612$
MOV      4(R5), @SP
BIC      #177760, @SP
BIS      -10(R5), @SP
BIS      #40, @SP
CALL     145612$
JMP      136056$

```

; П/п преобразования двух целых чисел в пять символов и их вывод

```

144454$:MOV  R5, -(SP)
MOV      SP, R5
TST      -(SP)
MOV      6(R5), R0
ASH      #-7, R0
MOV      R0, @SP
BIC      #177740, @SP
BIS      #40, @SP
CALL     145612$
MOV      4(R5), @SP
BIC      #177774, @SP
MOV      6(R5), R0
BIC      #177774, R0
ASL      R0
ASL      R0
BIS      R0, @SP
BIS      #140, @SP
CALL     145612$
MOV      6(R5), @SP
ASR      @SP
ASR      @SP
BIC      #177740, @SP
BIS      #140, @SP
CALL     145612$
MOV      4(R5), R0
ASH      #-7, R0
MOV      R0, @SP
BIC      #177740, @SP
BIS      #40, @SP
CALL     145612$
MOV      4(R5), @SP
ASR      @SP
ASR      @SP
BIC      #177740, @SP
BIS      #100, @SP
CALL     145612$
JMP      136056$

```

```

144650$:TST  (R1)+
BEQ      144664$
CMP      R0, (R1)+

```

```

BNE      144650$
JMP      @-4(R1)
144664$: JMP      @(R1)+
; П/п закраски прямоугольной области
144666$: MOV      R5, -(SP)
MOV      SP, R5
ADD      #-22, SP
MOV      #1, R0
MOV      10(R5), R1
BIC      #177770, R1
MOV      R1, 23440
ASH      23440, R0
MOV      R0, 23436
MOV      23432, R0
MOV      12(R5), R1
ASL      R1
ASL      R1
SUB      R1, R0
MOV      R0, 23434
MOV      @23434, R0
MOV      10(R5), R1
ASH      #-3, R1
MOV      R1, 23426
ADD      23426, R0
MOV      R0, 23430
MOV      23426, -14(R5)
MOV      23434, -22(R5)
MOV      #100000, R0
MOV      23440, R1
ADD      #-16, R1
ASH      R1, R0
COM      R0
MOV      R0, -12(R5)
MOV      #1, R0
MOV      4(R5), R1
BIC      #177770, R1
MOV      R1, 23440
ASH      23440, R0
MOV      R0, 23436
MOV      23432, R0
MOV      6(R5), R1
ASL      R1
ASL      R1
SUB      R1, R0
MOV      R0, 23434
MOV      @23434, R0
MOV      4(R5), R1
ASH      #-3, R1
MOV      R1, 23426
ADD      23426, R0
MOV      R0, 23430
MOV      #100000, R0
MOV      23440, R1
ADD      #-17, R1
ASH      R1, R0
MOV      R0, -10(R5)
MOV      -14(R5), R0
SUB      23426, R0
MOV      R0, -20(R5)
MOV      23430, R0
ADD      -20(R5), R0
MOV      R0, -16(R5)
TST      -20(R5)
BNE      145234$

```

```

MOV      -12(R5), R0
COM      R0
BIC      R0, -10(R5)
145234$: MOV 14(R5), 177016
145242$: CMP 23434, -22(R5)
BLO      145270$
MOV      23430, 177010
TST      177024
BEQ      145376$
BR       145376$
145270$: MOV #1, R0
MOV      23414, R1
BIC      #177770, R1
MOV      R1, 23440
ASH      23440, R0
MOV      R0, 23436
MOV      23432, R0
MOV      23416, R1
ASL      R1
ASL      R1
SUB      R1, R0
MOV      R0, 23434
MOV      @23434, R0
MOV      23414, R1
ASH      #-3, R1
MOV      R1, 23426
ADD      23426, R0
MOV      R0, 23430
JMP      136056$
145376$: MOV 23430, 177010
MOV      -10(R5), 177024
TST      -20(R5)
BEQ      145452$
145420$: INC 23430
CMP      23430, -16(R5)
BHIS     145512$
MOV      23430, 177010
TST      177024
BEQ      145530$
BR       145530$
145452$: SUB #4, 23434
MOV      @23434, R0
ADD      23426, R0
MOV      R0, 23430
MOV      23430, R0
ADD      -20(R5), R0
MOV      R0, -16(R5)
BR       145242$
145512$: MOV 23430, 177010
TST      177024
BEQ      145546$
BR       145546$
145530$: MOV 23430, 177010
MOV      #377, 177024
BR       145420$
145546$: MOV 23430, 177010
MOV      -12(R5), 177024
BR       145452$
; П/п приема символа
145564$: MOV R5, -(SP)
MOV      R4, -(SP)
MOV      R3, -(SP)
MOV      R2, -(SP)
CALL     110762$

```

```

MOV      (SP)+, R2
MOV      (SP)+, R3
MOV      (SP)+, R4
MOV      (SP)+, R5
RETURN
; П/п вывода символа
145612$:MOV      2(SP), R0
CALL     104160$
RETURN
; Все остальные п/п относятся к обработке последовательностей Esc UB и Esc UC
145624$:MOV      SP, 23602
CALL     152144$
BHIS     145642$
JMP      145716$
145642$:CALL     151000$
BLO      145732$
145650$:MOV      #23540, R1
CMP      2(R1), #23546
BEQ      145732$
CALL     145734$
BLO      145676$
CALL     146010$
145676$:MOV      #23546, R1
CALL     145762$
BLO      145650$
CALL     146104$
BR        145650$
145716$:MOV      23602, SP
MOV      #1, R0
BR        145732$
CLR      R0
145732$:RETURN
145734$:MOV      2(R1), R1
TST      @R1
BEQ      145756$
MOV      6(R1), 23564
BEQ      145734$
BR        145760$
145756$:SEC
145760$:RETURN
145762$:MOV      4(R1), R1
TST      @R1
BEQ      146004$
MOV      10(R1), 23564
BEQ      145762$
BR        146006$
146004$:SEC
146006$:RETURN
146010$:MOV      0(R1), 23560
MOV      #1, 23572
MOV      23474, 23574
INC      23574
MOV      #0, 23576
MOV      #2, 23600
MOV      #-2, 23606
MOV      #2, 23610
MOV      R1, 23554
ADD      #2, R1
CALL     146200$
RETURN
146104$:MOV      0(R1), 23560
MOV      #-1, 23572
MOV      #0, 23574
MOV      23474, 23576

```

```

        INC      23576
        MOV      #-2, 23600
        MOV      #-4, 23606
        MOV      #4, 23610
        MOV      R1, 23554
        ADD      #4, R1
        CALL     146200$
        RETURN
146200$: MOV      23564, R2
        MOV      23566, R3
        MOV      23502, R5
146214$: MOV      (R2)+, (R3)+
        SOB      R5, 146214$
        MOV      20(R1), 23616
        MOV      10(R1), 23620
        MOV      14(R1), 23562
        MOV      23570, R0
        MOV      23502, R5
146252$: CLR      (R0)+
        SOB      R5, 146252$
        MOV      10(R1), R5
        MOV      20(R1), R0
        SUB      R0, R5
        ASR      R5
        INC      R5
        MOV      23570, R2
        ADD      R0, R2
        ADD      23562, R0
        MOV      R2, -(SP)
        MOV      R0, -(SP)
146312$: CALL     147760$
        BIS      R4, (R2)+
        ADD      #2, R0
        SOB      R5, 146312$
        MOV      10(R1), R5
        ADD      23564, R5
        MOV      @R5, R5
        CALL     150136$
        BIT      #100000, R5
        BEQ      146412$
        MOV      23476, R5
        SUB      10(R1), R5
        ASR      R5
        DEC      R5
        BEQ      146412$
146370$: CALL     147760$
        BIS      R4, (R2)+
        CMP      #-1, R4
        BNE      146412$
        ADD      #2, R0
        SOB      R5, 146370$
146412$: MOV      (SP)+, R0
        MOV      (SP)+, R2
        MOV      @R2, R4
        MOV      20(R1), R5
        ADD      23564, R5
        MOV      @R5, R5
        CALL     150136$
        BIT      #1, R5
        BEQ      146476$
        MOV      20(R1), R5
        BEQ      146476$
        ASR      R5
146454$: SUB      #2, R0

```


	CALL	147760\$
	BIS	R4, -(R2)
	CMP	#-1, R4
	BNE	146476\$
	SOB	R5, 146454\$
146476\$:	MOV	#23604, R5
	CALL	150246\$
	MOV	23604, R0
	MOV	23570, R3
	MOV	20(R1), R2
	ADD	R2, R0
	ADD	R2, R3
	ADD	23564, R2
	MOV	10(R1), R5
	SUB	20(R1), R5
	ASR	R5
	INC	R5
	MOV	R5, -(SP)
	BR	146556\$
146552\$:	BIS	#1, @R2
146556\$:	MOV	R5, -(SP)
	MOV	(R2)+, R5
	MOV	(R3)+, R4
	CALL	150136\$
	BIS	R5, (R0)+
	MOV	(SP)+, R5
	BIT	#100000, R4
	BNE	146606\$
	SOB	R5, 146556\$
	BR	146674\$
146606\$:	SOB	R5, 146552\$
	MOV	R0, -(SP)
	MOV	R2, -(SP)
	MOV	R3, -(SP)
	MOV	23476, R5
	SUB	10(R1), R5
	ASR	R5
	DEC	R5
	BEQ	146666\$
146634\$:	MOV	R5, -(SP)
	BIS	#1, @R2
	MOV	(R2)+, R5
	MOV	(R3)+, R4
	CALL	150136\$
	BIS	R5, (R0)+
	MOV	(SP)+, R5
	BIT	#100000, R4
	BEQ	146666\$
	SOB	R5, 146634\$
146666\$:	MOV	(SP)+, R3
	MOV	(SP)+, R2
	MOV	(SP)+, R0
146674\$:	MOV	(SP)+, R5
	BR	146706\$
146700\$:	BIS	#100000, -2(R2)
146706\$:	MOV	R5, -(SP)
	MOV	-(R2), R5
	MOV	-(R3), R4
	CALL	150136\$
	BIS	R5, -(R0)
	MOV	(SP)+, R5
	BIT	#1, R4
	BNE	146736\$
	SOB	R5, 146706\$

	BR	147002\$
146736\$:	SOB	R5, 146700\$
	TST	20(R1)
	BEQ	147002\$
	MOV	@R0, R4
146750\$:	BIT	#1, R4
	BEQ	147002\$
	BIS	#100000, -(R2)
	MOV	@R2, R5
	MOV	-(R3), R4
	CALL	150136\$
	BIS	R5, -(R0)
	CMP	R0, 23604
	BHI	146750\$
147002\$:	MOV	23604, R0
	MOV	23570, R3
	MOV	23502, R5
147016\$:	MOV	(R0)+, (R3)+
	SOB	R5, 147016\$
	MOV	23604, R5
	CALL	150470\$
	BHIS	147050\$
	MOV	23536, @R5
	MOV	R5, 23536
	JMP	147644\$
147050\$:	CALL	150716\$
	MOV	23560, R3
	ADD	23572, R3
	CMP	R3, 23574
	BEQ	147426\$
	CALL	147732\$
	CALL	150576\$
	MOV	R4, 23556
	ADD	23610, R4
	MOV	23636, 14(R4)
	ADD	23630, 14(R4)
	MOV	4(R4), R3
	BNE	147160\$
	MOV	23604, 4(R4)
	MOV	23612, 20(R4)
	MOV	23614, 10(R4)
	BR	147240\$
147160\$:	MOV	23502, R5
	MOV	23604, R0
147170\$:	BIS	(R0)+, (R3)+
	SOB	R5, 147170\$
	CMP	23612, 20(R4)
	BGE	147212\$
	MOV	23612, 20(R4)
147212\$:	CMP	23614, 10(R4)
	BLE	147230\$
	MOV	23614, 10(R4)
147230\$:	MOV	23604, R5
	CALL	150300\$
147240\$:	MOV	R4, -(SP)
	MOV	R1, -(SP)
	MOV	23612, -(SP)
	MOV	23614, -(SP)
	MOV	R4, R1
	ADD	23600, R1
	MOV	4(R1), R2
	BEQ	147412\$
	MOV	4(R4), R3
	MOV	23502, R5

```

MOV      R2, -(SP)
MOV      R3, -(SP)
147304$: MOV @R2, R0
          BIC  @R3, (R2)+
          BIC  R0, (R3)+
          SOB  R5, 147304$
          MOV  (SP)+, R3
          MOV  (SP)+, R2
          MOV  23556, 23640
          MOV  R2, R5
          CALL 150470$
          BCC  147344$
          CALL 150332$
          BR   147360$
147344$: MOV 23612, 20(R1)
          MOV 23614, 10(R1)
147360$: MOV R3, R5
          CALL 150470$
          BCC  147376$
          CALL 150332$
          BR   147412$
147376$: MOV 23612, 20(R4)
          MOV 23614, 10(R4)
147412$: MOV (SP)+, 23614
          MOV (SP)+, 23612
          MOV (SP)+, R1
          MOV (SP)+, R4
147426$: SUB 23572, 23560
          CMP 23560, 23576
          BEQ 147644$
          MOV 23566, R0
          MOV 23570, R3
          MOV 23502, R5
147460$: BIC (R0)+, (R3)+
          SOB R5, 147460$
          MOV 23570, R5
          CALL 150470$
          BCS 147644$
          MOV 23560, R3
          CALL 147732$
          CALL 150576$
          ADD 23610, R4
          ADD 23600, R4
          MOV 23636, 14(R4)
          MOV 4(R4), R3
          BNE 147554$
          MOV R4, R5
          ADD #4, R5
          CALL 150246$
          MOV 4(R4), R3
147554$: MOV 23614, R5
          SUB 23612, R5
          ASR R5
          INC R5
          MOV 23570, R2
          ADD 23612, R2
          ADD 23612, R3
147604$: BIS (R2)+, (R3)+
          SOB R5, 147604$
          CMP 23612, 20(R4)
          BGE 147626$
          MOV 23612, 20(R4)
147626$: CMP 23614, 10(R4)
          BLE 147644$

```

```

MOV      23614,10(R4)
147644$:MOV R1,R5
MOV      23554,23640
CALL     150332$
RETURN
147662$:MOV (SP)+,23642
MOV      R1,-(SP)
MOV      R2,-(SP)
MOV      R3,-(SP)
MOV      R4,-(SP)
MOV      R5,-(SP)
MOV      23642,-(SP)
RETURN
147706$:MOV (SP)+,23642
MOV      (SP)+,R5
MOV      (SP)+,R4
MOV      (SP)+,R3
MOV      (SP)+,R2
MOV      (SP)+,R1
MOV      23642,-(SP)
RETURN
147732$:MOV R3,-(SP)
DEC      R3
NEG      R3
ASL      R3
ASL      R3
ADD      23504,R3
MOV      @R3,23636
MOV      (SP)+,R3
RETURN
147760$:MOV R3,-(SP)
MOV      R5,-(SP)
CLR      R4
DEC      R4
MOV      R0,177010
MOVB     177012,R5
MOV      23464,R3
XOR      R3,R5
BICB     R5,R4
MOVB     177014,R5
MOV      23466,R3
XOR      R3,R5
BICB     R5,R4
MOVB     177015,R5
MOV      23470,R3
XOR      R3,R5
BICB     R5,R4
INC      R0
SWAB     R4
MOV      R0,177010
MOVB     177012,R5
MOV      23464,R3
XOR      R3,R5
BICB     R5,R4
MOVB     177014,R5
MOV      23466,R3
XOR      R3,R5
BICB     R5,R4
MOVB     177015,R5
MOV      23470,R3
XOR      R3,R5
BICB     R5,R4
SWAB     R4
TST      23634

```

	BNE	150126\$
	COM	R4
150126\$:	DEC	R0
	MOV	(SP)+, R5
	MOV	(SP)+, R3
	RETURN	
150136\$:	TST	R5
	BEQ	150234\$
	COM	R4
	BEQ	150240\$
	BIC	R4, R5
	BEQ	150234\$
	MOV	R0, -(SP)
	MOV	R1, -(SP)
	MOV	R4, -(SP)
	COM	R4
	CLR	R0
150164\$:	MOV	R4, R1
	DEC	R1
	XOR	R4, R1
	BIS	R1, R4
	MOV	R4, R1
	INC	R1
	XOR	R4, R1
	BLT	150210\$
	CLC	
	ROR	R1
150210\$:	BIC	R0, R1
	BIS	R1, R0
	BIT	R1, R5
	BEQ	150222\$
	BIS	R1, R5
150222\$:	BIC	R0, R4
	BNE	150164\$
	BIC	(SP)+, R5
	MOV	(SP)+, R1
	MOV	(SP)+, R0
150234\$:	MOV	R5, R4
	RETURN	
150240\$:	COM	R4
	MOV	R4, R5
	RETURN	
150246\$:	MOV	23536, @R5
	BNE	150260\$
	JMP	145716\$
150260\$:	MOV	@23536, 23536
	MOV	R5, -(SP)
	MOV	@R5, R5
	CLR	@R5
	MOV	(SP)+, R5
	RETURN	
150300\$:	MOV	R3, -(SP)
	MOV	23476, R3
	ADD	R3, R5
	ASR	R3
150312\$:	CLR	-(R5)
	SOB	R3, 150312\$
	MOV	23536, @R5
	MOV	R5, 23536
	MOV	(SP)+, R3
	RETURN	
150332\$:	MOV	R5, -(SP)
	MOV	4(R5), R4
	BEQ	150372\$

```

MOV      R5, -(SP)
MOV      R4, R5
CALL     150300$
MOV      (SP)+, R5
CLR      4(R5)
MOV      #10000, 20(R5)
CLR      10(R5)
150372$: MOV 23640, R5
TST      6(R5)
BNE      150464$
TST      10(R5)
BNE      150464$
MOV      R2, -(SP)
MOV      R3, -(SP)
MOV      2(R5), R2
MOV      4(R5), R3
MOV      R3, 4(R2)
MOV      R2, 2(R3)
CLR      2(R5)
CLR      4(R5)
MOV      (SP)+, R3
MOV      (SP)+, R2
MOV      23506, 0(R5)
MOV      R5, 23506
150464$: MOV (SP)+, R5
RETURN
150470$: MOV R5, -(SP)
MOV      R2, -(SP)
MOV      R3, -(SP)
MOV      R5, R2
MOV      23502, R3
150504$: TST (R2)+
BNE      150524$
SOB      R3, 150504$
MOV      (SP)+, R3
MOV      (SP)+, R2
MOV      (SP)+, R5
SEC
RETURN
150524$: SUB #2, R2
MOV      R2, 23612
MOV      23476, R3
ADD      R3, R5
ASR      R3
150544$: TST -(R5)
BNE      150552$
SOB      R3, 150544$
150552$: MOV R5, 23614
MOV      (SP)+, R3
MOV      (SP)+, R2
MOV      (SP)+, R5
SUB      R5, 23612
SUB      R5, 23614
RETURN
150576$: MOV R2, -(SP)
MOV      R3, -(SP)
MOV      R5, -(SP)
MOV      #23540, R4
150610$: MOV 2(R4), R5
TST      @R5
BEQ      150634$
CMP      0(R5), R3
BGT      150634$
BEQ      150704$

```

```

        MOV     R5, R4
        BR      150610$
150634$: MOV     23506, R2
        BNE     150646$
        JMP     145716$
150646$: MOV     @R2, 23506
        MOV     R3, 0(R2)
        MOV     2(R4), 2(R2)
        MOV     R2, 2(R4)
        MOV     4(R5), 4(R2)
        MOV     R2, 4(R5)
        MOV     R2, R5
150704$: MOV     R5, R4
        MOV     (SP)+, R5
        MOV     (SP)+, R3
        MOV     (SP)+, R2
        RETURN
150716$: MOV     R2, -(SP)
        MOV     R3, -(SP)
        MOV     R5, -(SP)
        MOV     23612, R2
        MOV     23614, R5
        SUB     R2, R5
        ADD     #2, R5
        MOV     R2, R3
        ADD     23604, R3
        ADD     14(R1), R2
150754$: MOV     R2, 177010
        MOVB    (R3)+, 177024
        INC     R2
        SOB     R5, 150754$
        MOV     (SP)+, R5
        MOV     (SP)+, R3
        MOV     (SP)+, R2
        RETURN
151000$: MOV     23652, R3
        CALL    150576$
        MOV     R4, 23624
        MOV     #23604, R5
        CALL    150246$
        MOV     23604, 6(R4)
        MOV     R3, 0(R4)
        MOV     R3, 23626
        MOV     23654, R0
        DEC     R0
        MOV     R0, R1
        MOV     #3, R3
151056$: CLC
        ROR     R1
        SOB     R3, 151056$
        BIC     #1, R1
        MOV     R1, 22(R4)
        MOV     R1, 12(R4)
        ADD     R1, 23604
        MOV     0(R4), R3
        CALL    147732$
        MOV     23636, 16(R4)
        MOV     23636, 23562
        MOV     #1, R2
        BIC     #177760, R0
        BEQ     151150$
151142$: CLC
        ROL     R2
        SOB     R0, 151142$

```

```

151150$:MOV      R2,@23604
          MOV      R2,23622
          CALL     151166$
          RETURN
151166$:MOV      SP,23656
          MOV      23624,R1
          MOV      22(R1),R0
          MOV      R0,23616
          MOV      R0,23620
          MOV      6(R1),R2
          ADD      R0,R2
          ADD      23562,R0
          MOV      R0,-(SP)
          MOV      R2,-(SP)
          MOV      R3,-(SP)
          CALL     147760$
          TST      R4
          BNE      151246$
          JMP      151640$
151246$:MOV      23622,R5
          CALL     150136$
          TST      R4
          BNE      151266$
          JMP      151640$
151266$:MOV      R4,(R2)+
          ADD      #2,R0
          BIT      #100000,R4
          BEQ      151374$
          MOV      23476,R5
          SUB      23616,R5
          ASR      R5
          DEC      R5
          BEQ      151374$
151320$:CALL     147760$
          CMP      R4,#-1
          BNE      151352$
          MOV      R4,(R2)+
          ADD      #2,R0
          ADD      #2,23620
          SOB      R5,151320$
          BR       151374$
151352$:MOV      #1,R5
          CALL     150136$
          MOV      R4,(R2)+
          BEQ      151374$
          ADD      #2,23620
151374$:MOV      (SP)+,R3
          MOV      (SP)+,R2
          MOV      (SP)+,R0
          BIT      #1,@R2
          BEQ      151474$
          MOV      23616,R5
          ASR      R5
          BEQ      151474$
151420$:SUB      #2,R0
          CALL     147760$
          CMP      R4,#-1
          BNE      151452$
          MOV      R4,-(R2)
151440$:SUB      #2,23616
          SOB      R5,151420$
          BR       151474$
151452$:MOV      #100000,R5
          CALL     150136$

```



```

MOV      R4, -(R2)
BEQ      151474$
SUB      #2, 23616
151474$: MOV 23616, 22(R1)
MOV      23620, 12(R1)
MOV      0(R1), R3
DEC      R3
BLE      151634$
CALL     147732$
CALL     150576$
MOV      R3, 0(R4)
MOV      #23564, R5
CALL     150246$
MOV      12(R1), R5
SUB      22(R1), R5
ASR      R5
INC      R5
MOV      6(R1), R2
MOV      23564, R3
ADD      22(R1), R2
ADD      22(R1), R3
151600$: MOV (R2)+, (R3)+
SOB      R5, 151600$
MOV      22(R1), 24(R4)
MOV      12(R1), 14(R4)
MOV      23564, 10(R4)
MOV      23636, 20(R4)
151634$: CLC
BR       151646$
151640$: MOV 23656, SP
SEC
151646$: RETURN
151650$: CALL 147662$
CALL     151756$
MOV      #-1, 23634
CALL     145624$
MOV      23646, R1
MOV      23644, R0
CALL     176300$
CALL     147706$
RETURN
151714$: CALL 147662$
CALL     151756$
CLR      23634
CALL     145624$
MOV      23646, R1
MOV      23644, R0
CALL     176300$
CALL     147706$
RETURN
151756$: MOV 23414, 23652
INC      23652
MOV      23416, 23654
INC      23654
MOV      23442, R1
CLR      23464
ROR      R1
SBC      23464
CLR      23466
ROR      R1
SBC      23466
CLR      23470
ROR      R1
SBC      23470

```

```

MOV      23450, R1
NEG      R1
MOVB     R1, 177016
MOV      23432, 23504
MOV      22664, 23474
MOV      23412, 23500
BIC      #17, 23500
MOV      23500, 23476
ASR      23476
ASR      23476
ASR      23476
MOV      23476, 23502
ASR      23502
RETURN
152144$: MOV      #5322, R0
CALL     176132$
MOV      R0, 23644
MOV      R1, 23646
MOV      23644, R1
ASR      R1
SUB      23476, R1
BLE      152456$
MOV      23502, R2
ADD      #13, R2
CLR      R3
BR       152216$
152214$: INC      R3
152216$: SUB      R2, R1
BGT      152214$
TST      R3
BEQ      152456$
CMP      R3, #31
BLE      152240$
MOV      #27, R3
152240$: MOV      R3, 23650
MOV      23646, R1
INC      R1
BIC      #1, R1
MOV      23502, R2
MOV      R1, 23566
152266$: CLR      (R1)+
SOB      R2, 152266$
MOV      R1, 23570
MOV      23502, R2
152302$: CLR      (R1)+
SOB      R2, 152302$
MOV      R1, - (SP)
MOV      #23510, R2
MOV      R2, R1
MOV      R2, 23506
MOV      #11, R3
152326$: CLR      (R1)+
SOB      R3, 152326$
MOV      #10000, (R1)+
MOV      #10000, (R1)+
MOV      (SP)+, R1
MOV      23650, R5
DEC      R5
BEQ      152402$
152354$: MOV      R1, @R2
MOV      R1, R2
MOV      #11, R3
152364$: CLR      (R1)+
SOB      R3, 152364$

```

```

MOV      #10000, (R1)+
MOV      #10000, (R1)+
SOB      R5, 152354$
152402$: MOV      #23536, R2
MOV      23650, R5
152412$: MOV      R1, @R2
MOV      R1, R2
MOV      23502, R3
152422$: CLR      (R1)+
SOB      R3, 152422$
SOB      R5, 152412$
MOV      #23540, R1
CLR      (R1)+
MOV      #23546, (R1)+
CLR      (R1)+
CLR      (R1)+
CLR      (R1)+
MOV      #23540, @R1
BR       152460$
152456$: SEC
152460$: RETURN

```

```

/ *****
/ *
/ *
/ * ПРОГРАММА НАЧАЛЬНОГО ЗАПУСКА СИСТЕМЫ, ПУЛЬТОВЫЙ МОНИТОР, *
/ * ЗАГРУЗЧИКИ, ТЕСТОВЫЕ ПОДПРОГРАММЫ, ОБРАБОТЧИКИ ПРЕРЫВАНИЙ *
/ *
/ *
/ *                               Адреса 160000 - 176673
/ *
/ *****
/

```

```

/ *****
/ *
/ *
/ * Системные команды, выполняемые только в режиме HALT
/ *
/ *
/ * Мнемоника      Код      Выполняемое действие
/ *
/ * $MFPC$          22      Копия счетчика команд --> R0
/ * $MFPS$          24      Копия слова состояния --> R0
/ * $MTPC$          32      R0 --> Копия счетчика команд
/ * $MTPS$          34      R0 --> Копия слова состояния
/ * $MFPM$          21      (R5)+ --> R0 (режим USER)
/ * $MTPM$          31      R0 --> -(R5) (режим USER)
/ * $STEP$          16      Копия счетчика команд --> PC
/ *                  Копия слова состояния --> PS
/ *                  Запуск по значению PC с маскировкой
/ *                  всех прерываний на один шаг
/ * $RUN$           12      Копия счетчика команд --> PC
/ *                  Копия слова состояния --> PS
/ *                  Запуск по значению PC
/ *****
/

```

```

$MFPC$ = 22
$MFPS$ = 24
$MTPC$ = 32
$MTPS$ = 34
$MFPM$ = 21
$MTPM$ = 31
$STEP$ = 16
$RUN$  = 12

```

```

.=160000

```

```

160000$: .WORD    160300$, 200      ; Вектор прерывания по включению питания
160004$: .WORD    160230$, 600      ; Вектор прерывания по зависанию (HALT)
160010$: .WORD    165612$, 600      ; Вектор прерывания по инструкции FIS
; Адреса подпрограмм обработки прерываний EMT 14 - EMT 24
160014$: .WORD    173502$, 160270$, 103564$, 104132$, 104132$
; "Пустые" адреса
160026$: .REPT    5
        .WORD     160270$
        .ENDR
; Адреса подпрограмм обработки прерываний EMT 40 - EMT 76
160040$: .WORD    107754$, 117126$, 117162$, 117174$
        .WORD     117406$, 117250$, 117410$, 117240$
        .WORD     117412$, 117414$, 117416$, 117420$
        .WORD     117422$, 160270$, 117424$, 117426$
; "Пустые" адреса
160100$: .REPT    30
        .WORD     160270$
        .ENDR
; Адреса подпрограмм обработки прерываний TRAP 0 - TRAP 6
160160$: .WORD    174254$, 175516$, 175664$, 160270$
160170$: .WORD    160240$, 600      ; Вектор прерывания по команде HALT
160174$: .WORD    160250$, 600      ; Вектор прерывания по двойному зависанию
; Точка входа обработчика прерывания 4
160200$: MOV      #163627$, @172662$
        BR        160266$
; Точка входа обработчика прерывания 10
160210$: MOV      #163641$, @172662$
        BR        160614$
; Точка входа обработчика прерывания 24
160220$: MOV      #163657$, @172662$
        BR        160614$
; Точка входа обработчика прерывания 160004
160230$: MOV      #163627$, @172662$
        BR        160574$
; Точка входа обработчика прерывания 160170
160240$: MOV      #163731$, @172662$
        BR        160614$
; Точка входа обработчика прерывания 160174
160250$: MOV      #163707$, @172662$
        BR        160614$
; Точка входа обработчика прерывания 160274
160260$: MOV      #163736$, @172662$
160266$: BR        160614$
160270$: RETURN
160272$: .WORD    161236$           ; Адрес п/п снятия режима HALT (для ЦП)
160274$: .WORD    160260$, 600      ; Вектор прерывания по зависанию при приеме АВП
; Точка входа при включении питания (адрес начального пуска)
160300$: MOV      @#172660$, R4      ; R4 = адрес ОЗУ пультового отладчика
        CLR       R0                ; Признак ошибок = 0
        MOV       R4, SP
        BMI       160464$           ; Запуск в ЦП
        BIT       #20, @#177716     ; ЦП в режиме HALT ?
        BEQ       160332$           ; Нет
        MOV       @#0, R0
        BEQ       160332$
        JMP       @R0
160332$: MOV      #40, @#177716      ; Останов ЦП (установка DCLO и ACLO)
        CALL     173252$            ; Перекачка пультового монитора в сист. ОЗУ ЦП
        MOV      #70045, @#177010   ; Передача ключа теста памяти (делать или нет)
        MOV      42(R4), @#177014
        CLR      @#177716           ; Снятие DCLO
        MOV      #100, R0
160370$: SOB      R0, 160370$        ; Задержка
        MOV      #100000, @#177716 ; Запуск ЦП (снятие ACLO)

```

```

CALL    160410$      ; Тест ПЗУ ПП
SWAB    R0
BR      160474$      ; Переход на тест ОЗУ
; Подпрограмма теста ПЗУ ПП
160410$:MOV    #4,R5      ; R5 = число банков ПЗУ
MOV     #176776,R1      ; R1 = конечный адрес ПЗУ
MOV     #7377,R2        ; R2 = число слов последнего банка ПЗУ
CLR     R0              ; Очистка признака ошибок
160426$:ASL    R0          ; Освободить правый бит
CLR     R3
160432$:ADD    -(R1),R3    ; Вычисление контрольной суммы
ADC     R3
SOB     R2,160432$
ASL     R5
CMP     <176770$-2>(R5),R3 ; Сравнение с записанной в ПЗУ
BEQ     160452$          ; Совпадает
INC     R0              ; Установить правый бит
160452$:ASR    R5
MOV     #10000,R2        ; R2 = число слов в 1,2 и 3 банках ПЗУ
SOB     R5,160426$
RETURN
160464$:CMP    42(R4),#125252 ; Производить тест ОЗУ
BEQ     160534$          ; Нет
160474$:CLR    R1          ; R1 = начальный адрес
MOV     @#172674$,R2      ; R2 = объем ОЗУ в словах
160502$:MOV    R1,@R1      ; Занести в ячейку ее адрес
CMP     R1,@R1          ; Сравнить
BNE     160530$          ; Ошибка
COM     @R1              ; Инвертировать содержимое ячейки
MOV     @R1,R3           ; R3 = содержимое ячейки
COM     R3
CMP     R1,R3           ; Содержимое R3 равно адресу
BNE     160530$          ; Нет (ошибка)
CLR     (R1)+           ; Очистить ячейку
SOB     R2,160502$
BR      160534$
160530$:BIS    #2,R0      ; Установить признак ошибки ОЗУ
160534$:MOV    R4,R1      ; R1 = адрес ОЗУ пультового отладчика
MOV     #50,R2           ; R2 = объем ОЗУ пультового отладчика
160542$:CLR    (R1)+      ; Цикл очистки
SOB     R2,160542$
MOV     @#172672$,14(R4) ; Начальное значение указателя стека
MOV     #200,20(R4)      ; Начальное значение слова состояния
MOV     #125252,42(R4)   ; Ключ пропуска теста ОЗУ
JMP     172676$
160574$:BR     160744$
; Точка перехода из режима УСТАНОВКА при нажатии УПР-@ (пультовый монитор ПП)
160576$:MOV    #600,-(SP) ; Записать в стек слово состояния
MOV     #160744$,-(SP)   ; Записать в стек адрес начала пульт. отл.
CLR     @172662$         ; Адрес выводимой надписи = 0
RTI      ; Переход в пультовый отладчик
; Точка входа при обработке системных прерываний (адрес векторов 160***)
160614$:CLR    @172664$   ; Адрес открытой ячейки = 0
MOV     R0,@172660$      ; Сохранить содержимое R0
MOV     @#172660$,R0     ; R0 = адрес буфера сохранения
TST     (R0)+
MOV     R1,(R0)+         ; Сохранить содержимое других регистров
MOV     R2,(R0)+
MOV     R3,(R0)+
MOV     R4,(R0)+
MOV     R5,(R0)+
MOV     SP,(R0)+
MOV     R0,R1
$MFPC$      ; Прочитать КРСК

```

	MOV	R0, (R1)+	; Сохранить содержимое PC
	MOV	R0, R3	; R3 = PC
	\$MFPS\$; Прочитать КРСР
	MOV	R0, (R1)+	; Сохранить содержимое PS
	MOV	@#172660, R1	
	MOV	R1, R4	; R4 = начальный адрес ОЗУ пультного отладчика
	ADD	34(R4), PC	; Режим STEP/RUN ?
	BR	160744\$; STEP
160676\$:	MOV	#4, R2	; R2 = число точек останова
	SUB	#2, R3	; R3 = предполагаемый адрес останова
160706\$:	MOV	50(R1), R5	; R5 = адрес точки останова
	BEQ	160736\$; Отсутствует
	MOV	60(R1), R0	; R0 = значение яч. памяти по адресу останова
	ADD	#2, R5	
	\$MTPM\$; Запись значения в ОЗУ
	CMP	R3, R5	; Текущий адрес останова = точке останова ?
	BNE	160736\$; Нет
	MOV	R3, 16(R4)	; PC = подкоррект. адрес останова
160736\$:	ADD	#2, R1	; Переход к следующей точке останова
	SOB	R2, 160706\$	
160744\$:	MOV	@#172660\$, R4	; R4 = начальный адрес ОЗУ пультного отладчика
	MOV	R4, SP	
	BMI	160762\$; Исполнение в ЦП
	MOV	#2, @#22550	; Запрет перевода ЦП в режим HALT при Esc Гр-]
160762\$:	MOV	#70, 26(R4)	; Установка указателя на начало буфера
	ADD	R4, 26(R4)	
	MTPS	@#172670\$; Установка ССП
	JSR	R4, 163006\$; Установка системного режима клавиатуры
	.WORD	163476\$	
	TST	R4	; Программа выполняется под упр. ПП ?
	BPL	161030\$; Да
	MOV	36(R4), R0	; R0 = выход в меню загрузки/монитор
	MOV	#2, 36(R4)	; Установка выхода в монитор
	ADD	R0, PC	
	BR	161160\$; Переход на вывод меню загрузки
161030\$:	CALL	163316\$; Вывод причины останова, адр. ячейки и ее сод.
161034\$:	MOV	R4, SP	
	JSR	R4, 163006\$; Вывод промптера /@ /
	.WORD	163510\$	
161044\$:	CALL	162602\$; Набор восьм. значения и запрос ком. клавиши
	BIC	#240, R0	; Привести к верхнему регистру
	BNE	161062\$; Нажат не <Пробел>
	JMP	161606\$; Нажат <Пробел>
161062\$:	CMPB	R0, #3	; Нажата комбинация УПР-С
	BEQ	161206\$	
	CMP	R0, #14	; Нажат <СБРОС>
	BEQ	161300\$	
	CMP	R0, #15	; Нажат <Enter>
	BEQ	161034\$	
	CMP	R0, #33	; Код <Escape>
	BEQ	161306\$	
	CMP	R0, #'R	; R (переключение на регистры)
	BEQ	161320\$	
	CMP	R0, #'A	; A (переключение на точки останова)
	BEQ	161330\$	
	CMP	R0, #'M	; M (выход в меню)
	BEQ	161160\$	
	CMP	R0, #'T	; T (тест сетевого адаптера)
	BEQ	161154\$	
	CMP	R0, #'D	; D (переход в автономный режим)
	BNE	161034\$	
161150\$:	JMP	161530\$	
161154\$:	JMP	162460\$	

; Обработка вызова меню загрузки (М)

```
161160$:JSR      R4,163006$      ; Вызов меню загрузки
        .WORD    163505$
        CALL     172614$        ; Прочесть данные из канала 0 (номер пункта)
        MOV      R0,R1          ; R1 = номер выбранного пункта * 2
        CALL     172614$        ; Прочесть данные из канала 0 (номер уст-ва)
        CALL     @162350$(R1)    ; Вызов соответствующей п/п
        BR       161160$
```

; Обработка выхода из пультового отладчика (УПР-С)

```
161206$:TST      R4              ; Пультовый монитор работает под упр. ПП ?
        BMI      161034$        ; Нет
        CLR      @#22550
        MTPS     #600
        MOV      @#7144,SP      ; Восстановить стек режима УСТАНОВКА
        CLR      R0
        $MTPS$    ; PS = 0
        $MTPC$    ; PC = 0
        $RUN$     ; Запуск по адресу 0(там RETURN, возврат в УСТ)
```

; П/п снятия в ЦП сигнала HALT

```
161236$:MOV      R0,-(SP)
        MOV      R1,-(SP)
        MOV      R4,-(SP)
        MOV      @#172660$,R4   ; R4 = начальный адрес ОЗУ пультового отладчика
        INC      44(R4)         ; Установить ключ продолжения
        JSR      R4,163006$     ; Снять сигнал HALT в ЦП
        .WORD    163501$
161262$:TST      44(R4)         ; Ключ продолжения сброшен ?
        BNE      161262$        ; Нет
        MOV      (SP)+,R4
        MOV      (SP)+,R1
        MOV      (SP)+,R0
        RETURN
```

; Обработка команды очистки экрана (<СБРОС>) (R0 = 14)

```
161300$:CALL     172632$        ; Вывод символа УПР-L
        BR       161034$
```

; Обработка клавиш с префиксом Esc

```
161306$:CALL     162756$        ; Прочесть очер. код клавиши с его коррекцией
        CALL     @161556$(R0)   ; Вызвать соответствующую п/п
        BR       161044$
```

; Обработка команды переключения на регистры (R)

```
161320$:MOV      #10*400+'R,32(R4); Режим работы (10(8) регистров)
        BR       161336$
```

; Обработка команды переключения на точки останова (A)

```
161330$:MOV      #4*400+'A,32(R4); Режим работы (4 точки останова)
161336$:CALL     163370$        ; Перевод курсора на новую строку
        MOV      32(R4),R0      ; R0 = 'A' или 'R'
        CALL     172632$        ; Вывод символа
161352$:CALL     172614$        ; Ввод символа с клавиатуры
        MOV      R0,R3
        SUB      #60,R3         ; R3 в диапазоне '0' -- '7'
        BMI      161402$        ; Нет
        CMP      R3,#7
        BLOS     161410$        ; В диапазоне
        CMP      R0,#'S         ; Нажата 'S' ?
        BEQ      161470$        ; Да
161402$:CALL     163142$        ; Вывод всех регистров или точек останова
        BR       161034$
161410$:BICB     33(R4),R0
        CALL     172632$        ; Вывод номера регистра или точки останова
161420$:CALL     163234$        ; Вывод '=' и содержимого регистра
161424$:JSR      R4,163006$     ; Вывод двух пробелов
        .WORD    163560$
        CALL     162604$        ; Ввод восьм. числа и запрос ком. клавиши
        CMP      R0,#33         ; Ком. клавиша имеет префикс Esc ?
```

```

BNE      161512$      ; Нет
CALL     162756$      ; Прочсть постфикс ком. клавиши и подкорр. его
SUB      #2, R0
BMI      161524$      ; Нажата клавиша <ПОМ>
CALL     163264$      ; Получение адреса буфера
CALL     @162222$(R0) ; Вызов соответствующей п/п
BR       161424$
161470$:CMPB 32(R4), #'R ; Работа производится с регистрами ?
BNE      161352$      ; Нет
CALL     172632$      ; Вывод символа 'S' (R0 = 'S')
MOV      #10, R3      ; R3 = номер регистра (PS)
BR       161420$
161512$:CMP  R0, #40    ; Нажат <Пробел> ?
BNE      161524$      ; Нет
JMP      161606$      ; Да
161524$:JMP  161062$
; Обработка команды перехода в автономный режим (D)
161530$:MOV  #14, R0    ; R0 = Ctrl+L (очистка экрана)
161534$:CALL 172632$    ; Вывести символ на экран
CALL     172614$      ; Ввести символ с клавиатуры
CMP      R0, #3        ; Нажата комбинация УПР+С ?
BNE      161534$      ; Нет
JMP      161034$      ; Переход в пультовый монитор
; Адреса п/п обработки клавиш с префиксом Esc (ПОМ, ИСП, Вверх, Вниз, Влево,
; Вправо, НР+Вверх, НР+Вниз, НР+Влево, НР+Вправо, ?, ?)
161556$: .WORD 162572$, 161612$, 161762$, 161770$
        .WORD 161772$, 162026$, 162132$, 162040$
        .WORD 162132$, 162046$, 162164$, 162204$
; Обработка команды "шаг" (<Пробел>)
161606$:CLR  R3        ; R3 = режим STEP
BR       161616$
; Обработка команды "пуск" (<ИСП>)
161612$:MOV  #2, R3    ; R3 = режим RUN
161616$:MTPS #600
        MOV    20(R4), R0 ; R0 = PS
        $MTPS$ ; Запись КРСР
        TST    R2        ; Было набрано значение ?
        BEQ    161640$    ; Нет
        MOV    R1, 16(R4) ; Сохранить набранное значение
161640$:MOV  16(R4), R0 ; R0 = РС
        $MTPC$ ; Запись КРСР
        MOV    R3, 34(R4) ; Сохранение режима STEP/RUN
        BEQ    161724$    ; Режим STEP
        MOV    R4, R1    ; R1 = начальный адрес ОЗУ пультового отладчика
        MOV    #4, R2    ; R2 = число точек останова
161662$:MOV  50(R1), R5 ; R5 = адрес точки останова
        BEQ    161712$    ; Точка останова не используется
        $MFFPM$ ; Чтение содерж. памяти по адресу останова
        MOV    R0, 60(R1) ; Сохранение его в буфере
        CMP    16(R4), 50(R1) ; Адрес останова равен адресу запуска ?
        BEQ    161712$    ; Да
        CLR    R0        ; R0 = код команды HALT
        $MTPM$ ; Запись его в память по адресу останова
161712$:ADD  #2, R1    ; Переход к следующей точке останова
        SOB    R2, 161662$
        CALL   161236$    ; Снятие в ЦП сигнала HALT
161724$:MOV  R4, SP    ; SP = начальный адрес ОЗУ пультового отладчика
        BMI    161734$    ; Пультовый отладчик работает под управл. ЦП
        CLR    @#22550
161734$:MOV  (SP)+, R0 ; Восстановление регистров
        MOV    (SP)+, R1
        MOV    (SP)+, R2
        MOV    (SP)+, R3
        MOV    (SP)+, R4

```



```

162220$:RETURN
; Адреса п/п обработки клавиш с префиксом Esc для режима рег./точек останова
; (ИСП, Вверх, Вниз, Влево, Вправо, НР+Вверх, НР+Вниз, НР+Влево, НР+Вправо,?,?)
162222$: .WORD 161612$,162250$,162262$,162304$,162320$
           .WORD 162342$,162334$,162342$,162326$,162164$,162204$
; Обработка команды "открыть предыдущий регистр/точку останова" (Вверх)
162250$:TST R2 ; Значение было введено ?
           BEQ 162256$ ; Нет
           MOV R1,@R5 ; Занести введенное значение в рег./точку ост.
162256$:DEC R3 ; Перейти к предыдущему регистру/точке останова
           BR 162272$
; Обработка команды "открыть следующий регистр/точку останова" (Вниз)
162262$:TST R2 ; Значение было введено ?
           BEQ 162270$ ; Нет
           MOV R1,@R5 ; Занести введенное значение в рег./точку ост.
162270$:INC R3 ; Перейти к следующему регистру/точке останова
162272$:BIC #177770,R3 ; Подкорректировать номер рег./точки останова
           BICB 33(R4),R3
           BR 162312$
; Обработка команды "занести значение в регистр/точку останова" (Влево)
162304$:TST R2 ; Значение было введено ?
           BEQ 162312$ ; Нет
           MOV R1,@R5 ; Занести введенное значение в рег./точку ост.
162312$:CALL 163174$ ; Вывод содержимого другого рег./точки ост.
           RETURN
; Обработка команды "открыть ячейку" (Вправо)
162320$:MOV #161044$,@SP ; Коррекция адреса возврата
           BR 162026$ ; На исполнение команды
; Обработка команды "открыть ячейку с абсолютным адресом" (НР+Вправо)
162326$:MOV #161044$,@SP ; Коррекция адреса возврата
           BR 162046$ ; На исполнение команды
; Обработка команды "открыть ячейку с относительным адресом" (НР+Вниз)
162334$:MOV #161044$,@SP ; Коррекция адреса возврата
           BR 162040$ ; На исполнение команды
; Обработка команды "возврат на пред. ур. косв. адр." (НР+Вверх, НР+Влево)
162342$:MOV #161044$,@SP ; Коррекция адреса возврата
           BR 162132$ ; На исполнение команды
; Адреса подпрограмм исполнения пунктов меню ЗАГРУЗКА (не выбрано,
; диск, кассета ПЗУ, сеть, стык С2, магнитофон, отладка, тестирование)
162350$: .WORD 162572$,162400$,162420$,162440$,162472$,162504$,161034$,162370$
; Обработка пункта меню ЗАГРУЗКА [тестирование]
162370$:CALL 161236$ ; Перевод ЦП в режим USER
           JMP 167540$ ; На исполнение тестов
; Обработка пункта меню ЗАГРУЗКА [диск]
162400$:MOV R0,R3 ; R3 = номер диска (0 -- 3)
           JSR R5,162522$ ; Перекачка загрузчика и его запуск
           .WORD 163774$,44,164104$,5,0
; Обработка пункта меню ЗАГРУЗКА [кассета ПЗУ]
162420$:MOV R0,R3 ; R3 = номер кассеты ПЗУ (1 -- 2)
           JSR R5,162522$ ; Перекачка загрузчика и его запуск
           .WORD 163774$,44,164116$,5,0
; Обработка пункта меню ЗАГРУЗКА [сеть]
162440$:JSR R4,163006$ ; Вывод "*" ЗАГРУЗКА ИЗ СЕТИ "*"
           .WORD 163573$ ; Адрес надписи
           JSR R5,162522$ ; Перекачка загрузчика и его запуск
           .WORD 164160$,210,0
; Обработка команды "Т" пультавого монитора (тест локальной сети)
162460$:JSR R5,162522$ ; Перекачка программы теста и ее запуск
           .WORD 164600$,360,0
; Обработка пункта меню ЗАГРУЗКА [стык С2]
162472$:JSR R5,162522$ ; Перекачка загрузчика и его запуск
           .WORD 165540$,25,0
; Обработка пункта меню ЗАГРУЗКА [магнитофон]
162504$:JSR R5,162522$ ; Перекачка загрузчика и его запуск

```

```

        .WORD    163774$, 44, 164130$, 14, 0
; Подпрограмма перекачки загрузчика из сист. ОЗУ и его запуск
162522$:MOV     #1000, R0          ; R0 = адрес, куда перекачивается загрузчик
162526$:MOV     (R5)+, R1          ; R1 = адрес загрузчика в сист. ОЗУ
        BEQ     162542$           ; Конец перекачки
        MOV     (R5)+, R2          ; R2 = размер загрузчика в словах
162534$:MOV     (R1)+, (R0)+       ; Цикл перекачки
        SOB     R2, 162534$
        BR      162526$
162542$:CALL    161236$           ; Перевод ЦП в режим USER
        MOV     #200, R0          ; Копия ССП = 200 (прерывания запрещены)
        $MTPS$
        MOV     #1000, R0         ; Копия СК = 1000 (адрес загрузчика)
        $MTPC$
        MOV     #1000, SP
        MOV     R3, R0            ; R0 = номер устройства (диска или кассеты ПЗУ)
        $RUN$                     ; Запуск загрузчика
162572$:RETURN
; Подпрограмма ввода восьмеричного значения и запроса командной клавиши
162574$:MOV     @#172660, R4      ; R4 = начало ОЗУ пультавого монитора
        BR      162604$
162602$:CLR     R3
162604$:MOV     R4, R1
        ADD     #110, R1          ; R1 = адрес буфера для ввода чисел
        CLR     R2                ; R2 = 0 (инициализация счетчика введ. цифр)
162614$:CALL    172614$           ; Запрос нажатой клавиши
        CMP     R0, #'0           ; Введенный символ в диапазоне '0' - '7' ?
        BLO     162706$           ; Нет
        CMP     R0, #'7
        BHI     162654$           ; Нет
        CMP     R2, #8.           ; Введено восемь цифр ?
        BEQ     162614$           ; Да
        MOVB    R0, (R1)+         ; Поместить символ в буфер
        INC     R2                ; Увеличить счетчик символов
        CALL    172632$           ; Вывести символ на экран
        BR      162614$
162654$:CMPB    R0, #177           ; Нажата клавиша <ЗАБОЙ> ?
        BNE     162706$           ; Нет
        TST     R2                ; Буфер пуст ?
        BEQ     162614$           ; Да
        CLRB    -(R1)             ; Удалить последний символ из буфера
        DEC     R2                ; Уменьшить счетчик введенных символов
        MOV     R1, -(SP)
        JSR     R4, 163006$        ; Удалить последний символ с экрана
        .WORD    163551$
        MOV     (SP)+, R1
        BR      162614$
162706$:SUB     R2, R1             ; R1 = начало буфера ввода
        CLR     R2                ; R2 = 0 (счетчик обработанных символов)
        CLR     R5                ; R5 = 0 (введенное число (инициализация))
        MOV     R0, -(SP)         ; Сохранить в стеке код командной клавиши
162716$:MOVB    @R1, R0           ; R0 = очередная цифра из буфера
        CLRB    (R1)+
        SUB     #60, R0            ; Перевод из ASCII в BIN
        BMI     162750$           ; Достигнут конец буфера
        ASL     R5                ; R5 = R5 * 8 + R0
        ASL     R5
        ASL     R5
        ADD     R0, R5
        INC     R2                ; Увеличить счетчик обработанных символов
        CMP     R2, #8.           ; Обработано восемь символов ?
        BNE     162716$           ; Нет
162750$:MOV     (SP)+, R0         ; R0 = код командной клавиши
        MOV     R5, R1            ; R1 = введенное значение

```

```

RETURN
; П/п ввода клавиши и корректировка ее кода (за символом ESC)
162756$:CALL 172614$ ; Запрос нажатой клавиши
SUB #176,R0 ; Ее код в диапазоне 176(8) -- 207(8) ?
BMI 163002$ ; Нет
CMP R0,#11
BHI 163002$ ; Нет
ASL R0 ; Увеличение скорректированного кода вдвое
RETURN
163002$:CLR R0 ; R0 = 0 ("плохой" код)
RETURN
; Подпрограмма вывода строки на экран (адрес строки за командой JSR R4,...)
163006$:MOV (R4)+,R1 ; R1 = адрес строки
BR 163016$
163012$:CALL 172632$ ; Вывод символа на экран
; Подпрограмма вывода строки на экран (адрес строки в регистре R1)
163016$:MOVB (R1)+,R0 ; R0 = очередной символ
BNE 163012$ ; Еще не конец строки
RTS R4
; Подпрограмма вывода на экран адреса открытой ячейки памяти и ее содержимого
163024$:CALL 163370$ ; Перевод курсора в начало следующей строки
MOV 22(R4),R1 ; R1 = адрес открытой ячейки
CALL 163074$ ; Вывод на экран адреса открытой ячейки
JSR R4,163006$ ; Вывод разделительной дробной черты
.WORD 163516$
MOV 22(R4),R5 ; R5 = адрес открытой ячейки
CALL 163412$ ; Запрос содержимого ячейки
MOV R0,R1 ; R1 = содержимое открытой ячейки
CALL 163074$ ; Вывод на экран содержимого открытой ячейки
JSR R4,163006$ ; Вывод двух пробелов
.WORD 163560$
RETURN
; Подпрограмма вывода на экран числа в восьмеричной форме
163074$:MOV R1,24(R4) ; Сохранение значения в ОЗУ пульт. отладчика
CLR R0
MOV #6,R2 ; Число символов в ASCII-строке
BR 163122$
163110$:CLR R0 ; Цикл вытяжки трех старших разрядов R1 -> R0
ROL R1
ROL R0
ROL R1
ROL R0
163122$:ROL R1
ROL R0
ADD #'0,R0 ; Преобразование из BIN в ASCII
CALL 172632$ ; Вывод символа на экран
SOB R2,163110$
RETURN
; Подпрограмма вывода на экран регистров или точек останова
163142$:CALL 163370$ ; Перевод курсора в начало следующей строки
CLR R3 ; R3 = 0 (начальный номер рег. или точ.ост.)
163150$:CALL 163200$ ; Вывод содержимого регистра или точ. ост.
INC R3 ; Перейти к след. регистру или точ. ост.
CMPB R3,33(R4) ; Выведены все рег. или точ. ост.
BLO 163150$ ; Нет
CMP R3,#10 ; Номер регистра равен 10 (рег. сост. проц.) ?
BEQ 163150$ ; Да
RETURN
; Подпрограмма вывода на экран содерж. одного регистра или точки останова
163174$:CALL 163370$ ; Перевод курсора в начало следующей строки
163200$:MOVB 32(R4),R0 ; R0 = 'R' или 'A' (регистры или точ. ост.)
CALL 172632$ ; Вывод символа на экран
MOV R3,R0 ; R0 = номер регистра или точ. ост.
ADD #'0,R0 ; Перевод из BIN в ASCII

```

```

        CMP      R0, #'8          ; Это регистр состояния процессора
        BNE      163230$         ; Нет
        MOV      #'S, R0         ; Замена '8' на 'S'
163230$: CALL    172632$         ; Вывод символа на экран
163234$: JSR     R4, 163006$      ; Вывод знака равенства
        .WORD    163526$
        CALL     163264$         ; Получение в R5 адреса хранения знач. рег.
        MOV      @R5, R1         ; R1 = содержимое регистра или точ. ост.
        CALL     163074$         ; Вывод содержимого на экран
        JSR      R4, 163006$     ; Вывод разделительного пробела
        .WORD    163561$
        RETURN

; Подпрограмма получения в R5 адреса хранения содержимого рег. или точ. ост.
163264$: CLR     R5              ; R5 = 0 (инициализация)
        CMPB     33(R4), #4      ; Идет вывод точек останова ?
        BNE      163306$         ; Нет
        BICB     33(R4), R3
        MOV      #24, R5         ; R5 = смещение знач. точ. ост. в ОЗУ ПО / 2
163306$: ADD     R3, R5          ; Прибавить номер регистра или точки останова
        ASL      R5
        ADD      R4, R5          ; Прибавить адрес ОЗУ пультового отладчика
        RETURN

; Подпрограмма вывода на экран причины перехода процессора в режим HALT
163316$: MOV     @172662$, R2    ; R2 = адрес строки причины перехода
        BEQ      163366$
        JSR      R4, 163006$     ; Вывод "**** "
        .WORD    163530$
        MOV      R2, R1         ; R1 = адрес строки причины перехода
        JSR      R4, 163016$     ; Вывод причины перехода
        JSR      R4, 163006$     ; Вывод " ****"
        .WORD    163540$
        TST      22(R4)         ; Вывод содержимое ячейки останова ?
        BNE      163366$         ; Нет
        MOV      16(R4), 22(R4) ; 22(R4) = адрес останова
        CALL     163024$         ; Вывод содержимого ячейки адреса останова
163366$: RETURN

; Подпрограмма перевода курсора в начало следующей строки
163370$: MOV     #15, R0         ; R0 = код возврата каретки
        CALL     172632$         ; Вывод на экран (курсор в начало тек. строки)
        MOV      #12, R0         ; R0 = код перевода строки
        CALL     172632$         ; Вывод на экран (курсор на след. строку)
        RETURN

; Подпрограмма чтения содержимого ячейки памяти
163412$: ADD     40(R4), PC       ; Режим ЦП(40(R4)=4)/ПП(40(R4)=0)(только в ПП)
        $MFPMS$                ; Чтение содержимого ячейки (R5)+ -> R0
        RETURN
163422$: CLC                                ; Коррекция R5 для помещ. его в рег. адреса
        ROR      R5
        MOV      R5, @177010
        MOV      @177014, R0      ; Чтение содержимого ячейки памяти ЦП
        INC      R5
        ASL      R5
        RETURN

; Подпрограмма записи значения в ячейку памяти
163444$: ADD     40(R4), PC       ; Режим ЦП(40(R4)=4)/ПП(40(R4)=0)(только в ПП)
        $MTPMS$                ; Запись значения в ячейку памяти R0 -> -(R5)
        RETURN
163454$: CLC                                ; Коррекция R5 для помещ. его в рег. адреса
        ROR      R5
        DEC      R5
        MOV      R5, @177010
        MOV      R0, @177014     ; Запись значения в ячейку памяти ЦП
        ASL      R5
        RETURN

```

; Esc-последовательности и надписи, употребляемые пультовым отладчиком

```
163476$: .BYTE    33,275,0          ; Системный режим клавиатуры
163501$: .BYTE    33,277,275,0      ; Пользовательский режим клавиатуры
163505$: .BYTE    33,274,0          ; Вызов меню ЗАГРУЗКА
163510$: .ASCIZ   <15><12><17>/@ /
163516$: .ASCIZ   "/ "
163521$: .ASCIZ   /R/
163523$: .ASCIZ   /RS/
163526$: .ASCIZ   /=/
163530$: .ASCIZ   <15><12><16>/*** /
163540$: .ASCIZ   / ***/<17>
163546$: .ASCIZ   <15><12>
163551$: .ASCIZ   <33>/D /<33>/D/
               .EVEN
163560$: .ASCII   / /
163561$: .ASCIZ   / /
163563$: .ASCIZ   <6>/ЦП/
163567$: .ASCIZ   <6>/ПП/
163573$: .ASCIZ   <15><12><12><16>/* zagruzka iz seti * /<17>
163627$: .ASCIZ   /zawisanie/
163641$: .ASCIZ   /rezerwuj kod/
163657$: .ASCIZ   /awariq setewogo pitaniq/
163707$: .ASCIZ   /dwojnoe zawisanie/
163731$: .ASCIZ   /stop/
163736$: .ASCIZ   /zawisanie pri prieme a.w.p./
               .EVEN
163772$: .WORD    0
; Загрузчик с диска, кассеты ПЗУ, магнитофона
163774$: MOV      #176674,R1          ; R1 = адрес регистра сост. перед. канала 2
               MOVB  R0,164104$+3     ; Сохранить номер устройства
               CLRB  @R1              ; Запретить прерывания от канала 2
               MOV   #164104$-163774$+1000,R2 ; R2 = адрес блока параметров
164012$: MOV      #2,R3
164016$: TSTB     @R1                 ; Ожидание готовности канала 2
               BPL   164016$          ; Не готов
               MOVB  R2,2(R1)         ; Передача адреса БП в канал 2
               SWAB  R2
               SOB   R3,164016$
               MOV   #2,R3
164036$: TSTB     @R1                 ; Ожидание готовности канала 2
               BPL   164036$          ; Не готов
               MOVB  #377,2(R1)       ; Передача байтов завершения в канал 2
               SOB   R3,164036$
164052$: TSTB     @R1                 ; Ожидание завершения операции
               BPL   164052$          ; Операция не завершена
               TSTB  164104$          ; Есть ошибки ?
               BEQ   164072$          ; Нет
               CLRB  164104$          ; Очистить байт ошибок
               BR    164012$          ; Попробовать заново
164072$: CMP      #240,@#0            ; Первая команда BOOT-сектора NOP ?
               BNE   164012$          ; Нет
               CLR   PC               ; Запуск BOOT
; Блок параметров для диска
164104$: .BYTE    0,10,2,0,0,1
               .WORD 0,400
; Блок параметров для кассеты ПЗУ
164116$: .BYTE    0,10,21,0,0,0
               .WORD 0,400
; Блок параметров для магнитофона
164130$: .BYTE    0,11,11,0
               .WORD 0,0,0,0,0,0,0,0,0,400
; Загрузчик из сети
164160$: MTPS     #340                ; Запретить прерывания
               MOVB  @#176561,R5      ; Выделение в R5 номера сетевой станции
```

```

MOV      R5,R0
BIC      #177760,R5
ASR      R0
BIC      #177717,R0
BIS      R0,R5
MOVB     R5,164554$+2      ; Сохранение номера сетевой станции
MOVB     164554$+16,R0     ; R0 = 173(8)
XOR      R5,R0             ; R0 = 173(8) XOR номер_сетевой_станции
MOVB     R0,164554$+16
MOV      #30,R0            ; R0 = 30 (адрес вектора команды EMT)
MOV      #164500$-164160$+1000,(R0)+ ; Установка вектора прерывания EMT
CLR      (R0)+             ; Установка ССП при вызове EMT
MOV      #164512$-164160$+1000,(R0)+ ; Установка вектора прерывания TRAP
CLR      (R0)+             ; Установка ССП при вызове TRAP
164244$:MOV #10000,SP       ; Установка указателя стека
MOV      #164526$-164160$+1000,@#100 ; Устан. вект. прер. сет. таймера
CLR      @#102
MTPS     #0                ; Разрешить прерывания
CLR      164534$           ; Очистка счетчика ожидания
164272$:TRAP 0              ; Чтение пришедшего байта
CMP      164534$,#5        ; Счетчик ожидания больше пяти
BLO      164272$           ; Нет
164304$:EMT 0              ; Посылка и чтение очередного байта
CMPB     #376,R0           ; Пришедший байт равен 376(8)
BNE      164304$           ; Нет
MOV      #164554$-164160$+1000,R1 ; R1 = адрес блока передачи
164320$:MOVB (R1)+,R0       ; R0 = очередной байт из блока передачи
CMPB     #376,R0           ; Он равен 376(8) ?
BEQ      164334$           ; Да
EMT      0                 ; Посылка и чтение очередного байта
BR       164320$
164334$:EMT 0              ; Посылка и чтение очередного байта
CMPB     #377,R0           ; Прочитанный байт равен 377(8) ?
BNE      164334$           ; Нет
EMT      0                 ; Посылка и чтение очередного байта
CMPB     R5,R0             ; Полученный байт равен номеру станции ?
BNE      164334$           ; Нет
MOV      #377,R4           ; R4 = 377(8) (контрольная сумма)
XOR      R0,R4             ; R4 = 377(8) XOR номер_сетевой_станции
MOV      #50000,R1         ; R1 = адрес расположения загрузчика
EMT      0                 ; Посылка и чтение очередного байта
MOV      R0,R3             ; R3 = считанный байт
XOR      R0,R4             ; Подсчет контрольной суммы
164372$:EMT 0              ; Посылка и чтение очередного байта
CMPB     #373,R0           ; Считанный байт равен 373(8) ?
BLOS     164410$           ; Выше или равен 373(8)
XOR      R0,R4             ; Подсчет контрольной суммы
MOVB     R0,(R1)+          ; Передача принятого байта в память
BR       164372$
164410$:BNE 164426$        ; Принятый байт равен не равен 373(8)
EMT      0                 ; Посылка и чтение очередного байта
COM      R0                ; Инверсия принятого байта
XOR      R0,R4             ; Подсчет контрольной суммы
MOVB     R0,(R1)+          ; Передача принятого байта в память
COM      R0                ; Инверсия для передачи
BR       164372$
164426$:CMPB #374,R0       ; Принятый байт равен 374(8)
BNE      164244$           ; Нет
EMT      0                 ; Посылка и передача байта
CMPB     R0,R4             ; Он равен контрольной сумме ?
BNE      164244$           ; Нет
EMT      0                 ; Посылка и передача байта
BISB     #10,R0            ; R0 = R0 OR 10(8)
EMT      0                 ; Посылка и передача байта

```

```

CMPB    #377,R0          ; Принятый байт равен 377(8)
BNE     164244$          ; Нет
EMT     0                ; Посылка и передача байта
MOV     #4,@#176560      ; Включение кольцевой сети
MTPS    #340             ; Запретить прерывания
JMP     @#50000          ; Переход на исполнение загрузчика
; Подпрограмма передачи байта в локальную сеть (исполнение команды EMT)
164500$:TSTB @#176564      ; Передатчик CA готов ?
BPL     164500$          ; Нет
MOVB    R0,@#176566      ; Передача байта в сеть
; Подпрограмма чтения байта, пришедшего по сети (исполнение команды TRAP)
164512$:TSTB @#176560      ; Приемник CA готов ?
BPL     164512$          ; Нет
MOVB    @#176562,R0      ; Прием байта из сети
RTI
; Подпрограмма исполнения прерываний таймера
164526$:CLR @#176560      ; Включение режима загрузки
INC     (PC)+            ; Инкремент счетчика ожидания
164534$: .WORD 0
CMP     164534$,#1000    ; Счетчик ожидания превысил лимит времени ?
BHI     164550$          ; Да
RTI
164550$:JMP 164244$
; Блок передачи в сеть
164554$: .BYTE 377,0,0,204,0,0,0,0,0,0,0,0,0,374,173,60,377,376,0,0
; Программа теста локальной сети
164600$:MTPS #340         ; Запретить прерывания
MOV     #165130$-164600$+1000,@#360 ; Уст. вект. прер. приемника CA
MOV     #165120$-164600$+1000,@#100 ; Уст. вект. прер. сет. таймера
MOV     #165254$-164600$+1000,@#4 ; Уст. вект. прер. по ош. обр. к кан.
MOV     #165264$-164600$+1000,@#10 ; Уст. вект. прер. по недейст. ком.
MOV     #340,R0          ; Установить знач. ССП при вызове прер.
MOV     R0,@#362
MOV     R0,@#102
MOV     R0,@#6
MOV     R0,@#12
CLR     (PC)+            ; Установить пассивный режим
164662$: .WORD 0
CALL    165304$          ; Вывести номер сетевой станции
MTPS    #0               ; Разрешить прерывания
164674$:TSTB @#177560      ; Нажата клавиша ?
BPL     164674$          ; Нет
MOVB    @#177562,R0      ; R0 = код нажатой клавиши
BIC     #177600,R0
CMPB    #3,R0            ; Нажата комбинация <УПР+С>
BNE     164726$          ; Нет
CALL    164744$          ; Запрос команды и ее исполнение
BR      164674$
164726$:TST 164662$       ; Режим активный или пассивный ?
BEQ     164674$          ; Пассивный режим
MOVB    R0,R1            ; R1 = код нажатой клавиши
CALL    165104$          ; Посылка в сеть
BR      164674$
; Подпрограмма запроса команды и ее исполнение
164744$:TSTB @#177564      ; Готовность перед. канала 0 (терминал)
BPL     164744$          ; Не готов
MOVB    #'?',@#177566    ; Вывести промптер '?'
164760$:TSTB @#177560      ; Нажата клавиша ?
BPL     164760$
MOVB    @#177562,R0      ; R0 = код нажатой клавиши
BIC     #177640,R0
CMPB    R0,#'A           ; Нажата 'A' (активный режим) ?
BEQ     165046$          ; Да
CMPB    R0,#'P           ; Нажата 'P' (пассивный режим) ?

```



```

        BEQ      165056$          ; Да
        CMPB    R0,#'N          ; Нажата 'N' (номер станции) ?
        BEQ      165076$          ; Да
        CMPB    R0,#'K          ; Нажата 'K' (конец теста) ?
        BEQ      165064$          ; Да
        TST     164662$          ; Активный/пассивный режим ?
        BEQ      165102$          ; Пассивный режим
        MOVB    #3,R1
        CALL    165104$          ; Посылка в сеть
        BR      165102$
; Активный режим
165046$:MOV     #1,164662$        ; Установить признак активного режима
        BR      165102$
; Пассивный режим
165056$:CLR     164662$          ; Установить признак пассивного режима
        BR      165102$
; Конец теста
165064$:MOV     #165504$-164600$+1000,R5 ; Адрес строки "КОНЕЦ ТЕСТА"
        CALL    165402$          ; Вывести на экран
        HALT    ; Выход в пультовый отладчик
; Номер станции
165076$:CALL    165304$          ; Вывести номер станции на экран
165102$:RETURN
; Подпрограмма передачи данных в сеть
165104$:TSTB    @#176564         ; Передатчик готов ?
        BPL     165104$          ; Нет
        MOVB    R1,@#176566      ; Послать байт в сеть
        RETURN
; Подпрограмма обработки событий сетевого таймера
165120$:MOV     #100,@#176560    ; Установить режим прерываний СА
        RTI
; Подпрограмма обработки прерываний от СА
165130$:MOV     R1,-(SP)
        BIT     #110001,@#176560 ; Есть ошибки передачи ?
        BNE     165176$          ; Да
        MOVB    @#176562,R1      ; R1 = принятые данные
165146$:TSTB    @#177564         ; Готовность терминала
        BPL     165146$          ; Не готов
        MOVB    R1,@#177566      ; Вывести принятый символ на экран
        TST     164662$          ; Активный/пассивный режим
        BNE     165172$          ; Активный
        CALL    165104$          ; Послать данные в сеть
165172$:MOV     (SP)+,R1
        RTI
165176$:MOV     @#176560,R1      ; R1 = регистр состояния приемника СА
        BPL     165210$          ; Ошибка паритета отсутствует
        BIS     #4,R1            ; Установить признак ошибки паритета
165210$:BIT     #10000,R1        ; Есть ошибки переполнения ?
        BEQ     165222$          ; Нет
        BIS     #2,R1            ; Установить признак переполнения
165222$:BIC     #177770,R1       ; Выделить ошибку разрыва линии
        BIS     #60,R1           ; Перевод BIN -> ASCII
        MOVB    R1,165420$+7
        MOV     #165420$-164600$+1000,R5 ; Адрес сообщения "СБОЙ #X"
        CALL    165402$          ; Вывести сообщение на экран
        TSTB    @#176562         ; Сбросить готовность приемника СА
        BR      165172$
; Подпрограмма обработки прерывания 4 (ошибка обращения к каналу)
165254$:MOV     #"04,165461$+15 ; Номер прерывания в сообщение
        BR      165272$
; Подпрограмма обработки прерывания 10 (зарезервированная инструкция)
165264$:MOV     #"10,165461$+15 ; Номер прерывания в сообщение
165272$:MOV     #165461$-164600$+1000,R5 ; Адрес сообщения "ПЕРЕРЫВАНИЕ #XX"
        CALL    165402$          ; Вывести сообщение на экран

```

```

        HALT                                ; Переход в режим пультавого отладчика
; Подпрограмма вывода на экран номера сетевой станции
165304$:MOVB    @#176561,R0                ; Выделение номера станции, перевод
        MOVB    R0,R1                    ; из BIN в ASCII и запоминание в строке
        BICB    #360,R0                  ; "ТЕСТ СТАНЦИИ #XX"
        BICB    #237,R1
        ASRB    R1
        BISB    R1,R0
        MOV     R0,R1
        BIC     #177770,R1
        BIS     #60,R1
        MOVB    R1,165434$+20
        MOV     R0,R1
        ASR     R1
        ASR     R1
        ASR     R1
        BIC     #177770,R1
        BIS     #60,R1
        MOVB    R1,165434$+17
        MOV     #165434$-164600$+1000,R5 ; Адрес строки "ТЕСТ СТАНЦИИ #XX"
        CALL    165402$                  ; Вывести строку на экран
        RETURN

; Подпрограмма вывода строки на экран
165402$:TSTB    @#177564                  ; Готовность к принятию символа
        BPL     165402$                  ; Не готов
        MOVB    (R5)+,@#177566          ; Вывести очередной символ на экран
        BNE     165402$                  ; Еще не конец строки
        RETURN

165420$: .ASCIZ    <16>/sboj #0/<17><15><12>
165434$: .ASCIZ    <16>/test stancii #00/<17><15><12>
165461$: .ASCIZ    <16>/prerywanie #00/<17><15><12>
165504$: .ASCIZ    <16>/koniec testa/<17><15><12>
        .EVEN

165524$: .WORD     0,0,0,0,0,0
; Загрузчик из стыка С2
165540$:TSTB    @#176574                  ; Проверка готовности передатчика
        BPL     165540$                  ; Не готов
        MOV     #100,@#176576          ; Послать код 100(8) (готовность к обмену)
        CLR     R1                      ; R1 = 0 (адрес считывания BOOT)
        MOV     #1000,R2                ; R2 = 1000 (размер BOOT)
165562$:TSTB    @#176570                  ; Приемник готов ?
        BPL     165562$                  ; Нет
        MOVB    @#176572,(R1)+          ; Переслать принятый байт в память
        SOB     R2,165562$
        CMP     @#0,#240                ; Первая команда BOOT равна NOP ?
        BEQ     165610$                  ; Да
        HALT                                ; Переход в режим пультавого отладчика
165610$:CLR     PC                      ; Переход к исполнению BOOT
; Обработка команд FIS (FADD, FSUB, FMUL, FDIV)
165612$:MOV     R2,@167502$              ; Сохранить R2
        MOV     167502$,R2              ; R2 = адрес блока сохранения
        ADD     #2,R2
        MOV     R0,(R2)+                ; Сохранить регистры,СК и ССП
        $MFPC$
        MOV     R0,14(R2)
        $MFPS$
        MTPS    R0
        MOV     R0,16(R2)
        MOV     R1,(R2)+
        MOV     @167502$,(R2)+
        MOV     R3,(R2)+
        MOV     R4,(R2)+
        MOV     R5,(R2)+
        MOV     SP,(R2)+

```

```

MOV      (R2)+, R5      ; R5 = Значение СК за командой FIS
SUB      #2, R5         ; Получение адреса команды FIS
ADD      #2, R2
$MFPMS$      ; Прочесть код команды FIS
MOV      R0, R1         ; R1 = код команды FIS
BIC      #177770, R0    ; Выделение поля регистра
ASL      R0             ; Доступ к сохр. значению регистра
ADD      167502$, R0
ADD      #2, R0
MOV      R0, (R2)+
MOV      @R0, R5        ; R5 = значение регистра (указ. на блок арг.)
BIT      #1, R5         ; Адрес блока нечетный ?
BNE      166002$        ; Да
CLR      @167502$
CLR      @R2
$MFPMS$      ; Прочесть аргументы
MOV      R0, R2         ; R2 = старшая часть аргумента В
$MFPMS$
MOV      R0, R3         ; R3 = младшая часть аргумента В
$MFPMS$
MOV      R0, R4         ; R4 = старшая часть аргумента А
$MFPMS$
MOV      R0, R5         ; R5 = младшая часть аргумента А
BIC      #177747, R1    ; Выделение поля кода операции
ASR      R1             ; Переход на нужную п/п
ASR      R1
ADD      PC, R1
165772$: ADD      #166112$-165772$, R1
ADD      @R1, R1
JMP      @R1
166002$: MOV      #4, R5      ; R5 = адрес вектора по ош. обр. к каналу
166006$: $MFPMS$      ; Прочесть загр. значение СК
MTPS     #200
$MTPCS$      ; Установить значение РКСК
$MFPMS$      ; Прочесть загр. знач. ССП
$MTPS$      ; Установить значение РКСП
MOV      167502$, R0
ADD      #4, R0
MOV      (R0)+, R1      ; Восстановить регистры
MOV      (R0)+, R2
MOV      (R0)+, R3
MOV      (R0)+, R4
MOV      (R0)+, R5
MOV      (R0)+, SP
MOV      SP, R5         ; R5 = Указатель вершины стека
MOV      2(R0), R0      ; R0 = сохраненное значение ССП
$MTPMS$      ; Занести в стек
MOV      167502$, R0
MOV      20(R0), R0     ; R0 = сохраненное значение СК
$MTPMS$      ; Занести в стек
MOV      167502$, R0
MOV      R5, SP         ; Подкорректировать указатель стека
MOV      14(R0), R5     ; Восстановить R5
MOV      2(R0), R0      ; Восстановить R0
JMP      167500$
; Смещения подпрограмм исполнения действий с ПЗ (FADD, FSUB, FMUL, FDIV)
166112$: .WORD      166624$-166112$, 166620$-166112$
        .WORD      166122$-166112$, 166416$-166112$
; Подпрограмма эмуляции команды FMUL
166122$: BIT      #77600, R2      ; Порядок В = 0 ?
BEQ      166406$      ; Да
BIT      #77600, R4      ; Порядок А = 0 ?
BEQ      166406$      ; Да
MOV      167502$, R1      ; R1 = адрес буфера сохранения

```

```

MOV      R2,R0          ; R0 = старшая часть аргумента В
BPL      166152$        ; В >= 0
INC      26(R1)         ; Аргумент В отрицательный
166152$:TST R4
BPL      166162$        ; А >= 0
INC      26(R1)         ; Аргумент А отрицательный
166162$:MOV R4,R1        ; R1 = старшая часть аргумента А
BIC      #100177,R0     ; Выделение порядка В в R0
BIC      #100177,R1     ; Выделение порядка А в R0
BIC      #177600,R2     ; Выделение старшей части мантиисы В
BIC      #177600,R4     ; Выделение старшей части мантиисы А
BIS      #200,R2        ; Установка скрытого разряда мантиисы В
BIS      #200,R4        ; Установка скрытого разряда мантиисы А
ADD      R1,R0          ; R0(порядок В) = порядок_В + порядок_А
SUB      #40200,R0      ; Вычитание 129
MOV      167502$,R1     ; R1 = адрес блока сохранения
MOV      R0,@R1         ; Сохранить сложенный порядок
MOV      R2,30(R1)      ; Сохранить в буфере мантиису В
MOV      R3,32(R1)

```

ПРИНЦИП УМНОЖЕНИЯ ДВУХ 32-РАЗРЯДНЫХ ЧИСЕЛ

Два 32-разрядных числа А и В можно представить как :

A = ab, где a - старшая его часть, а b - младшая

B = cd, где c - старшая его часть, а d - младшая

Далее реализуется хорошо известный принцип столбика:

```

      c   d
      a   b
-----
      bd
      bc
      ad
      ac

```

```

MOV      R2,R0          ; R0 = старшая часть мантиисы В(c)
ASL      R2              ; Умножение ст.ч. В на 2 и дел. мл.ч. А на 2
ROR      R5              ; для избавления от знакового разр. в А
BCC      166260$        ; Отсутствует остаток от деления на 2
MUL      R5,R2           ; (R2,R3) = bc
ADD      R0,R3           ; Прибавление остатка
ADC      R2
BR       166262$
166260$:MUL R5,R2        ; (R2,R3) = bc
166262$:MOV 167502$,R0   ; R0 = адрес буфера сохранения
MOV      32(R0),R0      ; R0 = младшая часть мантиисы В(d)
ASL      R4              ; Избавление от знаковых разрядов в d
ROR      R0
BCC      166312$        ; Отсутствует остаток от деления на 2
MUL      R4,R0           ; (R0,R1) = ad
ASR      R4              ; Приведение а к нормальному виду
ADD      R4,R3           ; Прибавление остатка к bc
ADC      R2
BR       166316$
166312$:MUL R4,R0        ; (R0,R1) = ad
ASR      R4              ; Приведение а к нормальному виду
166316$:ADD R1,R3        ; (R2,R3) = bc + ad
ADC      R2
ADD      R0,R2
MOV      167502$,R0     ; R0 = адрес буфера сохранения
MOV      30(R0),R0      ; R0 = старшая часть мантиисы В(c)
MUL      R4,R0          ; (R1) = ac

```

```

ADD      R1,R2          ; (R2,R3) = (bc + ad) + ac
MOV      167502$,R0     ; R0 = адрес буфера сохранения
MOV      32(R0),R0      ; R0 = младшая часть мантиссы B(d)
CLC
ROR      R0              ; d = d/2 (b уже поделено (R5))
MUL      R5,R0           ; (R0,R1) = b/2 * d/2 = bd /4
ASHC     #2,R0           ; (R0,R1) = bd/4 * 4 = bd
ADD      R0,R3           ; (R2,R3) = ((bc + ad) + ac) + bd
ADC      R2
CLR      R1              ; Суть этой команды состоит в сбросе бита C
ROR      R2              ; Первый этап нормализации мантиссы
ROR      R3              ; (избавление от знакового разряда)
ASHC     #72,R2          ; Нормализация мантиссы
ROR      R1              ; Сброс бита C
JMP      167140$

```

; Получение нулевого результата

```

166406$:CLR R2           ; (R2,R3) = 0.0
CLR      R3
JMP      167306$

```

; Подпрограмма эмуляции команды FDIV

```

166416$:BIT #77600,R2    ; B = 0 (деление на нуль) ?
BNE      166432$        ; Нет
CLR      R5              ; Слово ошибок = 0
166426$:JMP 167330$      ; Деление на нуль
166432$:BIT #77600,R4    ; A = 0 (нуль / ... = нуль) ?
BEQ      166406$        ; Да
MOV      167502$,R1     ; R1 = адрес блока сохранения
MOV      R2,R0           ; R0 = старшая часть B
BPL      166454$        ; B >= 0
INC      26(R1)          ; Аргумент B отрицательный
166454$:TST R4
BPL      166464$        ; A >= 0
INC      26(R1)          ; Аргумент A отрицательный
166464$:MOV R4,R1        ; R1 = старшая часть A
BIC      #100177,R0     ; Выделение порядка B в R0
BIC      #100177,R1     ; Выделение порядка A в R0
BIC      #177600,R2     ; Выделение старшей части мантиссы B
BIC      #177600,R4     ; Выделение старшей части мантиссы A
BIS      #200,R2        ; Установка скрытого разряда мантиссы B
BIS      #200,R4        ; Установка скрытого разряда мантиссы A
SUB      R0,R1           ; R1(порядок A) = порядок_A - порядок_B
ADD      #37600,R1      ; Прибавить 127
MOV      R1,@167502$    ; Сохранить порядок

```

```

; /-----\
; !                ПРИНЦИП ДЕЛЕНИЯ ДВУХ 32-РАЗРЯДНЫХ ЧИСЕЛ                !
; !                                                                 !
; !      Деление осуществляется методом "угла" с порязрядным сдвигом !
; !      влево аргумента A и результата. Алгоритм вычисления представлен !
; !      ниже :                                                     !
; !      результат := 40                                           !
; !      нц повторять                                             !
; !      | если A < B                                             !
; !      | | то                                                  !
; !      | | | результат := результат * 2                        !
; !      | | | A := A * 2                                         !
; !      | | иначе                                              !
; !      | | | A := A - B                                         !
; !      | | | результат := результат * 2 + 1                    !
; !      | | | A := A * 2                                         !
; !      | все                                                  !
; !      кц пока результат[31] = 0                                !
; !      результат[31] := 0                                       !
; \-----/
MOV      #40,R1          ; Бит признака конца операции

```

```

        CLR      R0
166536$:CMP     R4, #200          ; Мантисса A < 0.5
        BLT      166566$        ; Да
        SUB      R3, R5          ; (R4, R5) = (R4, R5) - (R2, R3)
        SBC      R4
        SUB      R2, R4
        BMI      166560$        ; Мантисса A < B
        SEC      ; Установить бит C (+ 1 для результата)
        BR       166570$
166560$:ADD     R3, R5          ; (R4, R5) = (R4, R5) + (R2, R3) (восстановление
        ADC      R4             ; мантиссы A)
        ADD      R2, R4
166566$:CLC     ; Сбросить бит C (+ 0 для результата)
166570$:ROL     R1             ; Результат := результат * 2 + бит_C
        ROL      R0
        ASL      R5             ; A := A * 2
        ROL      R4
        TST      R0             ; Результат[31] = 0 ?
        BPL      166536$        ; Нет
        BIC      #100000, R0    ; Результат[31] := 0
        MOV      R0, R2         ; (R2, R3) = (R0, R1)
        MOV      R1, R3
        JMP      167140$
; Подпрограмма эмуляции команды FSUB
166620$:ADD     #100000, R2     ; B := (-B)
; Подпрограмма эмуляции команды FADD
166624$:BIT     #77600, R2     ; Аргумент B = 0 ?
        BNE      166660$        ; Нет
        BIT      #77600, R4     ; Аргумент A = 0 ?
        BNE      166650$
        CLR      R2             ; Результат := 0
        CLR      R3
        JMP      167306$
166650$:MOV     R4, R2          ; Результат := A (B = 0)
        MOV     R5, R3
166654$:JMP     167306$
166660$:BIT     #77600, R4     ; Аргумент A = 0 ?
        BEQ      166654$        ; Да (результат := B (A = 0))
        MOV     R2, R0          ; R0 = старшая часть аргумента B
        BIC      #177600, R2    ; Выделение старшей части мантиссы B
        BIS      #200, R2       ; Установка скрытого разряда
        TST      R0             ; Аргумент B < 0 ?
        BPL      166712$        ; Нет
        NEG      R3             ; Изменить знак мантиссы B
        ADC      R2
        NEG      R2
166712$:MOV     R4, R1          ; R1 = старшая часть аргумента A
        BIC      #177600, R4    ; Выделение старшей части мантиссы A
        BIS      #200, R4       ; Установка скрытого разряда
        TST      R1             ; Аргумент A < 0 ?
        BPL      166736$        ; Нет
        NEG      R5             ; Изменить знак мантиссы A
        ADC      R4
        NEG      R4
166736$:BIC     #100177, R0     ; Выделение порядка аргумента B
        BIC     #100177, R1     ; Выделение порядка аргумента A
        MOV     R1, @167502$    ; Запомнить порядок аргумента A
        SUB     R1, R0          ; R0 = порядок_B - порядок_A
        BNE     166762$        ; Порядки не равны
        CLR     R1             ; Последние разряды равны нулю
        BR      167074$
166762$:BMI     167034$        ; Порядок A > порядок B
        ADD     R0, @167502$    ; Сохранить порядок аргумента B (A+B-A=B)
        CLR     R1             ; Последние разряды равны нулю

```

	CMP	R0, #6000	; Разность порядков больше 24
	BLE	167006\$; Нет
	CLR	R4	; Мантисса А равна нулю
	CLR	R5	
	BR	167074\$	
167006\$:	ASL	R0	; Перенос порядка в младшие разряды
	SWAB	R0	
	NEG	R0	; Изменение знака для осущ. сдвига вправо
	INCB	R0	; +1 (далее для отлавл. двух выдвин. разр.)
	BEQ	167024\$	
	ASHC	R0, R4	; Сдвинуть мантиссу А вправо
	ROR	R1	; Запомнить последний разряд
167024\$:	ASHC	#77, R4	; Сдвинуть мантиссу А на 1 разряд вправо
	ROR	R1	; Запомнить предпоследний разряд
	BR	167074\$	
167034\$:	CLR	R1	; Последние разряды равны нулю
	CMP	R0, #172000	; Разность порядков больше -24
	BGE	167052\$; Нет
	CLR	R2	; Мантисса В равна нулю
	CLR	R3	
	BR	167074\$	
167052\$:	ASL	R0	; Перенос порядка в младшие разряды
	SWAB	R0	
	INCB	R0	; +1 (для отлавливания 2 выдвинутых разрядов)
	BEQ	167066\$	
	ASHC	R0, R2	; Сдвинуть мантиссу В вправо
	ROR	R1	; Запомнить последний разряд
167066\$:	ASHC	#77, R2	; Сдвинуть мантиссу В на 1 разряд вправо
	ROR	R1	; Запомнить предпоследний разряд
167074\$:	ADD	R5, R3	; (R2, R3) = (R2, R3) + (R4, R5)
	ADC	R2	
	ADD	R4, R2	
	BPL	167130\$; Результат > 0
	NEG	R1	; Изменить знак мантиссы
	ADC	R3	
	ADC	R2	
	NEG	R3	
	ADC	R2	
	NEG	R2	
	MOV	167502\$, R0	; R0 = адрес буфера сохранения
	INC	26(R0)	; Результат получился отрицательный
167130\$:	MOV	R3, R4	; Проверка на нул. мантиссу (отсюда рез. 0)
	BIS	R1, R4	
	BIS	R2, R4	
	BEQ	167306\$; Мантисса нулевая
167140\$:	CMP	R2, #400	; Мантисса >= 1.0 ?
	BLO	167164\$; Нет
	ROR	R2	; Сдвиг мантиссы вправо на 1 разряд
	ROR	R3	
	ROR	R1	
	ADD	#200, @167502\$; Порядок := порядок + 1
	BR	167140\$	
167164\$:	CMP	R2, #200	; Мантисса < 0.5 ?
	BHIS	167210\$; Нет
	ASL	R1	; Сдвиг мантиссы влево на 1 разряд
	ROL	R3	
	ROL	R2	
	SUB	#200, @167502\$; Порядок := порядок - 1
	BR	167164\$	
167210\$:	ROL	R1	; Бит С := предпоследний разряд
	ADC	R3	; Прибавить предпоследний разряд (округление)
	ADC	R2	
167216\$:	CMP	R2, #400	; Мантисса >= 1.0
	BLO	167242\$; Нет

```

ROR    R2                ; Сдвиг мантиссы вправо на 1 разряд
ROR    R3
ROR    R1
ADD    #200,@167502$    ; Порядок := порядок + 1
BR     167216$
167242$:CLR    R5        ; Регистр признаков = 0
MOV    @167502$,R4      ; R4 = порядок
BEQ    167334$          ; Порядок равен нулю (исчезновение порядка)
BPL    167262$          ; Положительный порядок
ROL    R4
BPL    167340$          ; Переполнение порядка
BR     167334$          ; Исчезновение порядка
167262$:MOV    167502$,R0 ; R0 = адрес буфера сохранения
ROR    26(R0)           ; Результат отрицательный ?
BCC    167300$          ; Нет
BIS    #100000,R2       ; Установка знакового разряда
167300$:BIC    #200,R2   ; Сброс скрытого бита
BIS    R4,R2            ; Установить поле порядка
167306$:CLR    R5        ; Регистр признаков = 0
TST    R2              ; Установить ССП
BNE    167320$         ; Результат не равен нулю
BIS    #4,R5           ; Установить бит Z
167320$:BPL    167350$   ; Результат больше нуля
BIS    #10,R5          ; Установить бит N
BR     167350$
167330$:BIS    #1,R5     ; R5 = 13 (деление на нуль)
167334$:BIS    #10,R5    ; R5 = 12 (исчезновение порядка)
167340$:BIS    #2,R5     ; R5 = 2 (переполнение порядка)
BIS    #100000,R5       ; Признак регистра ошибок
167350$:MOV    167502$,R0 ; R0 = адрес буфера сохранения
BIC    #157,22(R0)      ; Сброс битов N,Z,V,C в сохр. ССП
BISB   R5,22(R0)        ; Установка битов признаков из R5
TST    R5              ; Произошла ошибка ?
BPL    167402$          ; Нет
MOV    #244,R5         ; R5 = адрес вект. прер. при ош. с числами ПЗ
JMP    166006$
167402$:ADD    #4,@24(R0) ; Прибавить 4 к содерж. рег. из кода опер.
MOV    @24(R0),R5       ; R5 = адрес памяти для записи результата
ADD    #4,R5
MOV    20(R0),R4        ; R4 = сохр. значение СК
MOV    22(R0),R0        ; R0 = сохр. значение ССП
MTPS    #200           ; Запретить прерывания
$MTPS$          ; Установить РКСП
MOV    R4,R0
$MTPC$          ; Установить РКСК
MOV    R3,R0
$MTPM$          ; Запись младшей части результата
MOV    R2,R0
$MTPM$          ; Запись старшей части результата
MOV    167502$,R0       ; R0 = адрес буфера сохранения
CMP    (R0)+,(R0)+
MOV    (R0)+,R1         ; Восстановить регистры
MOV    (R0)+,R2
MOV    (R0)+,R3
MOV    (R0)+,R4
MOV    (R0)+,R5
MOV    @R0,SP
MOV    -14(R0),R0
167500$: $RUN$          ; Продолжить программу
167502$: .WORD    167504$ ; Адрес буфера сохранения
; Буфер сохранения (текущий порядок, R0, R1, R2, R3, R4, R5, SP, PC, PS, указатель на
; поле регистра из инструкции FP, знак результата, 2 слова для промежуточного
; хранения мантиссы)
167504$: .BLKW    16

```


; Подпрограмма исполнения пункта меню ЗАГРУЗКА [тестирование]

```
167540$:CLR    @#177656      ; Номер прохода
             CLR    @#177654      ; Число ошибок
             MOV    #177700,R1    ; R1 = адрес буфера регистрации ошибок
             MOV    #10,R2       ; Размер буфера
167560$:CLR    (R1)+          ; Цикл очистки буфера
             SOB    R2,167560$
             JSR    R4,163006$    ; Вывод заголовка на экран
             .WORD  170116$
167572$:CLV
             INC    @#177656      ; Увеличить номер прохода
             BVC    167610$       ; Нет переполнения
             MOV    #-1,@#177656 ; Номер прохода равен 65535
167610$:JSR    R4,163006$    ; Вывод надписи "ПРОХОД:"
             .WORD  170165$
             MOV    @#177656,R3   ; R3 = номер прохода
             CALL   170032$       ; Перевод номера прохода в строку символов
             JSR    R4,163006$    ; Вывод номера прохода на экран
             .WORD  177660
             MOV    #33,R0        ; Посылка ESC <276> (тест в области ПП)
             CALL   172632$
             MOV    #276,R0
             CALL   172632$
             CALL   170632$       ; Тестирование процессора (ЦП)
             TST    R0            ; Есть ошибки ?
             BEQ    167704$       ; Нет
             MOV    #16,R0        ; R0 = смещение в буфере регистрации ошибок
             CMP    #-1,177700(R0) ; Число ошибок достигло максимума ?
             BEQ    167704$       ; Да
             INC    177700(R0)    ; Увеличить счетчик числа ошибок
167704$:CLR    R4              ; R4 = 0 (начальный адрес памяти ЦП)
             MOV    #160000,R5    ; R5 = размер ОЗУ ЦП
             CALL   170224$       ; Тестирование памяти
             TST    R0            ; Есть ошибки ?
             BEQ    167742$       ; Нет
             MOV    #14,R0        ; R0 = смещение в буфере регистрации ошибок
             CMP    #-1,177700(R0) ; Число ошибок достигло максимума ?
             BEQ    167742$       ; Да
             INC    177700(R0)    ; Увеличить счетчик числа ошибок
167742$:CALL   172614$          ; Ожид. симв. с кодом 177(8)(ПП кончил тест)
             CLR    @#177654      ; Общее число ошибок = 0
             MOV    #177700,R1    ; R1 = адрес буфера регистрации ошибок
             MOV    #10,R2       ; R2 = размер буфера
             CLV
167764$:ADD    (R1)+,@#177654    ; Цикл получения общей суммы
             BVC    170002$       ; Нет переполнения
             MOV    #-1,@#177654 ; Общее число ошибок = 65535
             BR     170004$
170002$:SOB    R2,167764$
170004$:JSR    R4,163006$    ; Вывод надписи "ОШИБОК:"
             .WORD  170204$
             MOV    @#177654,R3   ; R3 = число ошибок
             CALL   170032$       ; Перевод числа ошибок в строку символов
             JSR    R4,163006$    ; Вывод числа ошибок на экран
             .WORD  177660
             BR     167572$
```

; Подпрограмма перевода 16-разрядного числа в его ASCII-представление

```
170032$:MOV    #177660,R0      ; R0 = адрес буфера приема символов
             MOV    #6,R2       ; R2 = размер буфера
170042$:MOVB   #40,(R0)+       ; Инициализация буфера (заполнение пробелами)
             SOB    R2,170042$
             CLRB   @R0         ; Отметить конец буфера
170052$:CLR    -(SP)           ; Остаток от деления равен 0 (инициализация)
             MOV    #16.,R1     ; R1 = число разрядов
```

```

170060$:ASL      R3              ; Цикл дел. на 10, частное в R3, остаток в @SP
        ROL      @SP
        CMP      @SP,#10.
        BMI      170100$
        SUB      #10.,@SP
        INC      R3
170100$:SOB      R1,170060$
        ADD      #'0,@SP        ; Преобразование из BIN в ASCII
        MOVB     (SP)+,-(R0)     ; Занести символ в буфер
        TST      R3              ; Деление окончено ?
        BNE      170052$        ; Нет
        RETURN
170116$: .BYTE    33,'H',33,'J',33,'Y','%',':',33,244
        .ASCII    <16>/t e s t i r o w a n i e/
        .BYTE     33,277,244,17,0
170165$: .ASCIZ   <33>/Y(5/<16>/prohod: /<17>
170204$: .ASCIZ   <16>/ o{ibok: /<17>
        .EVEN
; Подпрограмма тестирования оперативного запоминающего устройства
170224$:MOV      R4,R1          ; R1 = начальный адрес ОЗУ
        CLR      R3
170230$:MOV      R1,R0          ; R0 = текущий адрес
        TST      R3            ; Прямой адрес(0), инверсный адрес(<>0)
        BEQ      170240$
        COM      R0
170240$:MOV      R0,(R1)+       ; Записать в ячейку ее адрес(прям. или инв.)
        CMP      R1,R5         ; Достигли конца ?
        BLO      170230$       ; Нет
170246$:CMP      R0,-(R1)       ; Содержимое ячейки равно ее адресу
        BEQ      170260$       ; Да
        MOV      #1,R0         ; Ошибка
        RETURN
170260$:MOV      R1,R0          ; R0 = текущий адрес
        SUB      #2,R0         ; На одну ячейку назад
        TST      R3            ; Прямой или инверсный адрес ?
        BEQ      170274$       ; Прямой
        COM      R0
170274$:CMP      R1,R4          ; Достигли начала ?
        BHI      170246$       ; Нет
        COM      R3            ; Перейти на инверсный адрес
        BNE      170230$       ; Произвести тест с инверсным адресом
        MOV      #377,-12(SP)  ; Будет заполняться младший байт
        MOV      R4,R3         ; R3 = начальный адрес
170314$:CLR      -6(SP)
        MOV      R3,-10(SP)    ; Начальный адрес
        MOV      R3,R2
        BIS      #17776,R2
        TST      (R2)+         ; Плюс 8 КБайт
        CMP      R5,R2         ; Достигли конца ?
        BHIS     170342$       ; Нет
        MOV      R5,R2         ; R2 = конечный адрес
170342$:MOV      R3,-14(SP)    ; Текущий адрес
        MOV      -10(SP),R1    ; R1 = начальный адрес
170352$:MOV      -12(SP),R0    ; R0 = образец заполнения
        CMP      R1,R3
        BNE      170370$
        ADD      #200,R3       ; Плюс 128 байт (128 байт 1..10..0,0..01..1)
        SWAB     R0            ; Изменить байты образца
170370$:BITB     #1,-6(SP)     ; Делать запись
        BNE      170402$       ; Нет
        MOV      R0,@R1        ; Записать в ячейку
170402$:CMP      R0,@R1        ; Совпадает ?
        BEQ      170414$       ; Да
        MOV      #3,R0         ; Ошибка

```

```

RETURN
170414$:ADD    #2,R1          ; Перейти к следующей ячейке
CMP          R1,R2          ; Достигнут конец 8-КБайтной страницы ?
BLO          170352$        ; Нет
INCB         -6(SP)         ; Переключить режим записи/проверки
MOV          -14(SP),R3      ; R3 = текущий адрес
BITB         #1, -6(SP)     ; Какой режим ?
BNE          170342$        ; Проверка
TST          (R3)+          ; Текущий адрес + 2
CMP          R3,R2          ; Конец страницы ?
BHIS         170460$        ; Да
TSTB         -6(SP)         ; Сделано 128 проходов ?
BPL          170342$        ; Нет
170460$:MOV    -10(SP),R3    ; R3 = текущий адрес
SWAB         -12(SP)        ; Обменять байты образца
BEQ          170342$        ; Младший байт = 0 (для второго прохода)
MOV          R2,R3          ; Текущий адрес = начало след. 8-КБайтной стр.
CMP          R2,R5          ; Достигнут конец ?
BLO          170314$        ; Нет
MOV          #377, -12(SP)  ; Образец заполнения
170506$:MOV    R4,R1         ; R1 = начальный адрес
MOV          -12(SP),R0     ; R0 = образец заполнения
170514$:MOV    R0,(R1)+      ; Записать в ячейку
CMP          R1,R5          ; Достигнут конец ?
BLO          170514$        ; Нет
MOV          R4,R1         ; R1 = начальный адрес
170524$:CMP    R0,@R1        ; Сравнить содержимое ячейки с образцом
BEQ          170536$        ; Равно
MOV          #2,R0          ; Ошибка
RETURN
170536$:ADD    #2,R1          ; Перейти к следующей ячейке
CMP          R1,R5          ; Достигнут конец ?
BLO          170524$        ; Нет
CALL         170614$        ; Задержка
MOV          -12(SP),R0     ; R0 = образец заполнения
MOV          R4,R1         ; R1 = начальный адрес
170560$:CMP    R0,@R1        ; Сравнить содержимое ячейки с образцом
BEQ          170572$        ; Равно
MOV          #3,R0          ; Ошибка
RETURN
170572$:ADD    #2,R1          ; Перейти к следующей ячейке
CMP          R1,R5          ; Достигнут конец ?
BLO          170560$        ; Нет
SWAB         -12(SP)        ; Обменять байты образца
BEQ          170506$        ; На второй проход
CLR          R0             ; Нет ошибок
RETURN
; Подпрограмма реализации задержки
170614$:MOV    #1000.,R2
170620$:MOV    #1000.,R3
170624$:SOB    R3,170624$
SOB          R2,170620$
RETURN
; Подпрограмма тестирования процессора
170632$:CCC    ; Сбросить все биты признаков
BCS          170654$        ; C=1
BVS          170654$        ; V=1
BEQ          170654$        ; Z=1
BMI          170654$        ; N=1
BLT          170654$        ; N<>V
BLE          170654$        ; (N<>V) or Z=1
BLOS         170654$        ; Z=1 or C=1
BHI          170660$        ; Z=0 and C=0
170654$:JMP    172604$      ; Ошибка

```

170660\$:	SEV		; Установить V и N
	SEN		
	BVC	170674\$; V=0
	BLT	170674\$; N<>V
	BLE	170674\$; (N<>V) or Z=1
	BGE	170700\$; N=V
170674\$:	JMP	172604\$; Ошибка
170700\$:	SEZ		; Установить Z и C
	SEC		
	BNE	170714\$; Z=0
	BGT	170714\$; Z=0 and (N=V)
	BHI	170714\$; Z=0 and C=0
	BLE	170720\$; (N<>V) or Z=1
170714\$:	JMP	172604\$; Ошибка
170720\$:	MOV	SP, R3	
	TST	-(R3)	; R3 = R3 - 2
	CLR	@R3	
	MOV	R3, R0	
	TST	-(R3)	; R3 = R3 - 2
	MOV	R0, @R3	; (R3) = адрес ячейки с нулевым содерж.
	MOV	R3, R4	
	CCC		; Очистить все биты признаков
	TST	@(R3)+	; В ячейке нуль ?
	BEQ	170750\$; Да
	JMP	172604\$; Ошибка
170750\$:	SEC		; Установка бита C
	ROR	@-(R3)	; Уст. ст. разр. в 1, выдв. млад. =0 (10...0))
	BCS	170762\$; C=1
	BVC	170762\$; V=0
	BMI	170766\$; N=1
170762\$:	JMP	172604\$; Ошибка
170766\$:	CCC		; Очистить все биты признаков
	ASR	@(R4)+	; Оставить старший бит (110...0)
	BVC	170776\$; V=0
	BMI	171002\$; N=1
170776\$:	JMP	172604\$; Ошибка
171002\$:	CLN		; Очистить бит N
	ASL	@(R3)+	; Сдвиг влево (10...0)
	BCC	171014\$; C=0
	BVS	171014\$; V=1
	BMI	171020\$; N=1
171014\$:	JMP	172604\$; Ошибка
171020\$:	SCC		; Установить все биты признаков
	DEC	@-(R4)	; Минус 1 (01...1)
	BHIS	171034\$; C=0
	BVC	171034\$; V=0
	BEQ	171034\$; Z=1
	BPL	171040\$; N=0
171034\$:	JMP	172604\$; Ошибка
171040\$:	NEG	@-(R3)	; Смена знака (10...01)
	BCC	171050\$; C=0
	BVS	171050\$; V=1
	BMI	171054\$; N=1
171050\$:	JMP	172604\$; Ошибка
171054\$:	SEV		; Установка бита V
	COM	@(R4)+	; Инверсия (01...10)
	BCC	171064\$; C=0
	BVC	171070\$; V=0
171064\$:	JMP	172604\$; Ошибка
171070\$:	INC	@(R3)+	; Плюс 1 (01...1)
	BHIS	171076\$; C=0
	BPL	171102\$; N=0
171076\$:	JMP	172604\$; Ошибка
171102\$:	ADC	@-(R4)	; Прибавление переноса (10...0)

BCS	171112\$; C=1
BVC	171112\$; V=0
BMI	171116\$; N=1
171112\$: JMP	172604\$; Ошибка
171116\$: CCC		; Очистить все биты признаков
ROL	@(R4)+	; Сдвиг влево (0...0)
BCC	171130\$; C=0
BVC	171130\$; V=0
BEQ	171134\$; Z=1
171130\$: JMP	172604\$; Ошибка
171134\$: INC	@-(R3)	; Плюс 1 (0...01)
SBC	@-(R4)	; Вычитание переноса (0...0)
BCS	171144\$; C=1
BEQ	171150\$; Z=1
171144\$: JMP	172604\$; Ошибка
171150\$: MOV	SP, R2	
TST	-(R2)	; R2 = R2 - 2
MOV	R2, R0	
CLR	@R0	; Очистка (0...0)
TST	-(R2)	; R2 = R2 - 4
TST	-(R2)	
MOV	R0, (R2)+	; SP-2
INC	R0	
MOV	R0, (R2)+	; SP-1
MOV	R2, R0	; SP-2
MOV	R2, R1	; SP-2
COMB	@-(R2)	; Инверсия (High 1...1)
BCC	171204\$; C=0
BMI	171210\$; N=1
171204\$: JMP	172604\$; Ошибка
171210\$: TSTB	@-(R2)	; Проверка (Low 0...0)
BEQ	171220\$; Z=1
JMP	172604\$; Ошибка
171220\$: SEV		; Установка бита V
ASRB	@-(R1)	; Сдвиг вправо (High 1...1)
BCC	171232\$; C=0
BVS	171232\$; V=1
BMI	171236\$; N=1
171232\$: JMP	172604\$; Ошибка
171236\$: INCB	@(R2)+	; Плюс 1 (Low 0...01)
BCC	171244\$; C=0
BPL	171250\$; N=0
171244\$: JMP	172604\$; Ошибка
171250\$: CLC		; Сбросить бит C
RORB	@-(R1)	; Сдвиг вправо (Low 0...0)
BCC	171264\$; C=0
BVC	171264\$; V=0
BNE	171264\$; Z=0
BPL	171270\$; N=0
171264\$: JMP	172604\$; Ошибка
171270\$: ASLB	@(R2)+	; Сдвиг влево (High 1...10)
BCC	171300\$; C=0
BVS	171300\$; V=1
BMI	171304\$; N=1
171300\$: JMP	172604\$; Ошибка
171304\$: ADCB	@-(R2)	; Прибавление переноса (High 1...1)
BCS	171312\$; C=1
BMI	171316\$; N=1
171312\$: JMP	172604\$; Ошибка
171316\$: SCC		; Установка всех битов признаков
ROLB	@(R1)+	; Сдвиг влево (Low 0...01)
BLOS	171330\$; C=1 or Z=1
BVS	171330\$; V=1
BPL	171334\$; N=0

171330\$:	JMP	172604\$; Ошибка
171334\$:	SWAB	@-(R2)	; Обменять байты (High 0...01 Low 1...1)
	BMI	171344\$; N=1
	JMP	172604\$; Ошибка
171344\$:	SEC		; Установка бита C
	SBCB	@(R1)+	; Вычитание переноса (High 0...0)
	BCS	171354\$; C=1
	BEQ	171360\$; Z=1
171354\$:	JMP	172604\$; Ошибка
171360\$:	NEGB	@(R2)+	; Смена знака (Low 0...01)
	DECB	@-(R2)	; Минус 1 (Low 0...0)
	BCC	171370\$; C=0
	BEQ	171374\$; Z=1
171370\$:	JMP	172604\$; Ошибка
171374\$:	MOV	SP, R0	; R0 = SP
	TST	-(R0)	; R0 = R0 - 2
	MOV	R0, R1	; SP-2
	CMP	-(R1), -(R1)	; SP-6
	MOV	R0, @R1	; -6(SP) = ADDR(-2(SP))
	MOV	R1, R2	; SP-6
	MOV	R0, R4	; SP-2
	TST	-(R0)	; SP-4
	MOV	R0, R3	; SP-4
	MOV	R0, -(R2)	; R2 = SP-8 -8(SP) = ADDR(-4(SP))
	CLR	@R3	; -4(SP) = 0
	CLR	@R4	; -2(SP) = 0
	SCC		; Установка битов N, V, C
	CLZ		
	SUB	@(R2)+, @(R1)+	; -2(SP)=-2(SP) - -4(SP)=0
	BLO	171440\$; C=1
	BVS	171440\$; V=1
	BEQ	171444\$; Z=1
171440\$:	JMP	172604\$; Ошибка
171444\$:	BIS	#100000, @-(R2)	; -4(SP) = 100000
	ADD	#1, @-(R1)	; -2(SP) = 1
	SUB	@(R2)+, @(R1)+	; -2(SP) = 1 - 100000 = 100001
	BHIS	171464\$; C=0
	BVC	171464\$; V=0
	BMI	171470\$; N=1
171464\$:	JMP	172604\$; Ошибка
171470\$:	NEG	@R4	; -2(SP) = 077777
	BIT	@-(R2), @-(R1)	; 100000 and 077777 = 0
	BEQ	171502\$; Z=1
	JMP	172604\$; Ошибка
171502\$:	CMP	@(R2)+, @(R1)+	; 100000 - 077777 = 1 (C=1, V=1, N=0, Z=0)
	BVS	171512\$; V=1
	JMP	172604\$; Ошибка
171512\$:	COM	@-(R2)	; Инверсия -4(SP) = 077777
	CCC		; Очистить все биты признаков
	ADD	@(R2)+, @-(R1)	; -2(SP) = 077777 + 077777 = 177776
	BVC	171524\$; C=0
	BMI	171530\$; N=1
171524\$:	JMP	172604\$; Ошибка
171530\$:	SEC		; Установка бита C
	BIC	@-(R2), @(R1)+	; -2(SP) = 177776 and not 77777 = 100000
	BCC	171540\$; C=0
	BMI	171544\$; N=1
171540\$:	JMP	172604\$; Ошибка
171544\$:	COM	@-(R1)	; Инверсия -2(SP) = 77777
	CMP	@(R2)+, @(R1)+	; 077777 - 077777 = 0 (C=0, V=0, N=0, Z=1)
	BEQ	171556\$; Z=1
	JMP	172604\$; Ошибка
171556\$:	MOV	SP, R0	; R0 = SP
	TST	-(R0)	; SP-2

```

MOV      R0,R3          ; SP-2
MOV      R3,R1          ; SP-2
TST      -(R3)          ; SP-4
MOV      R0,-(R3)       ; -6(SP) = ADDR(-1(SP))
INC      @R3
MOV      R0,-(R3)       ; -8(SP) = ADDR(-2(SP))
MOV      R3,R4          ; SP-8
TST      -(R0)          ; SP-4
MOV      R0,-(R4)       ; -10(SP) = ADDR(-3(SP))
INC      @R4
MOV      R0,-(R4)       ; -12(SP) = ADDR(-4(SP))
SEC                      ; Установить бит C
MOV      #177001,@(R4)+ ; -4(SP) = 1      -3(SP) = 376
MOVB     #200,@(R4)+    ; -3(SP) = 200
MOVB     @-(R4),@(R3)+  ; -2(SP) = 200
MOVB     @-(R4),@(R3)+  ; -1(SP) = 1
BCS      171634$        ; C=1
JMP      172604$        ; Ошибка
171634$:CMP #600,@R1    ; 600 - 600 = 0 (C=0,V=0,N=0,Z=1)
BEQ      171646$        ; Z=1
JMP      172604$        ; Ошибка
171646$:CMP -(R3),-(R3) ; SP-8
BISB     @(R4)+,@(R3)+  ; -2(SP) = 200 or 1 = 201
BISB     @(R4)+,@(R3)+  ; -1(SP) = 1 or 200 = 201
CMP      #100601,@R1    ; 100601 - 100601 = 0 (C=0,V=0,N=0,Z=1)
BEQ      171666$        ; Z=1
JMP      172604$        ; Ошибка
171666$:BICB @-(R4),@-(R3) ; -1(SP) = 201 and not 200 = 1
BICB     @-(R4),@-(R3)  ; -2(SP) = 201 and not 1 = 200
BITB     @(R4)+,@(R3)+  ; 1 and 200 = 0
BNE      171702$        ; Z=0
BITB     @-(R4),@(R3)+  ; 1 and 1 = 1
BNE      171706$        ; Z=1
171702$:JMP 172604$      ; Ошибка
171706$:CMPB @(R4)+,@-(R3) ; 1 - 1 = 0 (C=0,V=0,N=0,Z=1)
BNE      171716$        ; Z=0
CMPB     @(R4)+,@-(R3)  ; 1 - 200 = 201 (C=1,V=1,N=1,Z=0)
BEQ      171722$        ; Z=1
171716$:JMP 172604$      ; Ошибка
171722$:MOV PC,R0        ; R0 = 171724
ADD      #171736$-.,R0   ; R0 = 171736
SCC                      ; Установить все биты признаков
JMP      @R0            ; PC = 171736
171734$:BR 171742$
171736$:CLN              ; Сбросить бит N
BR       171734$
171742$:BCC 171752$      ; C=0
BVC      171752$        ; V=0
BNE      171752$        ; Z=0
BPL      171756$        ; N=0
171752$:JMP 172604$      ; Ошибка
171756$:CLR R2
MOV      PC,R4          ; R4 = 171762
MOV      R4,R0          ; R0 = 171762
BR       171772$
171766$:COM R2          ; R2 = 177777
BR       172000$
171772$:CMP (R4)+,(R4)+  ; R4 = 171766
TST      (R4)+          ; R4 = 171770
JMP      -(R4)          ; PC = 171766      R4 = 171766
172000$:INC R2          ; R2 = 0
BNE      172012$        ; Z=0
CMP      (R0)+,(R0)+    ; R0 = 171766
CMP      R0,R4          ; 171766 - 171766 = 0 (C=0,V=0,N=0,Z=1)

```

```

        BEQ      172016$      ; Z=1
172012$: JMP      172604$      ; Ошибка
172016$: MOV      PC, R2      ; R2 = 172020
        ADD      #172044$-., R2 ; R2 = 172044
        SCC                      ; Установить биты N, C, Z
        CLV
        JSR      R1, @R2      ; PC = 172044
        TST      R2
        BEQ      172066$      ; Z=1
        JMP      172604$
172042$: RTS      R1
172044$: BCC      172060$      ; C=0
        BVS      172060$      ; V=1
        BNE      172060$      ; Z=0
        BPL      172060$      ; N=0
        CLR      R2
        BR       172042$
172060$: TST      (SP)+
        JMP      172604$      ; Ошибка
172066$: MOV      SP, R4
        TST      -(R4)        ; SP-2
        CLR      R3
        MOV      #5, R2
        MOV      #2, @R4
        SCC                      ; Установить все признаки
        MUL      @R4, R2      ; (R2, R3) = 5 * 2 = 12 (R2 = 0) (R3 = 12)
        BMI      172116$      ; N=1
        BLOS     172116$      ; Z=1 or C=1
        BVC      172122$      ; V=0
172116$: JMP      172604$      ; Ошибка
172122$: CMP      #12, R3      ; R3=12 ?
        BEQ      172134$      ; Да
        JMP      172604$      ; Ошибка
172134$: TST      R2           ; R2=0 ?
        BEQ      172144$
        JMP      172604$      ; Ошибка
172144$: CLR      R3
        MOV      #125252, R2
        MOV      #2, @R4
        MUL      @R4, R2      ; (R2, R3)=125252*2=37777652524 (R2=-1) (R3=52524)
        BCC      172170$      ; C=0
        BPL      172170$      ; N=0
        BVS      172170$      ; V=1
        BNE      172174$      ; Z=0
172170$: JMP      172604$      ; Ошибка
172174$: CMP      #-1, R2      ; R2=-1 ?
        BEQ      172206$      ; Да
        JMP      172604$      ; Ошибка
172206$: CMP      #52524, R3    ; R3=52524 ?
        BEQ      172220$      ; Да
        JMP      172604$      ; Ошибка
172220$: MOV      #1, @R4
        MOV      #125252, R2
        CCC                      ; Установить биты Z, N
        SEZ
        SEN
        ASH      @R4, R2      ; R2 = 125252 << 1 = 52524
        BMI      172250$      ; N=1
        BEQ      172250$      ; Z=1
        BVC      172250$      ; V=0
        BCS      172254$      ; C=1
172250$: JMP      172604$      ; Ошибка
172254$: CMP      #52524, R2    ; R2 = 52524 ?
        BEQ      172266$      ; Да

```



```

        JMP      172604$      ; Ошибка
172266$:MOV      #-1,@R4
        MOV      #52525,R1
        ASH      @R4,R1      ; R1 = 52525 >> 1 = 25252
        CMP      R1,#25252    ; R1 = 25252 ?
        BEQ      172312$      ; Да
        JMP      172604$      ; Ошибка
172312$:CLR      R2
        MOV      #20,@R4
        MOV      #125252,R3
        CCC                      ; Установка битов Z,C
        SEZ
        SEC
        ASHC     @R4,R2      ; (R2,R3)=125252<<20=25252400000
        BPL      172344$      ; N=0
        BEQ      172344$      ; Z=1
        BVC      172344$      ; V=0
        BCC      172350$      ; C=0
172344$:JMP      172604$      ; Ошибка
172350$:CMP      #125252,R2    ; R2 = 125252 ?
        BEQ      172362$      ; Да
        JMP      172604$      ; Ошибка
172362$:TST      R3           ; R3 = 0 ?
        BEQ      172372$      ; Да
        JMP      172604$      ; Ошибка
172372$:CLR      R3
        MOV      #-20,@R4
        MOV      #125252,R2
        CLN                      ; Установка Z,V,C
        SEZ
        SEV
        SEC
        ASHC     @R4,R2      ; (R2,R3)=25252400000>>20=37777725252
        BPL      172424$      ; N=0
        BLOS     172424$      ; Z=1 or C=1
        BVC      172430$      ; C=0
172424$:JMP      172604$      ; Ошибка
172430$:CMP      #-1,R2       ; R2=-1 ?
        BEQ      172442$      ; Да
        JMP      172604$      ; Ошибка
172442$:CMP      #125252,R3    ; R3=125252 ?
        BEQ      172454$      ; Да
        JMP      172604$      ; Ошибка
172454$:MOV      #52525,@R4
        CLR      R2
        MOV      #52525,R3
        SCC                      ; Установить все биты признаков
        DIV      @R4,R2       ; (R2,R3) = 52525 / 52525 = (1,0)
        BMI      172500$      ; N=1
        BLOS     172500$      ; Z=1 or C=1
        BVC      172504$      ; V=0
172500$:JMP      172604$      ; Ошибка
172504$:CMP      R2,#1         ; R2 = 1 ?
        BEQ      172516$      ; Да
        JMP      172604$      ; Ошибка
172516$:TST      R3           ; R3 = 0 ?
        BEQ      172526$      ; Да
        JMP      172604$      ; Ошибка
172526$:MOV      #157777,R2
        MOV      #100001,R3
        MOV      #100000,@R4
        SCC                      ; Установить все биты признаков
        DIV      @R4,R2       ; (R2,R3)=33777700001/100000=(40000,100001)
        BMI      172554$      ; N=1

```

```

        BLOS    172554$      ; Z=1 or C=1
        BVC     172560$      ; V=0
172554$: JMP    172604$      ; Ошибка
172560$: CMP    R2, #40000    ; R2 = 40000 ?
        BEQ     172572$      ; Да
        JMP     172604$      ; Ошибка
172572$: CMP    R3, #100001    ; R3 = 100001 ?
        BNE     172604$      ; Z=0 (ошибка)
        CLR     R0           ; Ошибок нет
        RETURN
172604$: MOV    #1, R0        ; Есть ошибки
        RETURN
172612$: .WORD  414          ; Конец стека пультового отладчика
; Подпрограмма ввода символа с ожиданием (для ПП)
172614$: EMT     22          ; Ввод символа
        RETURN
        .WORD   0, 0, 0, 0, 0
; Подпрограмма вывода символа на экран (для ПП)
172632$: EMT     42          ; Вывод символа
        RETURN
        .WORD   0, 0, 0, 0, 0, 0, 0, 0, 0
; Указатели на ОЗУ пультового отладчика, адрес выводимой надписи, адрес открытой
; ячейки, разрешение/запрет вывода надписей. Начальное значение PS и SP.
; Объем ОЗУ в словах. Эти данные для ПП.
172660$: .WORD   450
172662$: .WORD   500
172664$: .WORD   472
172666$: .WORD   504
172670$: .WORD    0
172672$: .WORD  100000
172674$: .WORD   40000

; Начальная инициализация и вывод ошибок стартового теста
172676$: MOV    #2000, SP
        ASLB    R0
        ASLB    R0
        MOV     R0, -(SP)
        CALL    173502$      ; Полн. иниц. (уст.вект.прер. и иниц. устр.)
        MOV     (SP)+, R2    ; R2 = слово ошибок
        MOV     @#177060, R0 ; R0 = данные К0 (терминал)(слово ошибок ЦП)
        CMP     R0, #2
        BHI     172742$
        BIS     R0, R2        ; Объединить ошибки ПП и ЦП
        BEQ     172746$      ; Нет ошибок
        CALL    172764$      ; Вывод ошибок на экран
        BR      172746$
172742$: CALL    117136$      ; Поместить символ в буфер терминала
172746$: MOV     #2, @#7042    ; Передача кодов клавиатуры по каналу 0
        MTPS     #0          ; Разрешить прерывания
        JMP     174152$

; Подпрограмма вывода результатов стартового теста
172764$: JSR     R5, 117204$    ; Вывод заголовка
        .WORD    173060$
        MOV     #173036$, R3    ; Адрес массива адресов надписей об ошибках
        BR      173004$
173000$: JSR     R5, 117206$    ; Вывод сообщения об ошибке на экран
173004$: MOV     (R3)+, R1      ; R1 = адрес сообщения об ошибке
        ASRB     R2
        BCS     173000$        ; Есть ошибка
        BNE     173004$        ; Еще есть ошибки
        MOV     #173046$, R3    ; Адрес массива адресов об ошибках ПЗУ
        MOV     (R3)+, R1      ; R1 = адрес сообщения об ошибке
        SWAB     R2
        BNE     173000$        ; Ошибки есть

```

```

        JSR      R5,117204$      ; Вывод ESC <270> (пауза)
        .WORD   173246$
        RETURN
; Массив адресов сообщений об ошибках
173036$: .WORD   173113$,173131$,173153$,173171$
; Массив адресов сообщений об ошибках ПЗУ
173046$: .WORD   173213$,173232$,173235$,173240$,173243$
173060$: .BYTE   15,12,40,40,33,244
        .ASCII   /СТАРТОВЫЙ ТЕСТ/
        .BYTE   33,277,244,15,12,12,0
173113$: .ASCIZ   /- ошибка ЦП/<15><12>
173131$: .ASCIZ   /- ошибка ОЗУ ЦП/<15><12>
173153$: .ASCIZ   /- ошибка ПП/<15><12>
173171$: .ASCIZ   /- ошибка ОЗУ ПП/<15><12>
173213$: .ASCIZ   /- ошибка ПЗУ /
173232$: .ASCIZ   /1 /
173235$: .ASCIZ   /2 /
173240$: .ASCIZ   /3 /
173243$: .ASCIZ   /4 /
173246$: .BYTE   33,270,0
        .EVEN
; Подпрограмма перекачки пультового отладчика в системное ОЗУ ЦП
173252$: MOV      #<160000/2>&77777, @#177010 ; Адрес начала системного ОЗУ ЦП
        MOV      #160000, R1 ; Адрес начала пультового отладчика
        MOV      #5305, R0 ; Размер перек. данных (160000 - 172611)
173270$: MOV      (R1)+, @#177014 ; Цикл перекачки
        INC      @#177010
        SOB      R0, 173270$
        MOV      #173356$, R1 ; Адрес подпрограмм для ЦП
        MOV      #44, R0 ; Размер (173356 - 173465)
173312$: MOV      (R1)+, @#177014 ; Цикл перекачки в 172612 - 172721
        INC      @#177010
        SOB      R0, 173312$
        MOV      @#177716, @#177014 ; (172722) = системный регистр управления
        MOV      #173466$, R1 ; Адр. мас. для уст. знач. ССП в вект. прер.
173336$: MOV      (R1)+, @#177010 ; Цикл установки
        MOV      #600, @#177014
        TST      @R1 ; Конец массива ?
        BNE      173336$ ; Нет
        RETURN
173356$: .WORD   160014$ ; Конец стека пультового отладчика (для ЦП)
        ; Ее адрес в СОЗУ ЦП - 172612.
; Подпрограмма ввода символа с ожиданием (для ЦП). Адрес в СОЗУ ЦП - 172614.
173360$: MOV      #177560, R5 ; R5 = адрес регистра сост. К0 (клавиатура)
        $MFPM$ ; Чтение регистра состояния
        TSTB     R0 ; Есть символ для чтения ?
        BPL      173360$ ; Нет
        $MFPM$ ; Чтение символа
        RETURN
; Подпрограмма вывода символа на экран (для ЦП). Адрес в СОЗУ ЦП - 172632.
173376$: MOV      R0, -(SP) ; Сохранить код символа в стеке
173400$: MOV      #177564, R5 ; R5 = адрес регистра сост. К0 (терминал)
        $MFPM$ ; Чтение регистра состояния
        TSTB     R0 ; Можно посылать данные ?
        BPL      173400$ ; Нет
        MOV      (SP)+, R0 ; R0 = код символа
        MOV      #177566+2, R5 ; R5 = адрес рег. данных (терминал) + 2
        $MTPM$ ; Посылка кода символа по К0
        RETURN
; Указатели на ОЗУ пультового отладчика, адрес выводимой надписи, адрес открытой
; ячейки, разрешение/запрет вывода надписей. Начальное значение PS и SP.
; Объем ОЗУ в словах. Эти данные для ЦП. Их адрес в СОЗУ ЦП - 172660.
173424$: .WORD   160050,160100,160072,160104,200,1000,70000
; Подпрограмма выбора режима работы (ЦП). Ее адрес в сист.ОЗУ ЦП - 172676.

```

```

173442$:CALL    173376$      ; Посылка по каналу 0 результатов теста
173446$:TST     @#163772    ; Ожидание ответа от ПП
        BEQ      173446$
        BPL      173462$    ; Обнаружен контроллер НГМД
        JMP      @#162440$   ; На загрузку из сети
173462$:JMP     @#160744    ; На вывод меню ЗАГРУЗКА
; Адреса знач. ЦСП в векторах прер. ЦП (в формате загрузки в рег.адреса)
173466$: .WORD   70001,70003,70075,70077,70137,0
; Подпрограмма проведения инициализации
173502$:CALL    107754$      ; Программная инициализация драйвера терминала
        MOV      #RETURN,@#0 ; Установка векторов внутренних прерываний
        MOV      #600,R0
        MOV      #160200$,@#4
        MOV      R0,@#6
        MOV      #160210$,@#10
        MOV      R0,@#12
        MOV      #160220$,@#24
        MOV      R0,@#26
        CALL     103564$      ; Программная инициализация драйв. клавиатуры
        CALL     126136$      ; Программная иниц. драйвера магнитофона
        CALL     135456$      ; Программная иниц. драйвера граф. дисплея
        CALL     136004$      ; Программная иниц. драйвера принтера
        CALL     100000$      ; Начальные данные для меню УСТАНОВКА
        CALL     173744$      ; Уст.вект.прер.периферийных устройств
        MOV      SP,@#7040    ; Сохранить SP
        MOV      #173632$,@#4 ; Уст.вект.ошибки обращ. к каналу
        CALL     130712$      ; Программная иниц. драйвера дисковод
        MOV      #1,R0        ; Обнаружен контроллер НГМД
        BR       173642$
173632$:MOV     @#7040,SP      ; Восстановить SP
        MOV      #-1,R0       ; Контроллер НГМД не обнаружен
173642$:MOV     #<163772$/2>&77777,@#177010 ; Адрес ключа о присут. НГМД
        MOV      R0,@#177014  ; Передать ключ
        MOV      R0,@#7044
        BIC      #1400,@#177054 ; Включить сетевой таймер в ЦП и ПП
        BIS      #100,@#177066 ; Разрешение прерывания по RESET ЦП
        CLR      @#177712     ; Очистить буферный регистр таймера
        MOV      #1,@#177710  ; Пуск программируемого таймера
173706$:MOV     #23660,R0     ; R0 = адрес начала ОЗУ ПП пользователя
        CLR      (R0)+        ; Блок свободен
        CLR      (R0)+        ; Ссылка на предыдущий блок
        MOV      #54104,@R0   ; Размер свободного блока
        ADD      (R0)+,R0     ; Переход к фиктивной области
        INC      @R0          ; Признак занятой области
        RETURN
; Подпрограмма очистки блока памяти
173730$:CLR     R0            ; Заполняемое значение (0)
        MOV      (R5)+,R1     ; Адрес начала
        MOV      (R5)+,R2     ; Размер в словах
173736$:MOV     R0,(R1)+      ; Цикл очистки
        SOB      R2,173736$
        RTS      R5
; Подпрограмма установки векторов прерываний периферийных устройств
173744$:MOV     #174270$,@#30 ; Вектор прерывания команды EMT
        CLR      @#32
        MOV      #174334$,@#34 ; Вектор прерывания команды TRAP
        MOV      #200,@#36
        MOV      #174612$,@#100 ; Вектор прерывания сетевого таймера
        MOV      #200,@#102
        MOV      #175754$,@#330 ; Вектор прерывания K1 (приемник)
        BIS      #2,@#177066   ; Разрешить прерывания по K1 (приемник)
        MOV      #175762$,@#340 ; Вектор прерывания K2 (приемник)
        MOV      #200,@#342
        BIS      #4,@#177066   ; Разрешить прерывания по K2 (приемник)

```

```

CLR      @#23202      ; Счетчик принятых байтов по K2
MOV      #100/2,@#177010 ; Установить вектор прерывания сетевого
MOV      #104,@#177014 ; таймера канала ЦП
INC      @#177010
MOV      #200,@#177014
INC      @#177010
MOV      #RTI,@#177014
MOV      #7777,@#7052
CALL     176422$      ; Инициализация программируемого таймера
MOV      #176130$,@#314 ; Вект. прер. по команде RESET в канале ЦП
MOV      #200,@#316
INC      @#7102      ; Индикатор вызова фиктивной подпрограммы
MOV      #174222$,@#7126 ; Адрес фиктивной подпрограммы
RETURN

```

; * ОСНОВНОЙ ЦИКЛ ДИСПЕТЧЕРА ПРОЦЕССОВ *****

```

174152$:MOV      #2000,SP
          MOV      #2,@#7042      ; Передача кодов клавиатуры K0
174164$:MOV      #7060,R0      ; R0 = буфер флагов вызова
174170$:TST      (R0)+      ; Запрошен вызов
          BEQ      174170$      ; Нет
          MOV      R0,@#7056      ; Сохранить адрес флага вызова + 2
          MOV      22(R0),R1      ; R1 = адрес точки прерывания
          BNE      174214$      ; Было прерывание
          CALL     @<174232$-7062>(R0) ; Вызов по стандартному адресу
          BR       174164$      ; На начало цикла
174214$:CLR      22(R0)      ; Очистить адрес прерывания
          JMP      @R1      ; Переход по адресу прерывания

```

; Фиктивная подпрограмма

```

174222$:MOV      #174222$,@#7126 ; Установить точку прерывания фикт. п/п
          BR       174164$      ; На начало цикла

```

; Адреса стандартных подпрограмм обработки событий :

; обработка скэн-кодов клавиатуры, обработка данных K2, вывод символов на
; экран, переключение курсора, меню УСТАНОВКА, меню ЗАГРУЗКА, вывод на
; принтер, запуск п/п пользователя, запуск теста

```

174232$: .WORD    104306$,125030$,111144$,176056$,100040$,100742$,176076$
          .WORD    176116$,176460$

```

; Подпрограмма обработки команды TRAP 0

```

174254$:MOV      @#7056,R0      ; R0 = адрес флага
          MOV      (SP)+,22(R0) ; Сохранить СК
          MTPS     (SP)+      ; Установить ССП
          BR       174164$      ; На начало основного цикла

```

; * ПОДПРОГРАММА ОБРАБОТКИ ПРЕРЫВАНИЯ ПО КОМАНДЕ EMT *****

```

174270$:MOV      R5,-(SP)      ; Сохранить регистры
          MOV      R4,-(SP)
          MOV      R3,-(SP)
          MOV      R2,-(SP)
          MOV      R1,-(SP)
          MOV      R0,-(SP)
          MOV      14(SP),R5
          MOV      -(R5),R5      ; R5 = код команды EMT
          CALL     @<160000$-EMT>(R5) ; Вызов соответствующей подпрограммы
          MOV      (SP)+,R0      ; Восстановить регистры
          MOV      (SP)+,R1
          MOV      (SP)+,R2
          MOV      (SP)+,R3
          MOV      (SP)+,R4
          MOV      (SP)+,R5
          RTI

```

; * ПОДПРОГРАММА ОБРАБОТКИ ПРЕРЫВАНИЯ ПО КОМАНДЕ TRAP *****

```

174334$:MOV      @SP,R0
          MOV      -(R0),R0      ; R0 = код команды TRAP
          JMP      @<160160$-TRAP>(R0) ; Переход на соответствующую подпрограмму

```

; Обработка прерываний сетевого таймера (скроллинг назад)

```
174344$:MOV    R0, -(SP)      ; Сохранить R0
          ADD    @#7136, PC    ; Рулон плавный/дискретный ?
          BR     174476$      ; Рулон плавный
174354$:MOV    R1, -(SP)      ; Сохранить R1
          MOV    @#23170, R1   ; R1 = число видеострок для скроллинга
          NEG    R1            ; Перевод в положительное число
          ASL    R1            ; Умножение на 4
          ASL    R1
          MOV    @#2476, R0    ; R0 = адр.яч.информ. о перв. стр. экр.польз.
          ADD    #264.*4, R0   ; Плюс размер видимой части экрана
          CMP    R0, #4670     ; Достигли конца описания ?
          BLO    174424$      ; Нет
          BHI    174420$      ; Превысили
          MOV    #2500, @#4666 ; Посл. строка - первая в описании
          BR     174430$
174420$:SUB    #286.*4, R0     ; R0 = R0 - общ. число видеострок экр. польз.
174424$:MOV    R0, -2(R0)     ; Поместить в область описания
174430$:SUB    R1, R0         ; Вычесть число видеострок для скроллинга
          CMP    R0, #2500     ; Вышло за начало ?
          BHI    174444$      ; Нет
          ADD    #286.*4, R0   ; R0 = R0 + общ. число видеострок экр. польз.
174444$:MOV    #6742, -(R0)   ; Адрес описания нижней инф. строки
          MOV    @#2476, R0    ; R0 = адр.яч.информ. о перв. стр. экр.польз.
          SUB    R1, R0       ; Вычесть число видеострок для скроллинга
          CMP    R0, #2500     ; Вышли за начало ?
          BHIS   174470$      ; Нет
          ADD    #286.*4, R0   ; R0 = R0 + общ. число видеострок экр. польз.
174470$:MOV    R0, @#2476     ; Адрес описания начала экр. пользователя
          BR     174750$
174476$:MOV    @#2476, R0     ; R0 = адр.яч.информ. о перв. стр. экр.польз.
          SUB    #4, R0        ; R0 = R0 - 1 видеострока
          CMP    R0, #2500     ; Вышли за начало ?
          BHIS   174520$      ; Нет
          ADD    #286.*4, R0   ; R0 = R0 + общ. число видеострок экр. польз.
174520$:MOV    R0, @#2476     ; Адрес описания начала экр. пользователя
          ADD    #265.*4, R0   ; R0 = R0 + 265 строк
          CMP    R0, #4670     ; Вышли за конец
          BLO    174554$      ; Нет
          BNE    174550$      ; Да
          MOV    #2500, @#4666 ; Посл. строка - первая в описании
          BR     174560$
174550$:SUB    #286.*4, R0     ; R0 = R0 - общ. число видеострок экр. польз.
174554$:MOV    R0, -2(R0)     ; Поместить в область описания
174560$:SUB    #4, R0         ; R0 = R0 - 1 видеострока
          CMP    R0, #2500     ; На начале ?
          BNE    174576$      ;
          ADD    #286.*4, R0   ; R0 = R0 + общ. число видеострок экр. польз.
174576$:MOV    #6742, -(R0)   ; Адрес описания нижней инф. строки
          MOV    (SP)+, R0     ; Восстановить R0
          INC    @#23170      ; Уменьшить число видеострок для скроллинга
          BR     175054$
```

; * ПОДПРОГРАММА ОБРАБОТКИ ПРЕРЫВАНИЙ ОТ СЕТЕВОГО ТАЙМЕРА *****

```
174612$:TST    @#23170       ; Количество видеострок для скроллинга
          BEQ    175072$      ; Нет
          BMI    174344$      ; Скроллинг назад
          MOV    R0, -(SP)     ; Сохранить R0
          ADD    @#7136, PC    ; Рулон плавный/дискретный ?
          BR     174762$      ; Рулон плавный
174632$:MOV    R1, -(SP)      ; Сохранить R1
          MOV    @#23170, R1   ; R1 = число видеострок для скроллинга
          ASL    R1            ; Умножение на 4
          ASL    R1
          MOV    @#2476, R0    ; R0 = адр.яч.информ. о перв. стр. экр.польз.
```

	ADD	#264.*4,R0	; Плюс размер видимой части экрана
	CMP	R0,#4670	; Достигли конца описания ?
	BLO	174700\$; Нет
	BHI	174674\$; Превысили
	MOV	#2500,@#4666	; Посл. строка - первая в описании
	BR	174704\$	
174674\$:	SUB	#286.*4,R0	; R0 = R0 - общ. число видеострок экр. польз.
174700\$:	MOV	R0,-2(R0)	; Поместить в область описания
174704\$:	ADD	R1,R0	; Прибавить число видеострок для скроллинга
	CMP	R0,#4670	; Вышли за конец ?
	BLOS	174720\$; Нет
	SUB	#286.*4,R0	; R0 = R0 - общ. число видеострок экр. польз.
174720\$:	MOV	#6742,-(R0)	; Адрес описания нижней инф. строки
	MOV	@#2476,R0	; R0 = адр.яч.информ. о перв. стр. экр.польз.
	ADD	R1,R0	; Прибавить число видеострок для скроллинга
	CMP	R0,#4670	; Вышли за конец ?
	BLO	174744\$; Нет
	SUB	#286.*4,R0	; R0 = R0 - общ. число видеострок экр. польз.
174744\$:	MOV	R0,@#2476	; Адрес описания начала экр. пользователя
174750\$:	MOV	(SP)+,R1	; Восстановить регистры
	MOV	(SP)+,R0	
	CLR	@#23170	; Число строк для скроллинга = 0
	BR	175056\$	
174762\$:	MOV	@#2476,R0	; R0 = адр.яч.информ. о перв. стр. экр.польз.
	BIC	#1,2(R0)	; Сбросить бит переключения курсора
	MOV	2(R0),@#2476	; Сдвиг на 1 строку вверх
	ADD	#264.*4,R0	; R0 = R0 + 264 видеостроки
	CMP	R0,#4670	; Вышли за конец ?
	BLO	175034\$; Нет
	BNE	175030\$; Да
	MOV	#2500,R0	; R0 = первый элемент описания
	MOV	R0,@#4666	; Посл. строка - первая в описании
	BR	175040\$	
175030\$:	SUB	#286.*4,R0	; R0 = R0 - общ. число видеострок экр. польз.
175034\$:	MOV	R0,-2(R0)	; Поместить в область описания
175040\$:	MOV	#6742,2(R0)	; Адрес описания нижней инф. строки
	MOV	(SP)+,R0	; Восстановить R0
	DEC	@#23170	; Уменьшить число видеострок для скроллинга
175054\$:	BNE	175306\$; Еще остались
175056\$:	MOV	@#22750,@#23164	; Разрешение/запрет использования курсора
	INC	@#7066	; Уст.флаг вызова п/п перекл. курсора
	BR	175306\$	
175072\$:	ADD	@#7134,PC	; Перекл. курсор/не изм. курсор/выдержка вр.
	BR	175150\$; Перекл. курсор, поменять его координаты
175102\$:	BR	175306\$; Не изм. состояния курсора
175104\$:	DECB	@#23166	; Уменьшить счетчик выдержки времени
	BGT	175306\$; Время еще не окнчилось
	BEQ	175120\$; Время закончилось
	MOV	#110060,@#23162	; Загрузить временной интервал
175120\$:	MOV	@#23162,@#23166	; Загрузить значение в счетчик
	SWAB	@#23162	; Поменять параметры выдержки(горит/не горит)
	BMI	175300\$; Включить курсор
	ASL	@#23166	
175140\$:	MOVB	#177,@23160	; Убрать курсор с экрана
	BR	175306\$	
175150\$:	MOV	R0,-(SP)	; Сохранить регистры
	MOV	R1,-(SP)	
	MOV	R5,-(SP)	
	MOV	@#23150,R5	; R5 = адрес текущей таблицы видеопараметров
	MOV	@R5,R0	; R0 = текущий адрес видео
	SUB	@32(R5),R0	; R0 = тек.адрес - адр.нач.строки = позиция X
	MOV	66(R5),R1	; R1 = режим экрана (80/40/20/10)
	BR	175200\$	
175176\$:	ASL	R0	; Приведение координат в соотв. с режимом

```

175200$:ROR      R1
          BCC      175176$
          MOV      R0,@#23156      ; Сохранить координату X
          MOV      60(R5),R1      ; R1 = адрес списка строк, где расп. курсор
          BIC      #1,6(R1)      ; Выключить курсор
          BIC      #1,52(R1)
          MOV      32(R5),60(R5)  ; Новые координаты курсора
          MOV      60(R5),R1
          BIS      #1,6(R1)      ; Включить курсор
          BIS      #1,52(R1)
          MOV      #4,@#7134      ; Режим мигания курсора
          MOV      (SP)+,R5      ; Восстановить регистры
          MOV      (SP)+,R1
          MOV      (SP)+,R0
          CLR      @#23166      ; Сброс временных параметров курсора
          BR       175140$
175300$:MOVB     @#23156,@23160  ; Поместить курсор в позицию X
175306$:TST      @#7130          ; Двигатель НГМД вращается ?
          BEQ      175326$      ; Нет
          DEC      @#7130      ; Уменьшить счетчик вращения двигателя
          BNE      175326$      ; Еще не конец
          CALL     @7132        ; Вызов п/п выключения двигателя НГМД
175326$:DEC      @#7046          ; Уменьшить сч. длительности звукового сигн.
          BNE      175342$      ; Еще не конец
          BIC      #17600,@#177716 ; Выключить звуковой сигнал
175342$:TSTB     @#7227          ; Есть данные для автоповтора клавиш
          BEQ      175376$      ; Да
          DECB     @#7227      ; Уменьшить сч. автоповтора
          BNE      175376$      ; Еще не конец
          INC      @#7060      ; Увеличить флаг вызова п/п обр. соб. клав.
          BIS      #200,@#177716 ; Подать полож.импульс на динамик (щелчок)
          MOV      #1,@#7046      ; Длительность импульса
175376$:RTI
; Подпрограмма заполнения ячеек информацией о длительности вращения
; двигателя НГМД и адресе п/п выключения двигателя
175400$:MOV      (R5)+,@#7130    ; Длительность вращения двигатель НГМД
          MOV      (R5)+,@#7132  ; Адрес п/п выключения двигателя НГМД
          RTS      R5
; *** ПОДПРОГРАММА ОБРАБОТКИ ПРЕРЫВАНИЙ ОТ КЛАВИАТУРЫ ***
175412$:MOVB     @#177702,@7200  ; Занести скэн-код клавиши в буфер
          BMI      175436$      ; Клавиша отжата
          BIS      #200,@#177716 ; Подать полож.импульс на динамик (щелчок)
          MOV      #2,@#7046      ; Длительность импульса
175436$:CMPB     @7200,#4        ; Нажата клавиша <СТОП> ?
          BEQ      175466$      ; Да
          INC      @#7200      ; Передвинуть указатель на след. позицию
          BIC      #10,@#7200
          INC      @#7060      ; Увеличить флаг вызова п/п обр. соб. клав.
          RTI
175466$:MOV      R0,-(SP)      ; Сохранить регистры
          MOV      R1,-(SP)
          MOV      R2,-(SP)
          CALL     104054$      ; Очистка буферов клавиатуры
          MOV      (SP)+,R2      ; Восстановить регистры
          MOV      (SP)+,R1
          MOV      (SP)+,R0
          BIS      #20,@#177716 ; Установить сигнал HALT в ЦП
          RTI
; Подпрограмма исполнения команды TRAP 2
175516$:CLR      @#7230          ; Запрет вызова TRAP 2 п/п обсл. клав.
          ADD      @#7042,PC      ; Передача данных по К0/в R0
          BR       175540$      ; Передача в R0
175530$:BIT      #10,@#177076    ; Готовность передатчика К0
          BEQ      175646$      ; Не готов

```



```

; *** ПОДПРОГРАММА ОБРАБОТКИ ПРЕРЫВАНИЙ ОТ ПЕРЕДАТЧИКА КАНАЛА 0(КЛАВИАТУРА) ***
175540$:MOV      R5, -(SP)          ; Сохранить R5
      MOV      @#7206, R5          ; R5 = указатель для чтения с буфера ASCII
175546$:CMP      R5, @#7210          ; Буфер пуст ?
      BEQ      175650$             ; Да
      TSTB     1(R5)               ; Данное слово содержит одиночный код ?
      BEQ      175620$             ; Да
      CLR      -(SP)
      MOVB     @(R5), @SP          ; Поместить в стек очередной символ из строки
      BNE      175606$             ; Еще не конец строки символов
      CMP      (R5)+, (SP)+        ; Указатель на след. позицию
      BIC      #40, R5
      MOV      R5, @#7206
      BR       175546$
175606$:INC      @R5                ; К следующему символу в строке
      MOV      #177, R5
175614$:SOB      R5, 175614$        ; Задержка
      BR       175632$
175620$:MOV      (R5)+, -(SP)        ; Поместить символ в стек
      BIC      #40, R5
      MOV      R5, @#7206
175632$:ADD      @#7042, PC          ; Передача данных по K0/в R0
      BR       175660$             ; Передача в R0
175640$:MOV      (SP)+, @#177070    ; Передача символа в K0
175644$:MOV      (SP)+, R5          ; Восстановить R5
175646$:RTI
175650$:MOV      #2, @#7230         ; Разрешение вызова TRAP 2 п/п обл. клав.
      BR       175644$
175660$:MOV      (SP)+, R0          ; Передача символа в R0
      BR       175644$
; Подпрограмма исполнения команды TRAP 4
175664$:CLR      @#22546            ; Запрет вызова TRAP 4 п/п обл. терминала
      BIT      #10, @#177066        ; Готовность приемника K0
      BEQ      175742$             ; Не готов
; *** ПОДПРОГРАММА ОБРАБОТКИ ПРЕРЫВАНИЙ ОТ ПРИЕМНИКА КАНАЛА 0(ТЕРМИНАЛ) ***
175700$:ADD      @#7042, PC          ; Передача данных по K0/в R0
      BR       175752$             ; Передача в R0
175706$:CMP      @#7064, #177        ; Буфер заполнен полностью ?
      BEQ      175744$             ; Да
      MOVB     @#177060, @22544      ; Занести символ в буфер
      INC      @#22544              ; Подкорректировать указатель буфера
      BIC      #200, @#22544
      INC      @#7064              ; Увеличить флаг вызова п/п обл. терм.
175742$:RTI
175744$:MOV      #2, @#22546         ; Разрешение выз. TRAP 4 п/п обл. терминала
175752$:RTI
; *** ПОДПРОГРАММА ОБРАБОТКИ ПРЕРЫВАНИЙ ОТ ПРИЕМНИКА КАНАЛА 1(ПРИНТЕР) ***
175754$:INC      @#7074             ; Уст. флаг вызова п/п обл. принтера
      RTI
; *** ПОДПРОГРАММА ОБРАБОТКИ ПРЕРЫВАНИЙ ОТ ПРИЕМНИКА КАНАЛА 2 ***
175762$:MOV      R0, -(SP)          ; Сохранить R0
      MOV      @#23202, R0          ; R0 = счетчик принятых байтов
      BMI      176046$             ; В состоянии инициализации
      CMP      R0, #2              ; Адрес МП передан
      BHIS     176016$             ; Да
      MOVB     @#177064, 23200(R0) ; Передача данных в буфер (адрес МП)
176006$:INC      @#23202            ; Увеличить счетчик принятых байтов
176012$:MOV      (SP)+, R0          ; Восстановить R0
      RTI
176016$:BNE      176034$            ; Принято 3 байта
      TSTB     @#177064            ; 3-й байт равен нулю ?
      BNE      176006$             ; Нет
      COM      @#23202             ; Инвертировать счетчик (инициализация канала)
      BR       176012$

```

```

176034$:INC      @#7062          ; Установить флаг вызова п/п обл. K2
176040$:CLR      @#23202        ; Очистить счетчик
BR              176012$
176046$:TSTB     @#177064       ; Принятый байт равен нулю ?
BEQ             176012$         ; Да
BR              176040$

; Подпрограмма переключения режима курсора
176056$:CLR      @#7066         ; Очистка флага вызова п/п
ADD             @#23164,PC      ; Разрешение / запрет использования курсора
RETURN          ; Запрет использования курсора
176070$:CLR      @#7134         ; Поменять позицию курсора и включить его
RETURN

; Подпрограмма печати символа на принтере, переданного по K1
176076$:CLR      @#7074         ; Очистить флаг вызова п/п
MOV             @#177062, -(SP) ; Поместить символ в стек
CALL            135722$         ; Печать символа на принтере
TST             (SP)+
RETURN

; Подпрограмма вызова процедуры пользователя
176116$:CALL     @7140          ; Вызов процедуры пользователя
CLR             @#7076         ; Очистить флаг вызова п/п
RETURN

; *** ПОДПРОГРАММА ОБРАБОТКИ ПРЕРЫВАНИЯ ПО ПОЯВ. НА КАНАЛЕ ЦП СИГНАЛА RESET ***
176130$:RTI

; Подпрограмма запроса блока памяти
176132$:MOV      R0,R1          ; R1 = размер запрашиваемого блока
BMI             176142$         ; Найти максимальный блок
ROR             R1              ; Приведение к четному адресу
ADC             R0
176142$:MOV      R2, -(SP)      ; Сохранить R2
CLR             R2              ; R2 = адрес найденного блока
CLR             -(SP)          ; Размер найденного блока
176150$:MOV      #23660,R1      ; R1 = адрес начала ОЗУ пользователя
176154$:TST      (R1)+          ; Блок занят ?
BEQ             176176$         ; Нет
TST             (R1)+
176162$:ADD      (R1)+,R1       ; Переход к следующему блоку
BPL             176154$         ; Еще не конец ОЗУ
TST             R2              ; Найден свободный блок ?
BNE             176150$         ; Да
CLR             R0              ; Нет свободной памяти
BR              176272$
176176$:TST      (R1)+
CMP             R0,@R1          ; Размер блока равен или больше запраш. ?
BLOS            176226$         ; Да
CMP             R1,R2           ; Это найденный максимальный блок
BEQ             176222$         ; Да
CMP             @R1,@SP         ; Меньше запраш. размера ?
BLOS            176162$         ; Да
MOV             R1,R2           ; R2 = адрес найденного макс. блока
MOV             @R1,@SP         ; Сохранить размер блока
BR              176162$
176222$:MOV      @SP,R0         ; R0 = размер найденного макс. блока
BR              176232$
176226$:INC      -4(R1)         ; Пометить блок, как занятый
176232$:MOV      (R1)+,R2       ; R2 = размер блока
SUB             #6,R2           ; Отнять размер заголовка
CMP             R0,R2           ; Запраш. размер больше реального ?
BHS             176272$         ; Да
MOV             R1,R2
ADD             R0,R2           ; R2 = адрес следующего блока
CLR             (R2)+           ; Пометить блок, как свободный
MOV             R1,@R2
SUB             #6,(R2)+        ; Ссылка на предыдущий блок

```

```

MOV      -(R1),@R2      ; Вычисление размера блока
SUB      R0,@R2
SUB      #6,@R2
MOV      R0,(R1)+      ; Размер выделенного блока
176272$:TST (SP)+
MOV      (SP)+,R2      ; Восстановить R2
RETURN

; Подпрограмма освобождения блока памяти
176300$:TST R1      ; Освободить блок или всю память ?
BPL      176310$      ; Освободить блок
JMP      173706$      ; Освободить всю память
176310$:MOV #23660,R0 ; R0 = адрес начала ОЗУ пользователя
176314$:ADD #6,R0      ; Плюс размер заголовка
BMI      176374$      ; Достигнут конец ОЗУ
CMP      R0,R1      ; Адрес освоб. блока равен текущему
BEQ      176342$      ; Да
ADD      -2(R0),R0    ; Переход к следующему блоку
BR       176314$
176334$:MOV @R1,R1      ; R1 = адрес предыдущего блока
ADD      #6,R1      ; Плюс размер заголовка
176342$:CLR -6(R1)      ; Пометить блок, как свободный
TST      -(R1)
MOV      R1,R0      ; R0 = адрес блока (поле размера)
ADD      (R0)+,R0    ; Переход к следующему блоку
TST      @R0      ; Блок занят ?
BNE      176370$      ; Да
ADD      4(R0),@R1    ; Прибавить размер к предыд. (слияние)
ADD      #6,@R1      ; Плюс размер заголовка
176370$:TST @-(R1)      ; Блок выше свободный ?
BEQ      176334$      ; Да
176374$:RETURN

; Подпрограмма резервирования программируемого таймера за процессом
176376$:TST @#7050      ; Программируемый таймер занят ?
BEQ      176416$      ; Да
CLR      @#177710      ; Останов таймера
CLR      @#7050      ; Таймер занят
RETURN
176416$:SEC      ; Признак занятости таймера
RETURN

; Подпрограмма освобождения процессом программируемого таймера
176422$:MOV #4,@#7050      ; Признак свободного таймера
MOV      #174612$,@#304 ; Восстановить вектор прерывания
MOV      #200,@#306
CLR      @#177712      ; Очистить буферный регистр таймера
MOV      #1,@#177710    ; Пуск таймера
RETURN

; Подпрограмма запуска тестов в ПП
176460$:CALL 176472$      ; Запустить тесты
CLR      @#7100      ; Очистить флаг вызова п/п
RETURN

; Подпрограмма вызова тестов и передачи результатов в ОЗУ ЦП
176472$:MTPS #200      ; Запретить прерывания
MOV      #176656$,@#300 ; Переназначить вектор прерывания клавиатуры
CALL     160410$      ; Тест ПЗУ
CLR      R1      ; R1 = 0 (смещение в буфере ошибок)
176512$:ASR R0
BCS      176522$      ; Ошибка банка ПЗУ
BNE      176526$      ; Еще есть ошибки
BR       176532$
176522$:CALL 176626$      ; Установка счетчика ошибок в ОЗУ ЦП
176526$:INC R1      ; Переход к след.ячейке в ОЗУ ЦП
BR       176512$
176532$:CALL 170632$      ; Тест процессора
MTPS     #0      ; Разрешить прерывания

```

```

TST      R0          ; Есть ошибки ?
BEQ      176556$     ; Нет
MOV      #5,R1       ; R1 = 5 (смещение в буфере ошибок)
CALL     176626$     ; Установка счетчика ошибок в ОЗУ ЦП
176556$:MOV    #-1,R0
CALL     176132$     ; Получить адрес наиб. блока памяти
MOV      R1,R4       ; R4 = начальный адрес
MOV      R1,R5
ADD      R0,R5       ; R5 = конечный адрес
CALL     170224$     ; Тест оперативной памяти
TST      R0          ; Есть ошибки
BEQ      176614$     ; Нет
MOV      #4,R1       ; R1 = 4 (смещение в буфере ошибок)
CALL     176626$     ; Установка счетчика ошибок в ОЗУ ЦП
176614$:MOV    #177,R0
CALL     104160$     ; Посл. ЦП по К0 символ с кодом 177(тест окон.)
RETURN

```

; Подпрограмма установки счетчика ошибок в ОЗУ ЦП

```

176626$:MOV    #<177700/2>&77777, @#177010 ; Адрес буфера счетчиков ошибок
ADD      R1, @#177010 ; Прибавить смещение
CMP      #-1, @#177014 ; Значение счетчика достигло максимума
BEQ      176654$     ; Да
INC      @#177014    ; Увеличить значение счетчика ошибок

```

176654\$:RETURN

; Подпрограмма обработки прерываний от клавиатуры во время тестирования

```

176656$:CMP    #4, @#177702 ; Нажата клавиша <СТОП> ?
BNE      176672$     ; Нет
JMP      @160000$    ; Перезапуск компьютера

```

176672\$:RTI

◆

```

; *****
; *
; *
; *          КОНТРОЛЬНЫЕ СУММЫ БЛОКОВ ПЗУ
; *
; *
; *          Адреса 176770 - 176777
; *
; *
; *****

```

. =176770

```

176770$: .WORD    63160          ; Адреса 100000 - 117777
176772$: .WORD    133314        ; Адреса 120000 - 137777
176774$: .WORD    162125        ; Адреса 140000 - 157777
176776$: .WORD    103607        ; Адреса 160000 - 176775

```

; * * * * * E n d o f R O M * * * * *

.END

Retrieved from "<http://www.felixl.com>"