



НАУЧНО-ТЕХНИЧЕСКИЙ КООПЕРАТИВ "ИНФОРКОМ", 107241, МОСКВА, Б-241 а/я 37

## Раздел для начинающих

Продолжение, начало на с. 23,45.

Сегодня мы продолжаем печатать небольшие программы для начинающих, присланные нашим постоянным читателем из г. Дубна, тов. Скитевым В.Н.

### Спирали.

```
10 LET N=36:LET A=0:LET D=2*PI/N
20 CIRCLE 128,88,80
30 FOR I=1 TO N
40 LET X =40*COS A
50 LET Y=40*SIN A
60 PLOT 128, 88:DRAW X,Y,-PI:DRAW X,Y,PI
70 LET A=A+D
80 NEXT I
```

### Звезда.

```
10 PLOT 128,174
20 DRAW 55,-160
30 DRAW -137,100
40 DRAW 164,0
50 DRAW -135,-100
60 DRAW 53,160
```

### Парабола.

```
10 PLOT 1,88
20 DRAW 254,0
30 FOR A=1.5 TO .4 STEP -.1
40 PLOT 200,5
50 DRAW 0.165,A
60 LET C=(165/2) / (TAN A)
70 DRAW -C,-(165/2)
60 DRAW C,-(165/2)
90 NEXT A
```

### Восход.

```
10 BORDER RND*6
20 INK RND*7
30 PAPER RND*6
40 CLS
50 LET Z=RND*10+2
60 FOR X=0 TO 174 STEP Z
70 PLOT 126,0
80 DRAW -126,X
90 BEEP .01,X/3
100 NEXT X
110 FOR X=-127 TO 127 STEP Z
120 PLOT 128,0
130 DRAW X,175
140 BEEP .01,60
```

```

150 NEXT X
160 FOR X=174 TO 0 STEP -Z
170 PLOT 128,0
180 DRAW 127,X
190 BEEP .01,X/3
200 NEXT X
210 PAUSE 200
220 GO TO 10

```

**50-100** левая часть экрана

**100-150** центральная часть экрана

**160-200** правая часть экрана

Полное окрашивание экрана монитора. (2-красный, 4-зеленый, 1-синий).

```

10 CLS
20 BORDER 2: PAPER 2
30 FOR N = 0 TO 21
40 PRINT "
50 NEXT N

```

В строке 40 в кавычках 32 пробела.

Программы, представленные ниже достаточно сложны, поэтому сначала введите основную часть программы, а затем после проверки вводите следующую часть.

**Пирамиды.**

```

10 BORDER 0
20 PAPER 1
30 INK 6: CLS
50 FOR Y = 0 TO 20 STEP 2
60 PLOT 0, Y
70 DRAW 255,0
80 NEXT Y
90 FOR N=100 TO 220 STEP 30
100 FOR X=-10-N/10 TO 10+N/10
110 PLOT N, 35+N/10
120 DRAW X, -N/4
130 NEXT X
140 NEXT N

```

**Продолжение пирамид.**

```

190 FOR F=1 TO 30
200 LET X=RND*255
210 LET Y=RND*104+71
220 LET L=INT(175-Y)/8
230 LET C=INT(X/8)
240 PLOT 0,0
250 DRAW OVER 1; X,Y
260 BEEP .01,X/4
270 PLOT 0,0
280 DRAW OVER 1; X,Y
290 PRINT AT L,C;"*"
300 NEXT F

```

**Мыльный пузырь.**

```

10 BORDER 1
20 FOR Z=3 TO 10
30 LET H=INT(RND*25)
40 LET V=INT(RND*21)
50 PRINT INK 2;PAPER 6; FLASH 1; AT V,H;"X"
60 NEXT Z
70 LET X=1: LET Y=1
80 PRINT AT V,H;" "
90 LET V=V+Y

```

```

100 LET H=H+X
110 IF H=0 OR H=31 THEN LET X=-X: BEEP .02,24
120 IF V=0 OR V=21 THEN LET Y=-Y: BEEP .02,12
130 IF SCREEN$(V,H)="X" THEN PRINT INK 1; PAPER 5; AT V,H;"*": BEEP .5,25:STOP
140 PRINT AT V,H;"o"
150 PAUSE 2
160 GO TO 80

```

С очередной почтой поступила и довольно поучительная программа "ЦВЕТОМУЗЫКА", позволяющая получить на экране разноцветные узоры, изменяющиеся в такт музыке.

Ее прислал наш читатель из г. Минска Кипорук А.А.

Пользоваться программой очень просто. После того, как программа считана с магнитофона и запущена, вставьте в магнитофон кассету с музыкальной записью и включите его. Динамику изображения можно менять, изменяя уровень входного сигнала.

```

5 REM Настройка переменных.
10 PAPER 0: BORDER 0: CLS
15 LET a=65022: LET s0=255: LET t=0: LET x=0: LET y=21: LET c=1: LET n=2: LET d=0: LET d1=0:
  LET d2=0
19 REM Анализ изменения сигнала на входе
20 LET s= IN a
25 INK 0: GO SUB 5000: INK c
30 IF s<>s0 THEN GO SUB n*1000
40 LET t=t+1
50 GO TO 20
999 REM Вывод узора 1-го типа.
1000 IF t=1 THEN INK 5: GO SUB 5000
1003 LET x=x+(x<31 AND t=4) - (x>0 AND t=3)
1006 LET y=y+(y<21 AND t=2) - (y>0 AND t=1)
1010 IF t>=5 THEN LET y=INT (RND*21): LET x=INT (RND*30)
1020 IF t>6 THEN CLS:LET n=1+INT(RND*3): GO TO 1055
1025 IF t<>4 THEN INK c: GO TO 1050
1030 LET c=1+INT (RND*7)
1040 INK c: BRIGHT INT (RND*2)
1050 PRINT AT y,x;"█";AT y,30-x;"█";AT 21-y,x;"█"; AT 21-y,30-x;"█"
1055 LET s0=s: let t=0
1060 RETURN
1999 REM Вывод узора 2-го типа
2000 IF t=1 THEN LET c=5: GO SUB 5000: GO TO 2900
2003 INK 0: GO SUB 5100
2005 IF t>5 THEN CLS: LET n=1+INT (RND*3): GO TO 2900
2010 BRIGHT INT (RND*2): INK t
2020 LET d1=t+INT (RND*4): LET d2=d1+INT (RND*5)
3040 GO SUB 5100
2900 LET s0=s: LET t=0
2910 RETURN
2999 REM Вывод узора 3-го типа
3000 IF t=1 THEN LET c=5:GO SUB 5000
3005 IF t>5 THEN CLS: LET n=1+INT(RND*3): LET x=20: LET y=30:GO TO 3900
3010 IF t>4 THEN CLS
3050 LET c=1+INT (RND*7)
3060 LET x=INT (RND*256); LET y=INT (RND*176)
3070 INS c: PLOT x,y
3080 IF t<3 THEN DRAW -INT (RND*X),-INT (RND*y)
3090 IF t>2 THEN CIRCLE 20+INT (RND*210),20+INT (RND*130,INT (RND*20)
3900 LET s0=s: LET t=0: LET x=15: LET y=10
3910 RETURN
4999 REM Вывод угловых квадратов
5000 PRINT AT 0,0;"█"; AT 0,31;"█"; AT 21,0;"█"; AT 21, 31;"█"
5010 RETURN
5100 PRINT AT 10+d1,15+d2;"█" AT 10+d1,15-d2;"█"; AT 10-d1,15=d2;"█";AT 10-d1,15-
  d2;"█"
5120 RETURN

```

# Мега-Бейсик

(Продолжение)

## Управление выполнением программы

### Процедуры.

Мега-Бейсик обеспечивает работу с процедурами, вызов которых осуществляется по их имени, точно так же, как и после команды, после имени процедуры может быть записана серия арифметических выражений или чисел. Эти выражения определяют необходимые переменные для работы в процедуре.

Недостатком процедур в Мега-Бейсике является то, что они не могут работать с локальными переменными: их переменные являются общими для всей программы.

Используя процедуры, можно разделить всю программу на ряд частных задач. Каждая процедура отлаживается отдельно, а затем все они объединяются в программу. Начало процедуры задается символом '@', следом за которым записывается имя процедуры. После имени процедуры записываются необходимые параметры. Окончание процедуры обозначается оператором `ENDPROC_`. Как только будет достигнут этот оператор, управление выполнением программы будет передано оператору, записанному за оператором вызова процедуры.

После оператора `ENDPROC_` можно записать имя процедуры:

```
9000 @DISPLAY_A,A$
9010 PAPER A: INK 9
9020 MODE_4: STIPPLE_5
9030 PRINT A$
9040 ENDFROC_DISPLAY
```

Это пример процедуры. В строке 9010 определяется имя процедуры `DISPLAY` и, как видно, для ее работы необходимо определить две переменные - `A` и `A$`. В строке 9030 определяется размер и оттенок символов, выводимых на экран. Для вызова процедуры необходимо записать, например:

```
DISPLAY_2, "MEGABASIC"
```

Процедуры могут вызываться и из процедур.

### REPEAT-UNTIL

Мега-Бейсик обеспечивает работу с циклами типа `REPEAT-UNTIL`. Оператор `REPEAT` определяет начало цикла. После оператора `UNTIL` должна быть записана односимвольная переменная. Если значение этой переменной равно нулю, то цикл прерывается и выполнение программы переходит к оператору, записанному после последнего оператора `REPEAT`, если же эта переменная отлична от 0, то выполняется следующий оператор программы. Допускается вложение до 10 циклов типа `REPEAT-UNTIL`.

### Процедурный стек

Для хранения номеров строк операторов, к которым выполняется возврат после использования процедур и циклов `REPEAT-UNTIL`, применяется специальный стек. Когда вызывается процедура, номер строки, следующий за оператором вызова, записывается в стек, что позволяет системе знать адрес возврата после окончания процедуры.

Аналогично и для циклов типа `REPEAT-UNTIL`. Здесь также в стек записываются номера строки и оператора, к которым осуществляется возврат после оператора `UNTIL`.

После выполнения оператора `ENDPROC_` и перехода к строке, номер которой хранился в стеке, этот номер удаляется со стека. Любая попытка записать в стек больше 10 номеров заканчивается сообщением "`PROC STACK OVERFLOW`" (Процедурный стек переполнен).

Команды `POP` и `PUSH`. Оператор `POP_` служит для "выталкивания" значений со стека, оператор `PUSH_` для записи значения в стек. За оператором `PUSH_` следуют 2 числа: номер

оператора в строке и номер строки. Для очистки стека перед началом выполнения программ необходимо записать команду PCLEAR.

Команда BRANCH\_. По этой команде после выполнения каждой программной строки будет вызываться подпрограмма, записанная в строках основной программы. Начальная строка этой подпрограммы указывается после команды BRANCH, а конец подпрограммы определяется оператором ENDPROC\_.

Оператор MTASK\_. Это оператор многозадачного режима. Он позволяет выполнять программу одновременно из двух разных мест. Считается, что программа как бы разделена на две части: первая начинается после строки, где записан оператор MTASK\_, а вторая - со строки, номер которой записан после оператора MTASK\_. Мега-Бейсик выполняет поочередно строки из каждой части программы.

## Отладка и редактирование программ

### Операторы BRANCH\_, TRON\_, TROFF\_ и SPEED\_.

Оператор BRANCH\_ оказывается весьма полезным при отладке программ. Если, к примеру, Вас интересует какая-то переменная, то Вы можете с помощью этой команды вызывать подпрограмму печати значения этой переменной после каждой программной строки. Другой способ проверить выполнение программы - выводить на экран номер выполняемой строки, что возможно по команде TRON\_. При этом номер выполняемой строки программы будет выведен в нижний угол экрана, команда TROFF\_ отменяет такой режим. Связана с этими двумя командами также команда SPEED\_, после которой записано число, определяющее скорость выполнения программных строк. Максимально возможное число - 255. При этом выполнение программы будет приостановлено до тех пор, пока пользователь не нажмет любую клавишу.

### Команды AUTO и DELETE\_.

При вводе больших программ весьма полезным может быть автоматический ввод номера следующей строки после нажатия клавиши ENTER. Такой режим работы возможен при применении команды AUTO\_, после которой следует 2 числа, разделенных запятой. Первое число задает начальный номер строки, с которого начнется автоматическая нумерация, а второе - шаг нумерации.

Остановить режим автонумерации можно так: EXTEND SYMB.SHIFT 'L'. Часто бывает необходимо удалить целый блок программных строк. Эту задачу можно решить, используя команду DELETE\_. После этой команды записывают два числа, являющиеся соответственно начальным и конечным номерами строк удаляемого блока.

### Команды BRON\_ и BROFF\_.

Желающие защитить свои программы от любопытных глаз, найдут полезным возможность отключать клавишу BREAK. это может быть сделано при применении команды BROFF\_. Возвращение клавише ее возможностей осуществляется по команде BRON\_.

Команда RESTART. Многие версии языка Бейсик имеют команду ON ERROR GO TO, позволяющую не прерывать выполнение программы в случае обнаружения ошибки, а осуществлять обход строки с ошибкой. Аналогичная команда имеется и в Мега-Бейсике. Это команда RESTART\_, после которой записывается номер строки, к которой осуществляется переход в случае обнаружения ошибки. По команде RESTART\_OFF такой режим отменяется. После обработки ошибки с помощью команды RESTART, некоторая полезная информация сохраняется в ячейках памяти: 59873/4 - номер строки, в которой обнаружена ошибка; 59875 - номер оператора внутри строки с ошибкой; 59862 - код обнаруженной ошибки.

## Звук.

Мега-Бейсик открывает для Вас новые возможности создания звуков на Вашем ZX SPECTRUM. Для этого необходимо использовать команду PLAY\_ и генератор звука ISG

## PLAY\_n,l,s,d,f

Здесь:

n=0 - ноты;

n=1 - белый шум;

l - длительность звучания на каждом шаге;

s - значение начальной звуковой частоты;

d - число шагов;

f - изменение частоты после каждого шага.

## ISG

Если команда PLAY\_ только улучшает возможности команды BEEP, то использование звукового генератора ISG позволит получить новые эффекты. Дело в том, что при выполнении программы в ZX SPECTRUM происходят прерывания 50 раз в секунду. Во время обслуживания прерывания специальная программа в машинных кодах производит сканирование клавиатуры. В это же время и работает ISG. В Мега-Бейсике имеется ряд команд, которые могут управлять этим генератором.

SOUND\_n,a,b,c,d - команда для работы с звуковым буфером.

Здесь:

n=0 очистка звукового буфера перед очередной загрузкой;

n=1 добавление новой информации в звуковой буфер;

a=0 воспроизведение нот;

a=1 воспроизведение шума;

b - шаг звуковой частоты;

c - число шагов в последовательности нот;

d - число повторений последовательности нот.

Другие команды:

SOFF - отключение ISG;

SON - включение ISG;

SREP\_n - количество повторений звуковой последовательности; если n=0, то данные, записанные в звуковом буфере будут использоваться только один раз, в противном случае - многократно.

Когда выполняется команда PLAY, ISG автоматически отключается. Для включения ISG применяйте команду SON. Более сложные звуковые эффекты можно получить, если замедлить выполнение программы. Этот короткий фрагмент программы демонстрирует работу ISG:

```
10 SOUND_ 0,0,1,20,255
20 SREP_1
30 SON
40 MODE_4: STIPPLE_6: FONT_2
50 VDU_(128+RND*15)
60 PAPER RND*7: INK 9
70 GO TO 50
```

## Машинные коды и МЕГА-БЕЙСИК

### Команды DOKE и CALL.

Мега-Бейсик позволяет включать в текст основной программы и фрагменты в машинных кодах. Мега-Бейсик занимает в ОЗУ участок памяти, начиная с адреса 45000.

Поэтому все программы в машинных кодах должны быть размещены в ОЗУ до этого адреса. Для того, чтобы записать в память двухбайтное число, можно воспользоваться командой DOKE, которая поддерживается в Мега-Бейсике. После этой команды должны быть записаны два числа: первое определяет адрес, куда записываются данные, а второе -

записываемое число.

Для вызова программ в машинных кодах в Мега-Бейсике используется команда CALL\_. После этой команды записывается число, указывающее адрес вызываемой подпрограммы, а затем произвольные числа, которые передаются в машинный стек и затем используются как параметры в подпрограмме.

### FRONT-PANEL.

Фронт-панель позволяет пользователю вносить изменения в ячейки памяти и регистры микропроцессора путем записи шестнадцатеричных чисел и является по сути встроенным монитором Мега-Бейсика.

Фронт-панель активизируется при выполнении команды MON, либо после 'SPACE F' при выполнении программы.

Фронт-панель использует окно 3, в котором после активизации фронт-панели Вы увидите столбцы HEX-чисел. Слева - регистры процессора и их значения, а справа - распечатка участка ОЗУ. Звездочка указывает на текущий регистр, а инверсная полоса - текущую ячейку памяти. Все изменения производятся с помощью однобуквенных команд, после которых могут быть записаны до трех шестнадцатеричных чисел:

SPACE - возврат в Бейсик;

R nn - запись в текущий регистр 16-битового HEX-числа;

L nn nn nn - перемещение участка памяти. Здесь:

1-е число определяет начальный адрес перемещаемого блока;

2-е число задает новый начальный адрес;

3-е число определяет длину перемещаемого блока в байтах;

M nn - смещение указателя ячейки памяти на адрес nn;

S - установка точки выхода в программе для текущей ячейки;

K - продолжение выполнения программы после S;

U - восстановление точки выхода;

I nn nn nn - заполнение блока ОЗУ. Здесь:

1-е число - начальный адрес блока;

2-е число - длина блока;

3-е число - байт, которым заполняется блок;

J nn - вызов подпрограммы в машинных кодах, расположенной по адресу nn;

ENTER - смещение указателя ячейки памяти на один байт вперед;

"-" - смещение указателя ячейки памяти на один байт назад.

### Новые команды.

Ниже приводится перечень новых команд Мега-Бейсика. Здесь литера после команды определяет число или числовое выражение, а литера со знаком \$ - символьное выражение.

AUTO\_a, b - автоматическая нумерация вводимых строк. Для выключения такого режима необходимо нажать "=" и ENTER.

BACKUP - эта команда используется для копирования файлов. После ввода этой команды на экране появляется сообщение "START TAPE"

Как только на входе EAR появится сигнал, оно сменится новым сообщением "OK searching" и будет идти загрузка очередного файла с кассеты. После окончания загрузки на экране появляется сообщение:

"Start tape, then press any key".

Когда Вы будете готовы записать этот файл на кассету, нажмите любую клавишу и файл переписется. После этого система запросит Вас о еще одной копии. Если Вам она нужна, то нажмите клавишу "Y", если же нет, то нажмите "N". После этой команды необходимо выполнить команду NEW.

BRANCH\_n - по этой команде будет вызываться подпрограмма после выполнения каждой программной строки.

BROFF - выключение клавиши BREAK.

BRON - включение клавиши BREAK после команды BROFF.

CALL\_a(n,n...) - вызов программы в машинных кодах, по адресу 'a'.

После адреса в скобках могут быть записаны числа, которые записываются на машинный стек для использования в подпрограмме.

CHANGE\_a,b - это команда для изменения атрибутного файла. Здесь 'a'- маска, а 'b'- данные.

CLW\_(n),h - эта команда используется для очистки окон. Если записаны два числа, то первое определяет, какое из окон очищается. Если же записано одно число, то очищается текущее окно.

(Окончание следует)

## Профессиональный подход

Сегодня мы продолжаем публиковать заметки знаменитого английского программиста Стива Тернера, автора не менее знаменитой игры OUAZATRON с полезными советами для тех, кто делает первые шаги в освоении "Спектрума".

Итак, следующая глава:

### Орудия производства.

Для того, чтобы взять максимум возможного от микрокомпьютера, совершенно необходимо начинать программировать в машинном коде. "Спектрум" в состоянии выполнять в среднем порядка трехсот тысяч команд машинного кода в секунду. Когда же Вы программируете на Бейсике, то строки Вашей программы на самом деле ведь не исполняются компьютером. Вместо этого исполняется большая программа, называемая интерпретатором Бейсика, которая спрятана в ПЗУ. Эта программа преобразует Вашу строку в машинный код и исполняет его, поэтому для исполнения одной небольшой Бейсик-строки компьютер вынужден делать много тысяч разных операций.

Многие любители находят, что переход от программирования на Бейсике к машинным кодам представляет совершенно невообразимую по сложности и очень скучную задачу. Я называю это явление "Барьером машинного кода". В жизни существует много барьеров и они преодолимы. Как и звуковой барьер, барьер машинного кода преодолим с помощью правильной технологии и грамотной проработки проекта.

### 1. Редактор/Ассемблер.

Многие начинают освоение машинного кода с применения Бейсика для того, чтобы включить в свою программу машинно-кодovou процедуру путем применения операторов POKE и списков DATA.

Да, конечно, это самый простой способ, чтобы вставить в готовую Бейсик-программу готовую и отлаженную процедуру, но этот путь никак не годится для проработки, написания и отладки этой процедуры. Вы ведь работаете с числами, беря их из DATA и засылая их в нужные адреса POKE. Это очень неудобно, поскольку смысл этих чисел неочевиден. Гораздо удобнее применить программу Ассемблер, тогда создаваемый машинный код становится наглядным, его легче проследить и, конечно, отладить.

Более того, хорошая практика применения меток в Ассемблере сделает программу даже более читаемой, чем при программировании на Бейсике.

Сами посудите, оператор Ассемблера CALL SOUND больше говорит о своем назначении, чем строка Бейсика GO TO 9700.

Когда выбираете для себя программу Ассемблер, обратите внимание на его систему редактирования. К сожалению немногие программисты знают о хороших стандартах.

Ввод и редактирование программ должны быть такими же простыми, как печать писем в текстовом редакторе. Желательно, чтобы редактор Ассемблера позволял делать следующие операции:

- перемещение блоков;
- удаление блоков;
- копирование блоков.

Без этих функций программирование будет не только утомительным, но самое главное, подверженным многочисленным ошибкам. Я рекомендую работать с полноэкранным редактором, в котором Вы можете использовать весь экран, а редактор сам оформит подготовленный Вами текст в удобочитаемые колонки.

Управление курсором, удаление и вставка символов должны выполняться точно также, теми же клавишами, что и при обычном программировании в Бейсике, иначе Вы будете испытывать неудобство. К сожалению, многие пакеты игнорируют этот элементарный здравый смысл.

Следующей чертой, на которую надо обратить внимание - это требование, чтобы

Ассемблер после ассемблирования кода мог выполнить распечатку объектного кода в установленном формате. Стандартов в общем-то нет, но примером того, как она может выглядеть, служит процедура, приведенная в конце данной главы. (В следующем номере ZX-РЕВЮ - "ИНФОРКОМ").

Некоторые Ассемблеры распечатывают листинг до того, как рассчитаны адреса меток. Это не очень хорошо. И, что еще важно здесь учесть, это те ограничения, которые накладывает Ваше аппаратное обеспечение. Так, если Вы используете какую-то дисковую систему, то Ассемблер должен ее поддерживать.

## **2. Монитор.**

Под "монитором" здесь понимается не дисплей, а дисассемблирующая программа. Их часто называют "мониторами", т.к. под их "управлением" производятся отладочные запуски разрабатываемой Вами программы. Я знаю некоторых программистов, работающих над игровыми программами, которые не пользуются мониторами. Очень не советую так поступать. Раз уж Вы работаете с компьютером, так используйте его возможности до конца. Пусть он поможет Вам в выполнении отладки Вашей программы.

Лучше всего выбирать такой Ассемблер, который имеет родственный ему монитор, тогда Вы сможете совместно их использовать. Конечно, монитор должен иметь возможность загружаться и без Ассемблера, поскольку в паре они занимают так много места в памяти, что вам остается пространство только на некрупные процедуры. Лично я предпочитаю использовать монитор минимальных размеров.

Необходимыми функциями монитора являются:

- просмотр содержимого ячеек памяти (адресов);
- просмотр содержимого регистров процессора;
- возможность изменения содержимого ячеек памяти;
- установка точек прерывания (Breakpoint).

Точки прерывания позволяют вам смело запускать не отлаженную процедуру. Когда программа дойдет до этой точки, она остановится, передаст управление монитору, и Вы сможете проверить содержимое регистров, адресов и т.п., а потом продолжить работу. Постепенно перемещая точку прерывания, Вы сможете найти, где в программе у Вас происходит сбой и внести изменения.

Точка прерывания выполняет функции, похожие на действие оператора Бейсика STOP.

Если же у Вас нет принтера, чтобы после ассемблирования распечатать объектный код, то Монитор становится для Вас абсолютно необходимым инструментом.

## **3. Внешняя память.**

Если Вы собираетесь разрабатывать большие программы, то быстро работающее устройство для записи программы себя обязательно окупит. Когда я работал над программой QUAZATRON, то выгружал версии его десяти основных модулей более шестисот раз. А уж сколько раз я выполнял пробные запуски, об этом я и говорить не смею. Но дело не только в этом. Когда программа зависает или сбрасывается, Вам надо успеть войти в работу как можно быстрее, иначе нить рассуждений может быть потеряна и затраты времени станут еще больше. Лично у меня в среднем два раза в день компьютер "вылетает" только в связи с проблемами "питания".

Микродрайвы вполне удовлетворяют необходимым требованиям и дешевы, но лично мне больше нравится работать с дисковой системой. Хотя должен сказать, что для этого мне пришлось адаптировать под дисковод мой Ассемблер, а это была отнюдь непростая задача.

## **4. Принтер**

С каждым годом принтеры становятся все дешевле и все доступнее. Для программиста принтер нужен прежде всего, чтобы распечатать текст программы, а его возможности по исполнению качественных документов отходят на второй план. Поэтому можно выбирать самый дешевый принтер из доступных.

Конечно подключается принтер к компьютеру через интерфейс и поэтому надо либо

выбирать стандартный интерфейс и, соответственно при выборе Ассемблера надо убедиться, что он его поддерживает, либо при нестандартном интерфейсе возникнет необходимость адаптировать Ассемблер под работу с Вашим интерфейсом.

## 5. Книги по Z-80.

О машинном коде процессора Z-80 написано уже немало книг. Наилучшим справочным руководством общепризнанно считается монография Л. Левенталя "ASSEMBLER LANGUAGE PROGRAMMING". В этой книге каждой команде процессора отведена отдельная страница, приведены подробнейшие данные об условиях работы каждой команды (состояния флагов, время исполнения и т.п.). Кроме этого, книга изобилует мно-<sup>1</sup> освоение машинного кода, то Вам необходимо что-нибудь попроще. Вы можете обратиться в ближайшую библиотеку и подобрать себе необходимые книги. Вы можете также обнаружить, что фирменное руководство к "Спектруму" в своих приложениях содержит очень много полезной информации.

Изучение машинного кода рано или поздно приведет Вас к необходимости понимания организации работы самого компьютера. Вам потребуется литература по ПЗУ компьютера. Наилучшим справочником по этому вопросу может служить книга доктора Яна Логана и доктора Фрэнка О'Хары "Полный дисассемблер ПЗУ" ("THE COMPLETE SPECTRUM ROM DISASSEMBLY").

\* \* \*

### Комментарий ИНФОРКОМа

В этом вопросе Стив Тернер вплотную подошел к тому, чем занимаемся мы, и в ответ на его предложение обратиться в ближайшую библиотеку мы считаем необходимым сделать ряд замечаний.

Конечно же в ближайшей библиотеке Вы почти ничего, а скорее всего вообще ничего не найдете. Если Вы живете в Москве, то еще сможете получить в читальном зале ГПНТБ на Кузнецком Мосту русский перевод книги Веккерса, являющейся официальным фирменным руководством к "Спектруму", а может быть даже заказать его ксерокопию, но что делать если Вы живете в другом городе?

Здесь Вам поможет "ИНФОРКОМ". Мы уже писали о нашем трехтомнике для изучающих программирование в машинных кодах. Более подробную информацию можете найти в рекламном приложении на стр. 104.

Уверены, что с его помощью Вы сможете решить основные проблемы.

По вопросам, касающимся ПЗУ компьютера, мы можем Вас отослать к постоянному разделу "Секреты ПЗУ", в котором пусть не столь полно, как в книге Я.Логана и Ф.О'Хары, но зато более популярно мы рассказываем о программах, входящих в ПЗУ.

Кроме того, по индивидуальным заказам мы подготовили и уже разослали желающим копию английского издания этой книги. У нас еще осталось около 40 экземпляров и, если Вам она необходима, то можете успеть ее заказать. Поставку не гарантируем, Вам может не хватить, т.к. сроки приема заказов уже кончились, и потому это единственное, что мы высылаем наложенным платежом. Пожалуйста, не надо высылать предоплату, а только запрос. Цена - 112 руб.

Теперь относительно работ Лэнса Левенталя. Кроме упомянутой выше книги "Z-80 ASSEMBLY LANGUAGE PROGRAMMING" (600 стр.) у него есть еще логическое продолжение, написанное в соавторстве с Уинтропом Сэвиллем - "Z-80 ASSEMBLY LANGUAGE SUBROUTINES" (500 с.)

В этой книге рассматриваются многочисленные примеры создания процедур в машинных кодах.

Мы в принципе могли бы рассмотреть вопрос о том, чтобы сделать несколько копий и той и другой книги и разослать их наложенным платежом, аналогично тому, как мы это сделали с книгой Логана и О'Хары. Правда, надо иметь в виду, что при нынешних ценах на копирование и их внушительном объеме стоить это будет очень дорого, о чем мы объявим в случае наличия пожеланий.

---

<sup>1</sup> В оригинале пропущена строка (Прим. OCR)

С книгой же Виккерса все гораздо проще. На сегодняшний день она распространена очень широко и ее русский перевод можно купить на всех рынках страны. Ее распространяют многие кооперативы. Мы ее не распространяем, поскольку на русском языке мы предлагаем только то, что сделали сами или то, что сделано по нашему заказу, а к данной книге это не относится.

(Продолжение следует)

# Секреты ПЗУ

## Программы, обслуживающие работу с магнитофоном.

Этот раздел ПЗУ занимает адреса с 04C2 по 09F3. Основная задача содержащихся здесь процедур - исполнение четырех команд - SAVE, LOAD, VERIFY и MERGE. Они выполняют все необходимые для этого функции:

- генерируют пилоттон (широкие полосы на экране телевизора, появляющиеся перед загрузкой/выгрузкой каждого блока программы);

- записывают и считывают "хэдер" (17-байтный заголовок, который предшествует загрузке/выгрузке каждого программного блока (program), а также блоков "машинных кодов (bytes) и блоков данных (character array или number array);

- записывают и считывают сам программный блок.

Общим в работе всех подпрограмм, обслуживающих команды SAVE,LOAD,VERIFY и MERGE является то, что адрес блока, для обработки которого они предназначены, должен быть предварительно установлен в регистровой паре IX, а длина этого блока - в регистровой паре DE. При этом аккумулятор (регистр A) содержит 0, если обрабатывается 17-байтный "хэдер", а если обрабатывается сам программный/кодовый блок или блок данных, то в аккумуляторе находится число 255 (0FFH).

Начальная точка входа в этот раздел ПЗУ находится в адресе 0605H, после чего исполнение разветвляется, в зависимости от того, с какой командой из SAVE, LOAD, VERIFY и MERGE Вы работаете.

Основные наиболее крупные блоки этого раздела такие:

0462-053E.

Подпрограмма "SA-BYTES".

Выполняет передачу байтов, содержащихся начиная с адреса, на который указывает содержимое IX, в количестве, указанном в DE, на внешний порт магнитофона. Эта же подпрограмма управляет изменением цвета бордюра при загрузке/выгрузке и звуковым динамиком.

053F-0555.

Подпрограмма "SA/LD-RET".

Небольшая подпрограмма, общая для операций SAVE и LOAD. Выполняет необходимые финишные операции после окончания загрузки/выгрузки.

0556-0604.

Подпрограмма "LD-BYTES".

Выполняет прием байтов с внешнего входного порта магнитофона при операциях LOAD,VERIFY,MERGE. Как и для программы "SA-BYTES" в паре IX находится адрес, начиная с которого располагается загружаемый (проверяемый) блок, а в паре DE - его длина.

0605-096F.

Главная рабочая область данного раздела.

Поскольку адрес 0605 является точкой входа в пакет процедур, обслуживающих работу с магнитофоном, здесь размещается основной логический блок, который и управляет прохождением исполнения поданной Вами команды.

0970-09A0.

Подпрограмма "SAVE CONTROL".

Выполняет вспомогательные действия при выгрузке.

Осуществляет связь с пользователем, например выдает сообщение "Start tape, then press any key" и ожидает нажатия клавиши.

09A1-09F3.

Это не подпрограмма. В этих адресах хранятся сами тексты сообщений (кроме сообщений об ошибках), которые могут появиться при работе с магнитофоном.

Например, хорошо знакомое сообщение, появляющееся при начале загрузки "Program:" или "Bytes:" и др.

Поскольку процедуры, связанные с обслуживанием магнитофона имеют особую важность для пользователей, мы их рассмотрим более подробно, раскрыв логику их работы и взаимосвязь. Эти данные могут пригодиться тем, кто сам организует нестандартный ввод/вывод, занимается разработкой загрузчиков, защищающих от копирования через копировщики, а также исследует возможность снятия таких защит.

Многие так или иначе работают вопросами ускорения ввода/вывода аппаратно. Поможет эта информация и тем, кто занимается разработкой или анализом копирующих программ.

На приведенных ниже блок-схемах показана взаимосвязь между отдельными процедурами, входящими в этот раздел, а далее прокомментированы эти процедуры.

Условные обозначения:

— переход, ветвление (команда Ассемблера JP);

≡ вызов с возвратом (команда CALL или RST).

Тем, кто не знает машинного кода или языка Ассемблера Z-80 и по этой причине будут испытывать трудности при чтении материала, очень решительно советуем приобрести наш трехтомник по машинному коду для начинающих.

SAVE\_ETC: 0605-0620

Это главная точка входа в программу для всех четырех команд LOAD, SAVE, VERIFY, MERGE. Чтобы установить какая конкретно из четырех обрабатывается, в системной переменной T\_ADDR (23668 DEC) выставляется соответствующее число:

SAVE - 00;           LOAD - 01;

VERIFY - 02;       MERGE- 03.

Процедура выполняет подготовку к выгрузке или загрузке "хэдера". При работе вызывает процедуры:

- EXPT\_EXP (1C8C) - для передачи числовых параметров "хэдера" на стек калькулятора;

- SYNTAX\_Z (2530) - для проверки в каком режиме работает БЕЙСИК - в режиме исполнения команды или проверки синтаксиса. В последнем случае выполняется переход вперед на процедуру SA\_DATA.

SA\_SPACE: 0631-0636

Резервирует место в рабочей области БЕЙСИКа для размещения "хэдера", для чего обращается к процедуре BC\_SPACES(0030).

SA\_BLANK: 0639-0641

Выполняет проверку параметров имени.

Вызывает процедуру STK\_FETCH (2BF1) для ввода со стека калькулятора параметров имени.

Если имя пустое или длиннее десяти символов, что допустимо только для LOAD, VERIFY и MERGE, то переход на SA\_NULL, если с именем все в порядке - переход на SA\_NAME.

REPORT\_F: 0642-0643

Если имя неправильное, вызывается ERROR\_1 (0008) с кодом перехватчиком 0E, что означает "invalid file name".

SA\_NULL: 0644-064A

Если имя - пустое, переход на процедуру SA\_DATA. "Обрезает" имя, если оно более 10

СИМВОЛОВ.

SA\_NAME: 0648-0651

Размещает имя в рабочей области БЕЙСИКа.

SA\_DATA: 0652-0669

Определяет с каким типом объекта производится операция.

Если это не блок данных, т.е. либо экран, либо программа, либо машинный код, то переход на процедуру SA\_SCR\$ и далее.

Если это блок данных, то переход на SA\_VOLD, если имеем дело с массивом данных, имеющимся в памяти, что бывает при операциях SAVE и VERIFY, или переход на SA\_NEW, если обрабатываем новый блок данных.

Вызываемые процедуры:

- GET\_CHAR (0018) - проверяет код первого символа для идентификации "DATA" это или нет;

- REPORT\_C (1C8A) - вызывается при попытке сделать MERGE для блока данных, что конечно же невозможно. А оттуда запускается ERROR\_1 (0008) с кодом перехватчиком 0B, что означает ошибку "Nonsense in BASIC";

- NEXT\_CHAR(0020)- подготовка к приему нового символа.

- LOOK\_VARS (28B2) - в таблице переменных разыскивает массив данных.

REPORT\_2: 0670-0671

Сообщение об ошибке, если соответствующий массив данных в памяти не существует (не разыскан). - Variable not found.

SA\_V\_OLD: 0672-0684

Обрабатывается массив, существующий в памяти.

Вызываемые процедуры: SYNTAX\_Z - см. выше.

Если идет проверка синтаксиса, то переход на SA\_DATA\_1.

SA\_V\_NEW: 0685-068E

Определяется тип данных - символьные или числовые.

SA\_V\_TYPE: 068F-0691

Подготавливает для выгрузки или загрузки код типа данных в качестве первого байта "хэдера".

SA\_DATA\_1: 0692- 069F

Здесь анализируется та часть команды загрузки/выгрузки, которая следует за оператором DATA и за именем массива. Следующим символом должна идти открывающая скобка "(" . Если это не так, то возврат на SA\_V\_OLD, а оттуда на сообщение об ошибке REPORT\_C.

Вызываемые процедуры:

- NEXT\_CHAR(0020) - см. выше.

- CHECK\_END(1BEE) - обеспечивает выдачу сообщения об ошибке, если вся строка до конца не прошла успешную проверку на синтаксис.

SA-SCR: 06A0-06C2

Проверяет, является ли обрабатываемый блок экраном. Если нет, то переход на SA\_CODE и далее, при попытке сделать MERGE для экрана выдает сообщение об ошибке REPORT-C. Если все в порядке, подготавливает регистры процессора к работе с экранной областью и переходит на SA\_TYPE-3.

SA\_CODE: 06C3-06E0

Проверяет, является ли обрабатываемый блок блоком машинных кодов. Если нет, то очевидно это БЕЙСИК (больше ничего не остается) и тогда переход на SA\_LINE. При попытке сделать MERGE выполняет переход на REPORT\_C. Если идет операция SAVE, то проверяется есть ли параметры начала блока и длины. Выгружать коды только по имени нельзя. В этом случае также переход на REPORT\_C.

Вызываемые процедуры:

- NEXT\_CHAR - см. выше;
- PR\_ST\_END (2048) - проверяет закончен ли оператор. Если да, то возвращает 0 и работа продолжается, если нет, то строка не закончена. ПЗУ находится не в режиме исполнения, а в режиме проверки синтаксиса и следует переход на процедуру SA\_CODE\_1.
- USE\_ZERO (1CE5) - помещает 0 на стек калькулятора в качестве адреса начала блока. Работа продолжается переходом на SA\_CODE\_2.

SA\_CODE\_1: 06E1-06EF

Анализирует синтаксис, дает переход на REPORTS при попытке сделать SAVE без указания начального адреса и длины блока.

Вызываемые процедуры:

- EXPT\_1NUM (1C82) - процедура калькулятора. Рассчитывает значение последнего выражения. Результат становится последним вложением на вершине стека калькулятора, которое далее считается адресом начала обрабатываемого блока кодов.
- GET\_CHAR (0018) - см. выше. Продолжение работы - SA\_CODE\_3.

SA\_CODE\_2: 06F0-06F4

Посредством вызова процедуры USE\_ZERO (1CE6) помещает 0 на вершину стека калькулятора в качестве параметра длины обрабатываемого блока машинного кода. Продолжение работы - SA\_CODE\_4.

SA\_CODE\_3: 06F5-06F8

С помощью процедуры калькулятора EXPT-1NUM рассчитывает значение последнего выражения, помещает его на вершину стека и считает его длиной блока кодов.

Начиная с этого времени все параметры хранятся в той части рабочей области БЕЙСИКа, которая отведена для хранения "хэдера".

SA\_CODE\_4: 06F9-0709

С помощью процедуры калькулятора FIND\_INT2 (1E99) преобразует параметр "длина" блока из интегральной формы, применяющейся в расчетах калькулятора в двухбайтную, приемлемую для машинного кодирования. Потом то же самое для параметра "начальный адрес".

SA\_TYPE\_3: 0710-0715

Подготавливает для выгрузки или загрузки код типа данных в качестве первого байта "хэдера". В данном случае и для экрана и для блока маш. кодов этот тип - 3.

SA-LINE: 0716-0722

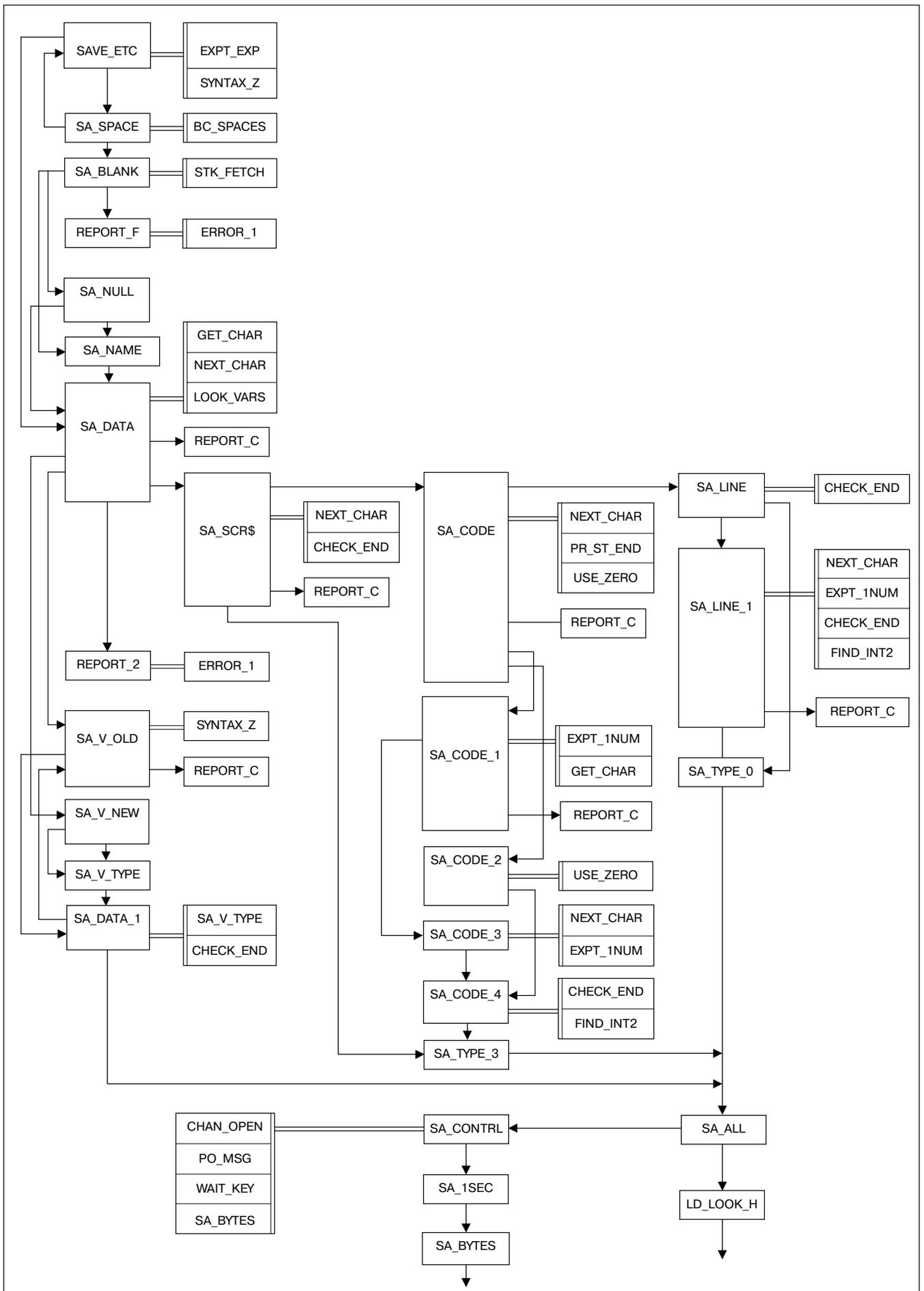
Проверяет присутствует ли оператор LINE в командной строке. Если да, то переход на SA\_LINE\_1. Туда же переход в режиме проверки синтаксиса, если по результатам вызова CHECK\_END (1BEE) установлено, что строка не завершена.

Продолжение работы - переходом на SA\_TYPE\_0.

SA\_LINE\_1: 0723-0739

Если не в режиме SAVE, то оператору LINE здесь делать нечего и выполняется переход на REPORT\_C.

Принимается число, стоящее после LINE, помещается на вершину стека калькулятора, переводится в двухбайтную форму и запоминается.



#### SA\_TYPE\_0: 073A-0759

Записывает в рабочую область БЕЙСИКа, отведенную для "хэдера" информацию о программе и ее переменных:

тип (в данном случае он равен нулю), а также содержимое системных переменных PROG и E-LINE.

Итак, в результате этих операций в рабочей области БЕЙСИКа сформировалась область информации по "хэдеру" загружаемого или выгружаемого блока. На начало этой области указывает число, находящееся в регистровой паре IX процессора, а далее информация распределена следующим образом:

IX+00 - тип (данные, коды, Бейсик).

IX+01 - IX+0A - имя (десять символов). Если имени нет, то по адресу IX+01 должно быть число FF.

IX+0B - IX+0C - количество байтов в блоке данных.

IX+0D - IX+10 - прочая информация, зависящая от типа.

#### SA\_ALL: 075A-0766

Здесь программа начинает вводить различия для разных операций. Для операции SAVE управление передается на SA\_CONTRL, а для остальных операций продолжается исполнение процедурой LD\_LOOK\_H.

Напоминаем, что различает программа о какой операции идет речь по содержимому системной переменной T-ADDR(23668-десятиричный адрес).

#### LD\_LOOK\_H: 0767-0789

Процедура выполняет поиск заголовка ("хэдера") на ленте и печать его на экране. Поиск организован в цикле, пока "хэдер" не будет найден.

Если заголовок найден, но его тип не совпадает с ожидаемым - переход на LD\_TYPE. Если совпадает, то дается команда на сравнение десяти символов имени и выполняется прямой проход на LD\_TYPE.

Вызываемые процедуры: - LD\_BYTES (0556) – выполняет загрузку встреченного заголовка. Подробнее об этой подпрограмме ниже.

CHAN\_OPEN (1601) - открывает канал "S" для печати имени на экране.

#### LD\_TYPE: 078A-07A5

Печатает на экране сообщение Program или Bytes, Number array, Character array. Печать выполняется вызовом предназначенной для этого процедуры PO\_MSG(0C0A).

Если в "хэдере" прочитан несуществующий тип - 4 и более, возврат на LD\_LOOK\_H.

Если принятое имя надо сравнить с имеющимся в компьютере, то переход на LD\_NAME, но если в памяти содержится пустое имя, то прямой проход туда же с указанием, что сравнивать их не надо.

#### LD\_NAME: 07A6-07AC

Сравнивает имена символ за символом и переходит на LD\_CH\_PR. Если имена сравнивать не надо, сразу выполняется переход туда же.

#### LD\_CH\_PR: 07AD-07CA

Печатает символ за символом, используя для этого вызываемую процедуру PRINT\_A\_1 (0010). После печати очередного символа возвращается за новым в LD\_NAME.

Если в имени оказалось менее 10 символов, то это не имя и тогда выполняется возврат в процедуру LD\_LOOK\_H для поиска на ленте другого блока.

Далее исполнение разветвляется в зависимости от того, с какой из загрузочных команд мы имеем дело.

VERIFY - VR\_CONTRL(07CB)  
LOAD - LD\_CONTRL(0808)  
MERGE - ME\_CONTROL(08B6).

VR\_CONTRL: 07CB-07E8

Процедура принимает длину загружаемого блока, сравнивает ее с заказанной и переходит на VR\_CONT\_1. Если длина не была указана при подаче команды, то сразу переходит туда же. Если длина читаемого блока не совпадает с заказанной длиной, переход на REPORT\_R (0806).

VR\_CONT\_1: 07E9-07F3

Выставляет адрес начала загружаемого (проверяемого блока) и переходит на VR\_CONT\_2. Если при подаче команды этот параметр был опущен, то принимается тот, который был прочитан в заголовке загружаемого блока и сразу выполняется переход туда же.

VR\_CONT\_2: 07F4-07FF

Принципиально верификация мало чем отличается от загрузки. Просто при верификации байт считанный с магнитфонного порта не запоминается в памяти, а только сравнивается с тем, который там находится. Поэтому значительная часть процедур, обрабатывающих режим VERIFY используется также и в режиме LOAD, если речь идет о загрузке блоков маш. кодов (Bytes) или экранов (Screen\$), но только не БЕЙСИКа.

Различие начинается здесь, в этой процедуре.

Процедура выставляет флаг C, если идет режим LOAD и снимает его, если режим VERIFY.

VR\_CONT\_3: 0800-0801

Выставляет FF в аккумуляторе (подготовительная операция перед загрузкой блока).

LD\_BLOCK: 0802-0605

Вызывает процедуру LD\_BYTES(0556), выполняющую загрузку. О ней мы поговорим ниже. Если загрузка прошла нормально (а проверяют это проверкой флага C) - "выход". Если нет, переход на REPORT\_R.

REPORT\_R: 0806-0807

При появлении ошибки вызывает процедуру ERROR\_1(0008) с кодом перехвата 1A, что означает Tape loading error.

LD\_CONTRL: 0808-0818

Отсюда начинается управление загрузкой БЕЙСИКа.

Возможны два варианта - загружается БЕЙСИК-программа или именованный массив данных.

Напомним, что управлением загрузкой прочих типов занималась ветвь программы, предназначенная для верификации - VR\_CONTRL и др.

Из "хэдера" принимается длина загружаемого массива. Если она не указана - переход на LD\_CONT\_1, а если указана - на LD\_CONT\_2.

LD\_CONT\_1: 0819-0824

Проверяет, достаточно ли в существующей рабочей области БЕЙСИКа места под загружаемую переменную или массив. Если достаточно, то переход на LD\_DATA(082E).

Если нет - продолжение работы.

LD\_CONT\_2: 0825-082D

Резервирует под переменную дополнительно 5 байтов и проверяет хватает ли на это

оперативной памяти. Проверка выполняется вызовом TEST-ROOM(1F05).

LD-DATA: 082E-084B

Здесь рассматривается загрузка массива.

Если идет загрузка БЕЙСИК-программы, то сразу переход на процедуру LD\_PROG.

Если загружается новый массив, то сразу делается переход на процедуру LD\_DATA\_1.

Если такой массив уже существует, резервируется дополнительное место под описание массива и работа продолжается.

Вызываемая процедура:

- RECLAIM\_2 (19E8) – выполняет резервирование памяти.

LD\_DATA\_1: 084E-0872

Отводит место под содержимое элементов массива в конце области переменных БЕЙСИКа.

Резервирование места в этой области выполняет вызываемая процедура MAKE\_ROOM (1655).

Далее производится сама загрузка блока данных (массива). Это выполняется переходом на процедуру LD\_BLOCK (0802), которую мы уже рассмотрели.

LD\_PROG: 0873-08AC

Отсюда начинаются операции по загрузке БЕЙСИК-программы вместе с переменными.

Сначала с помощью процедур RECLAIM\_1(19E5) и MAKE\_ROOM(1655) отводится место в памяти. Выставляются некоторые системные переменные. Если программа загружается без номера строки автостарта, то выполняется переход на LD\_PROG\_1, а если с номером, то выставляются соответствующие системные переменные и потом делается переход туда же.

LD\_PROG\_1: 08AD-08B5

"Оформляет" БЕЙСИК-программу как блок данных, выставляя соответствующие значения в регистрах процессора и переходит для загрузки на LD\_BLOCK(0802) и далее, о чем мы уже говорили.

ME\_CONTRL: 08B6-08D1

Отсюда начинается управление исполнением команды MERGE.

Работа происходит в три приема. Сначала подзагружаемые БЕЙСИК-строки загружаются как блок данных, затем производится слияние БЕЙСИК-строк, а затем сливаются новые переменные.

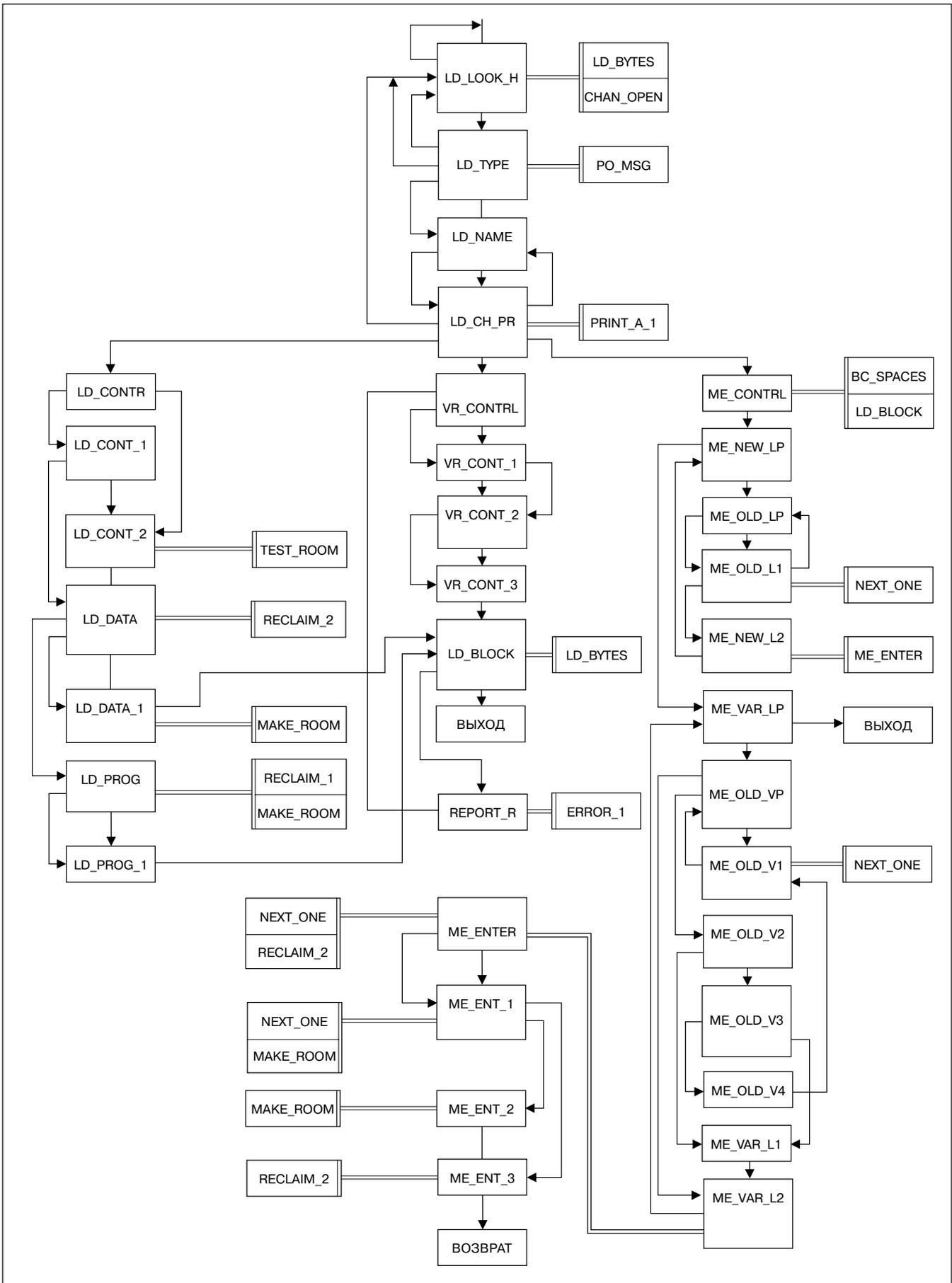
Процедура ME\_CONTRL сначала принимает параметры загружаемого блока и резервирует в рабочей области необходимое место посредством вызова BC\_SPACES(0030), затем подготавливает загрузку блока и выполняет ее вызовом LD\_BLOCK (см. выше).

ME\_NEW\_LP: 08D2-08D6

Это начало внешнего цикла, в котором один за другим вводятся номера строк новой программы и проверяются. Когда все строки просмотрены, выполняется переход на ME\_VAR\_LP(08F0).

ME\_OLD\_LP: 08D7-08DE

Это начало внутреннего цикла, в котором рассматриваются строки "старой" программы. Проверяется и сравнивается с номером "новой" строки сначала старший байт номера. Если они не совпадают, то сразу переход к следующей процедуре, в противном случае проверяется еще и младший байт.



ME\_OLD\_L1: 08DF-08EA

Когда по результатам сравнения включится флаг CARRY регистра F это означает, что подходящее место для вставки "новой" строки найдено и выполняется переход на ME\_NEW\_L2.

В противном случае отыскивается с помощью вызываемой процедуры NEXT\_ONE адрес, в котором начинается следующая строка "старой" программы и выполняется возврат к началу цикла (ME\_OLD\_LP) для рассмотрения очередной строки "старой" программы.

ME\_NEW\_L2: 08EB-08EF

Вызовом процедуры ME\_ENTER (см. ниже) выполняется вставка "новой" строки между строками "старой" программы и происходит переход в начало внешнего цикла - к процедуре ME\_NEW\_LP.

ME\_VAR\_LP: 08F0-08F8

Примерно так же, как выше происходило слияние строк, здесь начинается слияние переменных.

Это начало внешнего цикла. Сначала производится проверка на конец работы. Если все имена переменных обработаны - "ВЫХОД". В противном случае продолжение работы.

ME\_OLD\_VP: 08F9-0901

Это начало внутреннего цикла, в котором сканируется область переменных "старой" программы и просматриваются их имена.

Если найден маркер конца области, то выполняется переход на ME\_VAR\_L2.

Имена переменных старой и новой программы сравниваются сначала по первой букве. Если они совпадают, то сразу делается переход на ME\_OLD\_V2, где их рассмотрят более подробно.

ME\_OLD\_V1: 0901-0908

Если имена не совпадают, то с помощью вызываемой процедуры NEXT\_ONE (19B8) устанавливается адрес следующей "старой" переменной и происходит возврат к началу внутреннего цикла ME\_OLD\_VP.

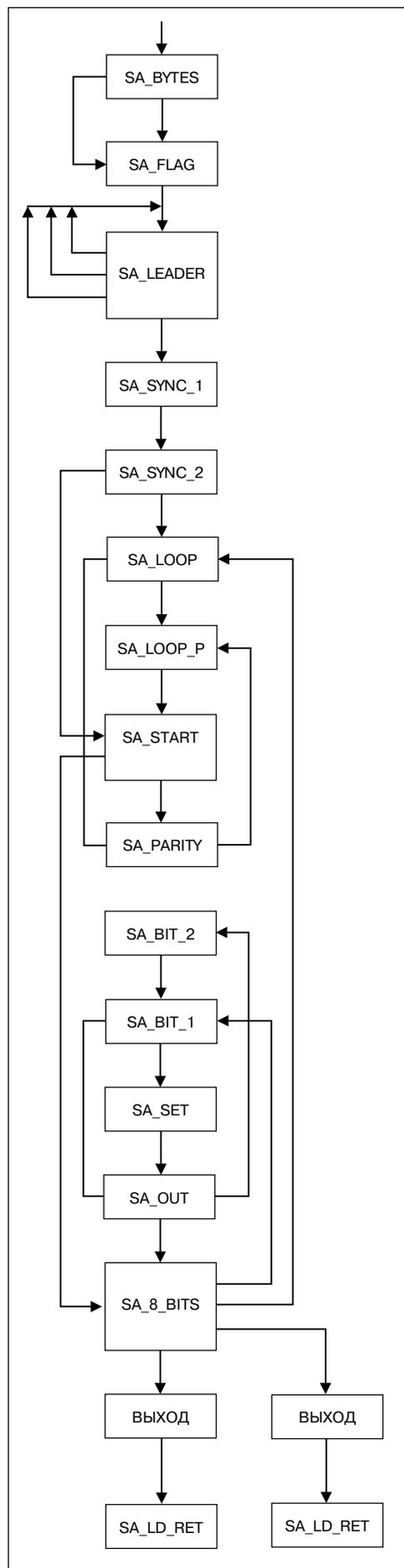
ME\_OLD\_V2: 0909-0911

Если имена состоят из одной буквы, а они совпадают, то выполняется переход на ME\_VAR\_L1, если же имя "длинное", то с переменной займется следующая процедура.

ME\_OLD\_V3: 0912-091D

Сравниваются вторые и последующие буквы имен, для чего в процедуре организован внутренний цикл. Если какая-то из букв не совпадает, то выполняется переход на ME\_OLD\_V4.

Если установлено, что имена совпали, то выполняется переход вперед на ME\_VAR\_L1.



ME\_OLD\_V4: 091E-0920

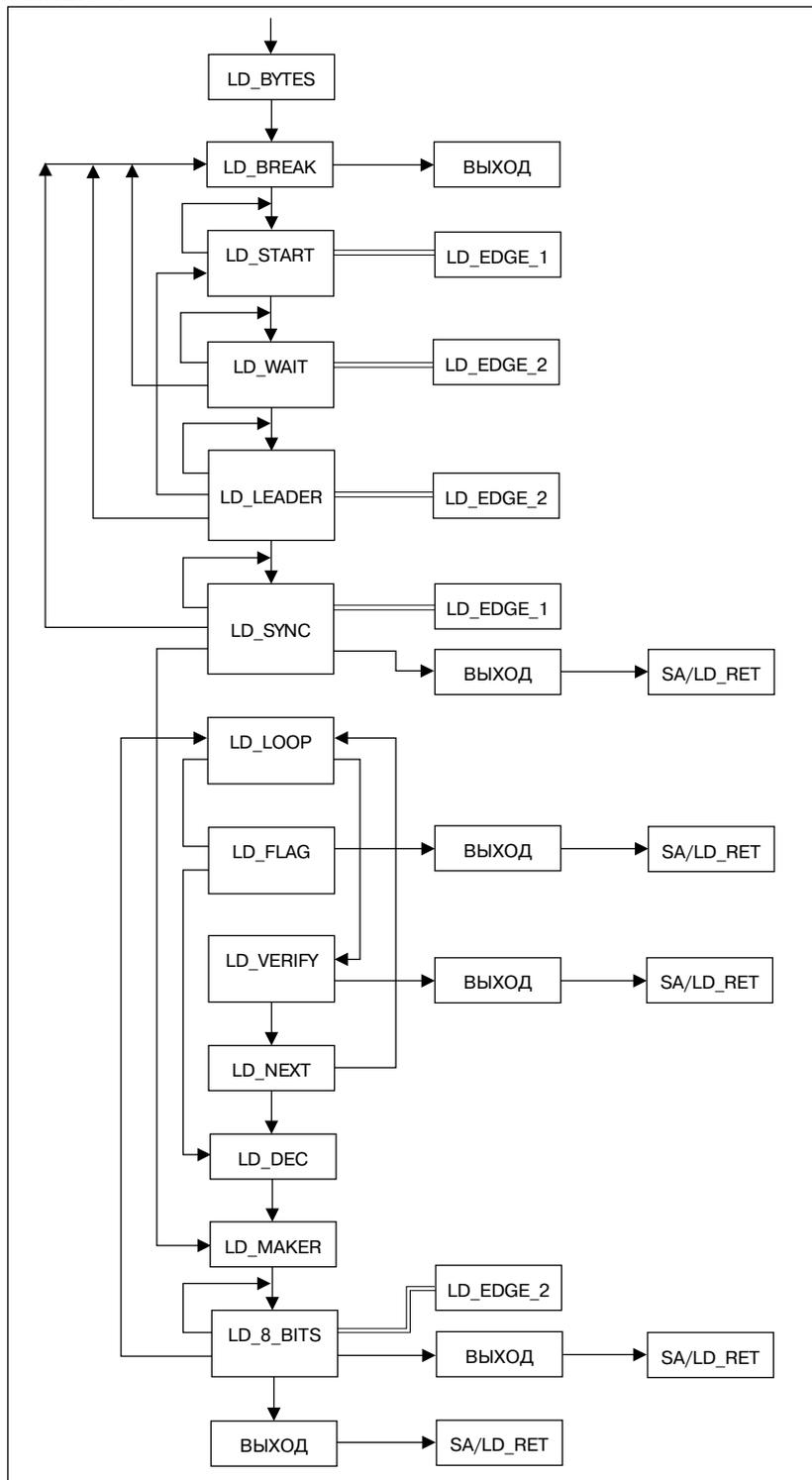
Если имена в предыдущей процедуре не совпали, то отсюда выполняется возврат в начало внутреннего цикла для выбора следующей переменной - ME\_OLD\_V1.

ME\_VAR\_L1: 0921-0922

Сюда Вы попадаете, если переменные совпали (по имени). Здесь в регистр А засылается код FF, что в дальнейшем будет означать - "заменить содержимое".

ME\_VAR\_L2: 09E3-092B

Сюда Вы попадаете, если достигнут конец области переменных и есть переменные, которым не нашлось пары. В результате регистр А содержит код + 80, что означает "должны быть добавлены переменные".



Процедура вызывает подпрограмму ME\_ENTER для добавления новой переменной и возвращается к началу внешнего цикла ME\_VAR\_LP.

#### ME\_ENTER: 09EC-093D

С этой процедуры начинается пакет, предназначенный для "сливания" строк БЕЙСИКа и программных переменных. Регистры процессора и флаги несут в себе информацию, необходимую для выполнения этих операций:

##### Флаг CARRY (C)

Выключен - слияние БЕЙСИК-строки. Включен - слияние переменной.

##### Флаг ZERO (Z)

Выключен - добавление. Включен - замена.

##### Регистровая пара HL

Указывает на адрес начала новой строки (переменной).

##### Регистровая пара DE

Указывает на адрес ее места назначения.

Процедура ME-ENTER проверяет флаг Z и если речь идет о добавлении, а не о замене, сразу передает управление на ME\_ENT\_1.

Вызываемыми процедурами NEXT\_ONE и RECLAIM\_2 (см. выше) запрашивает и заносит в регистры информацию об адресе старой переменной (строки).

#### ME\_ENT\_1: 093E-0954

Проводит подготовительные операции, например резервирует место для новой строки. Переходит на ME\_ENT\_2, если добавляется переменная и на ME\_ENT\_3, если строка.

#### ME\_ENT\_2: 0955-0957

Резервирует место для новой переменной.

#### ME\_ENT\_3: 0958-096F

Выполняет добавление/замену. Удаляет из рабочей области копии "новых" строк и переменных. Переустанавливает системную переменную PROG и заканчивает работу, выполняя возврат в вызывающую процедуру.

#### SA\_CONTRL: 0970-0990

Мы возвращаемся к операции выгрузки - SAVE. Данная процедура служит для управления процессом выгрузки.

Сначала выполняется вызов CHAN\_OPEN(1601), что открывает канал "K", который служит для вывода информации в нижние две строки экрана.

Затем вызовом PO\_MSG(0C0A) выполняется печать в этих строках системного сообщения Start tape and press any key.

Далее процедура WAIT\_KEY(15D4) организует ожидание нажатия клавиши.

И, наконец, вызовом SA\_BYTES, о чем мы еще поговорим ниже выполняется выгрузка "хэдера" на ленту. Он выгружается таким образом, что первый байт указывает на тип выгружаемого блока, а последний байт является байтом чётности.

#### SA\_1\_SEC: 0991-09A0

Выдерживает одну секунду и, после подготовки, выполняет переход на SA\_BYTES для выгрузки самого блока.

#### 09A1 - 09F3 - сообщения.

Здесь хранятся тексты системных сообщений, выдаваемых на экран во время работы с магнитофоном. Последний байт каждого сообщения инвертирован (к нему прибавлено число +80H).

09A2 - Start tape, then press any key.

09C1 - # Program:

09CB - # Number array:

09DA - # Character array:

09EC - # Bytes:

Мы здесь употребили знак # вместо символа "возврат каретки" CHR\$ 13.

Теперь мы рассмотрим вызываемые процедуры, выполняющие выгрузку байтов - SA\_BYTES и их загрузку - LD\_BYTES.

SA\_BYTES: 04C2-04CF

Сначала процедура выставляет на машинном стеке (не путать со стеком калькулятора) адрес 055F. Это означает, что по окончании операции, при выходе из нее по команде RET, будет сделан переход по этому адресу, где расположены финишные процедуры, связанные с окончанием загрузки/выгрузки SA/LD\_RET и другие.

Устанавливаются постоянные, обеспечивающие продолжительность пилоттона перед "хэдером" продолжительностью 5 сек (в HL устанавливается число 1F80). Если SA\_BYTES была вызвана для выгрузки "хэдера", то отсюда сразу следует переход на процедуру SA\_FLAG. Если для выгрузки блока, то выставляется продолжительность пилоттона примерно 2 сек (в HL число 0C96).

SA\_FLAG: 04D0-04D7

Отключает маскированное прерывание, загружает в аккумулятор 02, что означает цвет бордюра - красный и состояние динамика - включен.

SA\_LEADER: 04D8-04E9

Здесь организуются выдача импульсов, формирующих сигнал пилоттона. Это сделано введением трех циклов. Продолжительность каждого импульса (включен/выключен) - 2048 тактов процессора - первый цикл, выполненный на базе команды DJNZ. Количество фронтов импульсов определяется содержимым регистровой пары HL. После каждого прохода L уменьшается на единицу - это второй цикл на базе команды JR NZ. Если L упадет до нуля, то снимается единица из H, что вызывает появление третьей ветви возврата к началу процедуры по команде JP P. Движение по третьей ветви конечно более длинное, чем по второй, поэтому в ней вводится компенсация DEC B.

Мы специально так подробно комментируем эту очень короткую процедуру, с которой Вы теперь разберетесь даже имея минимальные навыки по машинным кодам, "прощелкав" командой PEEK содержимое указанных адресов (если нет под рукой дисассемблера). Дело в том, что создание собственного пилоттона, отличного от стандартного - излюбленный прием тех, кто делает процедуры, защищенные от копирования. Впрочем, это только один из множества приемов.

SA\_SYNC\_1: 04EA-04F1

Вырабатывается синхронизирующий импульс. На входе в процедуру стоит задержка, обеспечивающая отключение динамика в течение 667 тактов, при этом цвет бордюра - красный, затем командой OUT по порту FE выдается сигнал на динамик и переустанавливается содержимое аккумулятора для изменения цвета бордюра на голубой.

SA\_SYNC\_A: 04FE-04FD

Окончание формирования синхроимпульса. На входе в процедуру стоит задержка, обеспечивающая включенное состояние динамика в течение 735 тактов. Затем командой OUT (FE), А выключается динамик и включается голубой цвет бордюра (в аккумуляторе - 0D).

Тем, кто не знает как производится управление внешним портом FE при вводе и выводе, мы можем только порекомендовать заглянуть на стр. 89-91 нашей книги "Первые шаги в машинных кодах" (т.1 трехтомника). Там все очень популярно написано. Мы не случайно называем этот трехтомник базовым - мы будем ссылаться на него еще не раз.

Далее управление передается процедуре SA\_START.

SA\_LOOP: 04FE-0504

Процедура является вершиной цикла, связанного с выгрузкой пакета байтов.

Проверяется не достиг ли нуля счетчик выгружаемых байтов (пара DE). Если да, то перед концом работы надо выдать байт четности, для чего переход на SA\_PARITY. Если нет - продолжить работу.

SA\_LOOP\_P: 0505-0506

Фиксируется текущее значение четности.

SA\_STAST: 0507-050D

Подготавливается побитная выдача на порт магнитофона восьми битов, составляющих байт. Включается флаг переноса (CARRY), служащий маркером для 8-ми битов байта. По нему процедура SA\_8\_BITS будет распознавать, что все 8 битов отправлены. В аккумуляторе выставляется 01 (бордюр красный, динамик включен).

Управление передается процедуре SA\_8\_BITS.

SA\_PARITY: 050E-0510

Сюда Вы попадаете, если блок фактически выгружен и осталось отправить завершающий байт четности. Он вводится в регистр H и управление передается назад в SA\_LOOP\_P для последнего прохода.

SA\_BIT\_2: 0511-0513

Каждый бит выгружается после двух проходов по нижележащей петле. На первом проходе вырабатывается как бы передний фронт импульса, на втором – задний. Процедура SA\_BIT\_2 - вершина второго прохода. Здесь устанавливается желтый цвет бордюра и сигнал на отключение динамика.

SA\_BIT\_1: 0514-0519

Вершина петли первого прохода. Здесь расположен замедляющий цикл, обеспечивающий задержку на время 801 такта процессора. Далее если выгружаемый бит равен нулю, то переход на SA\_OUT.

SA\_SET: 051A-051B

Если выгружаемый бит равен единице, то здесь выполняется принудительная задержка еще на 855 тактов в дополнение к имевшимся ранее 801. Поэтому частота следования импульсов нулей в "Спектруме" примерно в два раза выше частоты единиц.

SA\_OUT: 051C-0524

Выдает на порт FE при первом проходе 01 (динамик включен, бордюр-синий), а при втором проходе 0E (динамик выключен, бордюр желтый). После первого прохода переход на SA\_BIT\_2, после второго - на SA\_8\_BITS.

SA\_8\_BITS: 0525-053B

Организует побитную выдачу восьми битов байта. Отправляет бит за битом в SA\_BIT\_1, пока не достигнут ранее установленный маркер.

Путем опроса внешнего порта клавиатуры (тоже FE) проверяет нажатие клавиши BREAK и в этом случае выполняет выход и, соответственно, переход по содержимому машинного стека на SA/LD\_RET.

Проверяет счетчик выданных байтов (DE) и, если он не достиг FFFF, обращается назад к процедуре SA\_LOOP. Обращаем внимание, что когда счетчик равен нулю, то нужен еще один проход для выдачи байта четности, отсюда и взялось число FFFF, которое в двоичной дополнительной арифметике тождественно минус единице.

SA\_DELAY: 053C-053D

Короткая задержка перед выходом.

Теперь рассмотрим работу пакета процедур, непосредственно занимающихся

загрузкой байтов.

При входе сюда аккумулятор содержит 00, если загружается "хэдер" или FF, если блок.

Флаг C включен при операциях загрузки и выключен при верификации.

LD\_BYTES: 0556-056A

Процедура вызывается как при загрузке информации из "хэдера", так и при загрузке/верификации самого блока.

Выполняет подготовительные операции, отключает маскируемое прерывание, устанавливает цвет бордюра белым (0F на порт FE). Устанавливает на машинный стек адрес возврата 053F (SA/LD\_RET), считывает байт, поступивший с порта FE, но рассматривает только бит EAR (бит 6) путем ротации RRA и маскирования AND 20, подготавливает переключение бордюра на красный, включает флаг нуля (Z).

LD\_BREAK: 056B

Проверяет факт нажатия клавиши BREAK по состоянию флага Z. Если он выключен, выход на SA/LD\_RET.

LD\_START: 056c-0573

Вызывает процедуру LD\_EDGE\_1 для поиска фронта сигнала. Если в течение заданного времени он не отыскивается (свидетельствует об этом выключенный флаг переноса), то выполняется возврат на LD\_BREAK. Если найден, бордюр становится голубым.

LD\_WAIT: 0574-057F

Чтобы убедиться, что сигнал не случайный, здесь выполняется задержка примерно на 1 сек. И повторяется попытка найти фронт вызовом LD\_EDGE\_2. Если его нет, то возврат на LD\_BREAK.

LD\_LEADER: 0580-058E

Вызывает LD\_EDGE\_2 для поиска фронта сигнала. Если он не поступил в отведенное время, возврат на LD\_BREAK. Если поступил, то по содержимому регистра В определяется время, прошедшее с момента приема предыдущего фронта. Если оно больше 3000 тактов, то выполняется возврат на LD\_START.

В регистре Н подсчитывается количество пар фронтов. Только когда их насчитывается 256 пилоттон считается принятым и - работа может быть продолжена.

LD\_SYNC: 058F-05A8

За пилоттоном идут две части синхроимпульса. Любой принятый фронт считается за передний фронт импульса "on", но только если в заданном интервале времени поступит еще один фронт, он будет принят за задний, иначе возврат на LD\_BREAK.

Теперь ожидается импульс "off". Если он не поступил - выход на SA/LD\_RET с выключенным флагом переноса (ошибка ввода).

Теперь здесь перед загрузкой "хэдера" или блока данных подготавливаются регистры к приему первого байта, свидетельствующего о типе данных и последнего байта (байта четности). Работа продолжается переходом на LD\_MARKER.

LD\_LOOP: 05A9-05B2

Вершина цикла загрузки байтов. Отсюда управление распределяется в следующем порядке:

При загрузке первого (флагового) байта - на LD\_FLAG.

При верификации - на LD\_VERIFY.

При загрузке последующих байтов - на LD\_NEXT.

LD\_FLAG: 05B3-05BC

Здесь проверяется первый загруженный байт, определяющий тип загружаемых

данных. Если он не совпадает с тем, что было указано в команде, выполняется выход на SA/LD\_RET. В противном случае - продолжение работы - переход на LD\_DEC.

LD\_VERIFY: 05BD-05C1

В аккумулятор вводится текущий байт из памяти. В регистре L - байт, принятый с ленты. Если они не совпадают - выход (SA/LD\_RET).

LD\_NEXT:05C2-05C3

Переход к очередной ячейке памяти.

LD\_DEC: 05C4-05C7

Уменьшает на единицу счетчик загружаемых, байтов, вводит в регистр В постоянную времени, сравнением с которой впоследствии определяют что было загружено - ноль или единица.

LD\_MARKER: 05C8-05C9

Очищает регистр L, в котором будет "собираться" из битов загружаемый байт и включает в нем младший бит. После приема 8-ми битов он смещаясь влево отойдет в позицию флага С и это будет свидетельствовать о том, что байт принят полностью. Таким образом он выполняет роль маркера.

LD\_S\_BITS: 05CA-05E2

Организуется цикл по загрузке 8-ми битов от младшего до старшего. Цикл заканчивается, когда включится флаг С.

Загрузка битов выполняется вызовом LD\_EDGE\_2. Там "ловится" пара фронтов, замеряется интервал между ними. Сравняется с величиной, установленной в В. Если больше, то принята единица, а если меньше, то ноль.

В регистре Н фиксируется текущее значение четности.

По содержимому DE определяется весь ли блок обработан и если да, то выход на SA/LD\_RET, а если нет, то возврат в вершину цикла - в процедуру LD\_LOOP.

Далее следуют процедуры, занимающиеся отысканием на ленте фронтов импульсов и определением временного интервала между ними, по которому и судят о том, что это - ноль, единица, пилоттон или вообще случайный импульс.

LD\_EDGE\_2: 05E3 - 05E6

Служит для отыскания пары фронтов, первого путем вызова LD\_EDGE\_1, а второго путём прямого прохода туда же.

LD\_EDGE\_1: 05E7-05E8

Устанавливает в аккумуляторе величину первичной задержки.

LD\_DELAY: 05E9-05EC

Выполняет эту задержку в течение 358 тактов перед входом в сравнивающую процедуру.

LD\_SAMPLE: 05ED-0604

Организуется цикл, в котором с порта FE "ловится" фронт импульса. Регистр В выполняет роль счетчика проходов (таймера). Если он достиг нуля, а фронта нет - это означает "время истекло" и выполняется возврат с включенным флагом нуля (Z). Одновременно по сигналу с этого порта определяет факт нажатия BREAK-клавиши и выходит с выключенными флагами переноса (C) и нуля, если она нажата.

Если в отведенное время фронт найден, то меняет цвет бордюра и состояние звуковой динамика, и производится возврат в вызывающую процедуру.

Нам осталось только рассмотреть несколько финишных процедур, занимающихся оформлением окончания процессов загрузки/ выгрузки/ верификации/ слияния.

SA/LD\_RET:053F-0551

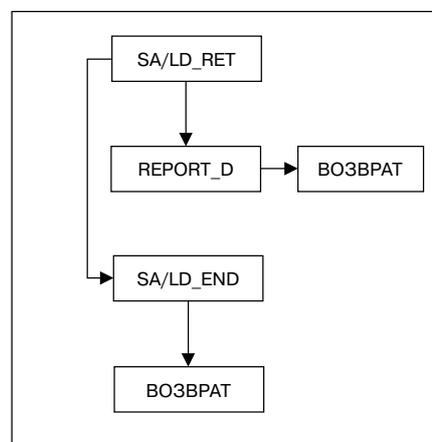
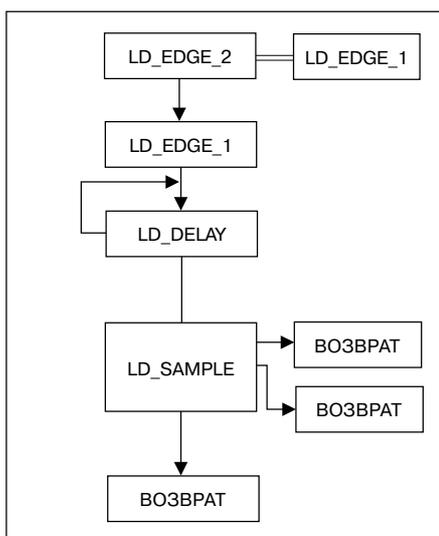
Восстанавливает исходный цвет бордюра, в последний раз проверяет нажатие клавиши BREAK, разрешает маскируемое прерывание. Если BREAK не нажата, о чем свидетельствует включенный флаг C, то переходит на SA/LD\_END.

REPORTED: 0552-0553

Если нажата BREAK-клавиша, вызывается процедура обработки ошибок ERROR\_1 (0008) с кодом перехватчиком 0C, что означает выдачу сообщения: BREAK-CONT repeats.

SA/LD\_END: 0554-0555

Восстанавливает со стека флаги и возвращает в вызывающую программу.



## FORUM

Мы продолжаем начатый разговор по письмам читателей о пределах совместимости отечественных модификаций "Спектрумов" с фирменным программным обеспечением. Пусть не сразу, пусть постепенно, но надеемся с Вашей активной помощью внести какую-то ясность в этот вопрос.

Тов. Змерзлюк из г. Краснокаменска сообщает, что на его компьютере модели "Дубна-48" не работают программы TF-COPY, COPY-86/M, LERM-7, а также программы, имеющие нестандартный загрузчик типа BOMB JACK, BOULDER DASH и др. Он был бы очень рад узнать о причине этого явления.

Завод "ТЕНЗОР" на запрос товарища Клейменова по этому поводу любезно сообщил, что ПК "Дубна-48" может не работать с теми программами, которые выполняют проверку содержимого ПЗУ.

Что по этому поводу можно сказать? В своей книге "Первые шаги в машинном коде Z-80" мы высказали отношение к доработкам ПЗУ и предложили выполнять их посредством замещающего, "теневого" ПЗУ. В рекомендациях мы основывались на многолетнем опыте, накопленном в Европе при работе с этим и другими подобными компьютерами. Сэр Синклер чего только не модифицировал в своих машинах, но вот ПЗУ - не трогал, хотя хорошо знал о том, сколько в нем ошибок. Да и фирма "АМСТРАД", перекупив у него все права на производство Синклер-совместимых моделей, тем не менее, даже начав выпускать модели 128K, обеспечила возможность работы в режиме 48K почти со 100%-ной совместимостью. Фирма "Таймекс" в своей модели 2048, серьезно переделав архитектуру компьютера, не тронула ПЗУ, а к модели 2068, которая была исполнена с расчетом на работу в дисковой системе под управлением CP/M, что потребовало полной замены ПЗУ, прилагает специальный кэтриндж Синклер-совместимости, который вставляется в специальный отсек. Есть непроверенные сведения, что так же поступили и с компьютером UNIPOLBRIT в Польше.

Мы предлагаем всем, кто столкнулся с подобными проблемами и умеет читать машинный код фирменных программ обратить усилия на то, чтобы вскрыть в них места, в которых они выполняют проверку ПЗУ и организовать обход, по возможности несложный. Мы же с радостью напечатаем сообщения о ваших открытиях.

Конечно же проблема эта относится совсем не только к "Дубне", а и к десяткам других моделей. Нужна координация усилий и очень своевременной выглядит идея научно-производственной фирмы "Икар" о создании Всесоюзной ассоциации производителей аппаратного, программного и информационного обеспечения компьютеров, совместимых с "ZX Spectrum" (ВАПРОС).

Не работает "BOMB JACK" на версии компьютера "АИОА" (Козлов, Сургут).

Товарищ Смирнов из Ленинграда пишет о том, что у него на компьютере версии BALTIC не идет программа "ELITE", т.е. она стартует, но через три-четыре минуты "зависает". То же проявляется и на других, проверенных им компьютерах этой версии. Кто что-то знает по этому вопросу, отзовитесь!

Поступили сведения о том, что то же происходит и с программой "THE LAST WORD TWO" после проведения в ней русификации. Она стартует, после набора нескольких символов зависает. Поскольку сведения поступили примерно от пяти человек из разных мест (Витебск, Мурманск и др.) и совершенно совпадают, можно предположить, что это не случайность. С другой стороны, есть уже сотни сообщений о том, что русификация прошла успешно. По-видимому, сказывается какая-то особенность применяемого компьютера. Делать выводы пока рано. Не хватает статистики, ждем Ваших наблюдений.

Интересная информация поступила от товарища Ермакова о появлении модификации компьютера "Синклер-90". Судя по приведенной им информации, ПЗУ компьютера настолько развито, что обеспечивает очень мощный мониторинг исполняемых программ и даже доступ к регистрам процессора.

Нам об этом ничего не известно, но если кто-то из авторов разработки читает эти строки, то мы надеемся на то, что он сочтет возможным дать на страницах "ZX-РЕВЮ" несколько уроков по работе с новой системой.

Крайне интересный вопрос задал в своем письме товарищ Максимов из г. Кемерово. Он посоветовал на то, что нет единого стандарта в присвоении букв русского алфавита клавишам "Спектрума". Действительно, вопрос весьма важен, т.к. от него может зависеть совместимость разрабатываемого Вами программного обеспечения.

Мы решали этот вопрос, когда несколько лет назад выполняли русификацию мощного текстового редактора "THE LAST WORD TWO". Примененный там подход мы изложили в разработке, посвященной этой программе, но поскольку не все ее имеют, попробуем осветить его здесь.

Во-первых, Вы конечно знаете, что русификацию можно проводить двумя основными методами. Первый основан на использовании символов графики пользователя. Эти символы имеют номера со 144-го по 164-ый включительно. Если Вы с их помощью набираете русский текст, то вводите его в графическом режиме (курсор G). Понятно, что 21 символа Вам недостаточно, чтобы задать все буквы русского алфавита (хотя конечно есть способы создания и переключения многочисленных банков UDG, о чем мы еще в свое время поговорим на наших страницах), поэтому приходится использовать и те английские буквы, которые имеют с русскими одинаковое написание, да и вообще ограничиваться русификацией только прописных букв. Стандартов здесь нет и быть не может, поскольку Вы и так ограничены в своем выборе. Проводите замену как вам удобно. Хотя должны сказать, что на заре Синклер-машин в СССР, где-то году в 84-м или 85-м появилась нехитрая программа ABC (неизвестного автора). Мы приводили ее в нашей разработке "Большие возможности Вашего Спектрума". Поскольку за столько лет эта программа стала общераспространенной, то может быть имеет смысл использовать заданные в ней соотношения:

"Q" - "Ы"	"D" - "Д"
"E" - "Э"	"F" - "Ф"
"R" - "Ь"	"G" - "Г"
"T" - "У"	"H" - "Ч"
"U" - "Ю"	"J" - "Ж"
"I" - "И"	"L" - "Л"
"O" - "У"	"C" - "Ц"
"P" - "П"	"B" - "Б"
"A" - "Я"	"N" - "Ш"
"S" - "Й"	"M" - "Щ"

Совсем другое дело, если Вы проводите русификацию полной заменой знакогенератора, здесь мы уже вторгаемся в поле действия международного стандарта ASCII и для начала маленькая историческая справка.

Набор символов компьютера содержит 256 символов, начиная с нулевого и до 255-го. Стандарт ASCII определяет символы с нулевого по 31-ый как управляющие коды и с 32-го по 127-ой как буквы английского алфавита, цифры и знаки препинания. Символы со 128-го по 255-ый этот стандарт не определяет, разрешая размещать в них графические символы или национальные шрифты. Здесь бы и разместить русский шрифт, как это делается, например в IBM-совместимых и других компьютерах, но для "Спектрума" из этого ничего не выйдет. Дело в том, что К. Синклер воспользовался предоставленной ему здесь свободой и уже разместил начиная со 128-го по 143-ий символы блочной графики, со 144-го по 164-ый отдал Вам в виде символов UDG, а со 165-го по 255-ый использовал для определения токенов ключевых слов, что дает Вам возможность набирать такие слова как LOAD одним нажатием клавиши. Так что в "Спектруме" LOAD - это не слово, а всего лишь один символ (239) и потому это токен, так же и SAVE - это символ 248. Подробности найдете в нашем "Справочнике по программированию в машинных кодах, Том 3 из трехтомника".

Вывод один, необходимо производить переключение шрифтов с английского на русский и наоборот. Неважно каким способом это сделано - программно или аппаратно введением дополнительного регистра, но если и русский и английский шрифты занимают одни и те же символы с 32-го по 127-ой, то на наш взгляд лучше всего делать замену так,

чтобы используемый Вами принтер тоже мог бы легко переключаться одной командой.

Наиболее распространенными для работы с 8-разрядными компьютерами в СССР являются принтеры ROBOTRON (ГДР) и D-100 (Польша). И в том и в другом реализован стандарт КОИ-7.

Для них соответствие клавиш должно быть таким (мы начинаем с 64-го кода, поскольку цифры и знаки препинания совпадают):

Код	Англ	Рус	Код	Англ	Рус
64	@	ю	96	фунт	Ю
65	A	а	97	a	А
66	B	б	98	b	Б
67	C	ц	99	c	Ц
68	D	д	100	d	Д
69	E	е	101	e	Е
70	F	ф	102	f	Ф
71	G	г	103	g	Г
72	H	х	104	h	Х
73	I	и	105	i	И
74	J	й	106	j	Й
75	K	к	107	k	К
76	L	л	106	l	Л
77	M	м	109	m	М
78	N	н	110	n	Н
79	O	о	111	o	О
80	P	п	112	p	П
81	Q	я	113	q	Я
82	R	р	114	r	Р
83	S	с	115	s	С
84	T	т	116	t	Т
85	U	у	117	u	У
86	V	ж	118	v	Ж
87	W	в	119	w	В
88	X	ь	120	x	Ь
89	Y	ы	121	y	Ы
90	Z	з	122	z	З
91	[	ш	123	}	Ш
92	\	э	124		Э
93	]	щ	125	{	Щ
94	↑	ч	126	~	Ч
95	-	ъ	127		"копирайт"

Если предположить, что кому то из наших ассов удалось или удастся так переработать ПЗУ "Спектрума", что токены будут сделаны "прозрачными", т.е. они как бы есть, но для пользователя их как бы и нет, и удастся привязать русские литеры к верхней половине таблицы кодов ASCII, то тогда имеет смысл привязывать русские буквы к клавишам так, как это сделано на стандартной пишущей машинке.

Английский шрифт - "QWERTY..."

Русский шрифт - "ЙЦУКЕН..."

В этом случае можно было бы без перешивки ПЗУ или без программной перекодировки использовать принтеры EPSON-совместимые или IBM-совместимые, поскольку в них национальные шрифты лежат в верхней половине таблицы кодов.

Товарищ Максимов высказал еще пожелание внести ясность в вопрос стандартизации джойстиков, и он абсолютно прав, вопрос назрел, но поскольку "ИНФОРКОМ" аппаратными вопросами не занимается, то мы предпочли бы выслушать мнение специалистов, а потом его довести.

В одном из прошлых выпусков "ZX-РЕВЮ" мы сообщили, что нашим читателям нужны рекомендации по русификации программы "MASTERFILE 09". Семя упало в благодарную почву и несколько человек откликнулись. Спасибо им за это.

Должны сказать, что русификация программ бывает полной и неполной. При неполной встраивается русский шрифт и появляется возможность вводить информацию на русском языке, но программные сообщения, меню, запросы и т.п. продолжают печататься хоть и русскими буквами, но на английском языке. Это наиболее быстрый способ и к возникающему при этом неудобству активный пользователь быстро привыкает и перестает его замечать.

При полной русификации заменяются и все программные сообщения. Высшим шагом является такая переделка программы, когда после русификации работают правильно и такие режимы как сортировка по новому алфавиту.

В нашей почте есть несколько писем с идеями неполной русификации. Мы их сегодня дадим в качестве "неотложной помощи", есть также предложения о разработке статьи с рекомендациями по полной русификации. С этими авторами мы выходим на контакт и будем готовить материал для последующих выпусков.

Товарищ Козлов из г. Сургут любезно поделился собственным опытом русификации МАСТЕРФАЙЛа.

1. С помощью графического редактора "ART/STUDIO" он сформировал набор русских знаков и выгрузил его на ленту (768 байтов).

2. Загрузил его командой:

```
CLEAR 56559:LOAD""CODE 56560,768
```

3. Загрузил MASTERFILE 09:

```
LOAD "MF 09 LEER"
```

4. После загрузки MF вошел в БЕЙСИК-область из главного меню.

5. Работая в БЕЙСИКе ввел новые строки в программу "MF 09 LEER":

Строка 2:

```
POKE VAL "23606",VAL"0":
```

```
POKE VAL "23607",VAL"60":
```

```
GO TO VAL "1"
```

Здесь указатель набора символов (системная переменная CHARS) устанавливает набор знаков на "Спектрумовский" и выполняется возврат в "MF".

Строка 3:

```
POKE VAL "23606",VAL"240":
```

```
POKE VAL "23607",VAL"219":
```

```
GO TO VAL "1"
```

Здесь CHARS устанавливается на новый набор знаков и выполняется возврат в MF. Строка 4030.

Здесь необходимо поменять операнды во втором операторе SAVE.

```
4030 ... SAVE "MFcode" CODE VAL "56560",VAL"8976".
```

6. Выгружается измененная программа на ленту с помощью<sup>1</sup>

Перед загрузкой программы для работы теперь надо давать прямую команду CLEAR 56559.

Адаптированная таким образом программа позволяет формировать файлы либо русским, либо латинским шрифтом, но не одновременно.

Работать с программой просто. После загрузки Вы выходите из Главного Меню в БЕЙСИК, а там даете прямую команду GO TO 2. Если работать будете с латинским шрифтом или GO TO 3, если с русским.

Примерно так же, используя БЕЙСИК-область произвел изменения и т. Соболев из Москвы. Поскольку он ввел русский знакогенератор через операторы READ и DATA, то его прием может пригодиться тем, у кого нет графического редактора "ARTSTUDIO", хотя конечно работать с ним приятнее, чем набивать 96 строк DATA, и его надо иметь, тем более

---

<sup>1</sup> В оригинале пропущена строка (Прим. OCR)

что генерация шрифта это всего лишь один процент его богатейших возможностей.

Вот что пишет товарищ Соболев. Он поставил задачу ввести русский шрифт не вторгаясь в машиннокодую часть программы и использовал для этого возможность МАСТЕРФАЙЛа исполнять БЕЙСИК-программу пользователем нажатием клавиши V в главном меню.

Сначала надо загрузить первый блок программы (mf lader) и прервать дальнейшую загрузку нажатием BREAK. После этого оператором CLEAR изменяется установка системной переменной RAMTOP: 10 CLEAR 56319: LOAD "" Сохраняется эта часть на ленте командой SAVE "mf lader" LINE 10. Запускаем измененный блок командой RUN. Начинает загружаться следующий блок MF 09 LEER. После его загрузки также прерываем дальнейшую работу командой BREAK и изменяем следующие строки программы:

```
4035 LOAD ""CODE: LOAD""CODE: GO TO VAL "1"
9000 INPUT "(RUS-LAT)-->(R/L?)"; n$: IF n$="r" OR n$="R" THEN POKE 23607,219: GO TO 9050
9010 IF n$="l" OR n$="L" THEN POKE 23607,60: GO TO 9050
9020 GO TO 9000
9050 GO TO 1
```

Измененную программу записываем на ленту:

```
SAVE "MF09LEER" LINE 4035
```

Теперь создадим новый знакогенератор с помощью приведенной ниже программы:

```
1 CLEAR 56319
5 LET A=10: LET B=11: LET C=12
10 LET D=13: LET E=14: LET F=15
12 LET x=56320
15 READ a$
20 FOR n=1 TO 15 STEP 2 30 LET W = VAL a$(n)
40 LET w=w*16: LET w=w + VAL a$(n+1)
50 POKE x,w: LET x=x+1
60 NEXT n
70 GO TO 15
```

(Пробелы в строках DATA даны для удобочитаемости. Вам их давать при наборе не надо).

100 DATA "0000 0000 0000 0000"	131 DATA "3C66 060C 1800 1800"
101 DATA "1818 1818 1800 1800"	132 DATA "0000 CEDB FBDB CE00"
102 DATA "6C6C 0000 0000 0000"	133 DATA "0000 3C06 3E66 3F00"
103 DATA "6C6C FE6C FE6C 6C00"	134 DATA "043C 607C 6666 3C00"
104 DATA "187E E07C 0EFC 3800"	135 DATA "0000 6666 6666 7F03"
105 DATA "00C6 CC18 3066 C600"	136 DATA "0000 7E66 6666 FFC3"
106 DATA "386C 3877 DECE 7B00"	137 DATA "0000 3C66 7E60 3E00"
107 DATA "1818 3000 0000 0000"	138 DATA "0018 7EDB DBDB 7E18"
108 DATA "000C 1818 1818 0C00"	139 DATA "0000 7C50 6060 6000"
109 DATA "0030 1818 1818 3000"	140 DATA "0000 C66C 386C C600"
110 DATA "0066 3CFF 3C66 0000"	141 DATA "0000 6666 6E76 6600"
111 DATA "0018 187E 1818 0000"	142 DATA "1300 6666 6E76 6600"
112 DATA "0000 0000 1818 3000"	143 DATA "0000 666C 7866 5600"
113 DATA "0000 007E 0000 0000"	144 DATA "0000 0E1E 3666 C600"
114 DATA "0000 0000 0018 1800"	145 DATA "0000 C6EE D6C6 C600"
115 DATA "0306 0C18 3060 C000"	146 DATA "0000 6666 7E66 6600"
116 DATA "7CC6 CED6 E6C6 7C00"	147 DATA "0000 3C66 6666 3C00"
117 DATA "1838 1818 1818 3C00"	148 DATA "0000 7E66 6666 6600"
118 DATA "3C66 061C 3066 7E00"	149 DATA "0000 3E66 3E36 6600"
119 DATA "3C66 061C 0666 3C00"	150 DATA "0000 7C66 7C60 6000"
120 DATA "0E1E 3666 7F06 0F00"	151 DATA "0000 3C66 6066 3C00"
121 DATA "7E60 7C06 0666 3C00"	152 DATA "0000 7E16 1816 1800"
122 DATA "1C30 607C 6666 3C00"	153 DATA "0000 6666 663E 063C"
123 DATA "7E66 060C 1618 1800"	154 DATA "0000 DB7E 187E DB00"
124 DATA "3C66 663C 6666 3C00"	155 DATA "0000 7C66 7C63 7E00"
125 DATA "3C66 663E 060C 3600"	156 DATA "0000 6060 7C66 7C00"
126 DATA "0018 1800 1618 0000"	157 DATA "0000 C6C6 F6DA F600"
127 DATA "0018 1800 1818 3000"	158 DATA "0000 3C66 0C66 3C00"
128 DATA "0C18 3060 3018 0C00"	159 DATA "0000 D6D6 D6D6 FE00"
129 DATA "0000 7E00 007E 0000"	160 DATA "0000 7C06 3E06 7C00"
130 DATA "6030 180C 1830 6000"	161 DATA "0000 D6D6 D6D6 FF03"

162 DATA "0000 6666 3E06 0600"	179 DATA "7CC6 C6C6 C6C6 7C00"
163 DATA "0000 0000 0000 000F"	180 DATA "FEC6 C6C6 C6C6 C600"
164 DATA "CEDB DBFB DBDE CE00"	181 DATA "7EC6 C67E 3666 C600"
165 DATA "0E1E 3666 FEC6 C600"	182 DATA "FCC6 C6FC C0C0 C000"
166 DATA "FCC0 C0FC C6C6 FC00"	183 DATA "7CC6 C0C0 C0C6 7C00"
167 DATA "CCCC CCCC CCCC FC06"	184 DATA "FC30 3030 3030 3000"
168 DATA "7E65 6666 6666 FFC3"	185 DATA "C6C6 C67E 06C6 7C00"
169 DATA "FCC0 C0F8 C0C0 FE00"	186 DATA "DBDB 7E18 7EDB DB00"
170 DATA "187E DBDB DB7E 1800"	187 DATA "F8CC CCFC C6C6 FC00"
171 DATA "7C60 6060 6060 6000"	188 DATA "C0C0 C0FC C6C6 FC00"
172 DATA "C356 3C18 3C66 C300"	189 DATA "C6C6 C6F6 DADA F600"
173 DATA "C6C6 CEDE F6E6 C600"	190 DATA "3C66 061C 0666 2C00"
174 DATA "D6C6 CEDE F6E6 C600"	191 DATA "D6D6 D6D6 D6D6 FE00"
175 DATA "C6CC D8F8 CCC6 C600"	192 DATA "7CC6 061E 06C6 7C00"
176 DATA "0E1E 3666 C6C6 C600"	193 DATA "D6D6 D6D6 D6D6 FF03"
177 DATA "C6EE FED6 C6C6 C600"	194 DATA "6666 663E 0606 0600"
176 DATA "C6C6 C6FE C6C6 C600"	195 DATA "3C42 99A1 A199 423C"

После запуска программы командой RUN компьютер сформирует новый знакогенератор и разместит его начиная с адреса 56320.

Выгрузите его на ленту прямой командой SAVE "RUS"CODE 56320,768

В заключение необходимо с помощью какой-либо копирующей программы собрать полученные блоки программы MASTERFILE09 в следующей последовательности: "mf lader" "MF 09 LEER" "RUS" "MF mcode"

Переключение на русский шрифт или наоборот выполняется выбором опции V в главном меню. В ответ на запрос "R/L?" Вам надо нажать клавишу R или L, в зависимости от того, какой шрифт Вам нужен.

Теперь небольшой финансовый вопрос. Многие читатели заметили в мартовском выпуске, что цена на подписку увеличилась на 20 руб. и прислали вопросы - надо ли производить доплату? А некоторые даже и перевели деньги!

Уважаемые товарищи! Если Вы подписались, то ничего вам доплачивать не нужно, что бы мы с ценой ни делали. Это имеет отношение только к новым подписчикам. Мы и дальше может быть будем отслеживать конъюнктуру рынка, но Вы страдать не должны. Единственное, когда Вы можете произвести доплату - это если хотите расширить условия подписки. Например подписавшись на игровое приложение Вы хотели бы получать выпуски целиком - вот тогда пожалуйста, но и то это дело добровольное. Мы выслали первые три выпуска всем в полном составе, независимо от того, кто на какой раздел подписался и дали возможность их оценить.

Теперь у нас будет головная боль как вернуть деньги тем людям, кто прислал по 20 рублей. Может быть Вы учтете их, если будете заказывать что-либо новое, хотя должны предупредить, что до сентября месяца мы новых изданий не планируем. Мы их готовим, но консервируем.

В заключение мы благодарим всех читателей, приславших письма для "ФОРУМа" или по выступлениям "ФОРУМа" и приносим извинения, что очень многим мы не в состоянии ответить, т.к. писем очень много. Но читаем и принимаем к сведению абсолютно все.

До встречи в июне!

ИНФОРКОМ.

## Маленькие хитрости

Продолжение. Начало на с. 51.

Сегодня мы продолжаем разговор; о простейших приемах, позволяющих экономить объем памяти, необходимой для написания вашей программы. Если Вы даже и не пишете еще<sup>1</sup>

Еще большую экономию при работе с числами и другими типами переменных можно получить, если Вы знаете заранее, что изменяться в программе эти переменные не будут или если Вам эти изменения безразличны.

Когда Вы запускаете программу (RUN), то все переменные, которые есть в программе, копируются дополнительно в область переменных (VARS), т.е. они хранятся дважды. Почему бы их не хранить только в области переменных, конечно после того, как программа Вами отлажена и работает. Сделать это можно так. Попробуйте дать прямую команду

```
LET a=10
```

Прямая команда вводится без номера строки и выполняется компьютером немедленно. Начиная с этого момента компьютер "помнит", что а равно десяти, хотя такой строки в программе и нет. Проверьте его командой PRINT а. Убедились? Помнит компьютер это число потому, что значение а хранится в области переменных. Посмотрите, как оно там хранится:

```
10 LET p = PEEK 23627+ 256* PEEK 23628: FOR f=p TO p+17: PRINT f,PEEK f : NEXT f
```

Вот только запускать эту строку командой RUN уже нельзя потому, что RUN перед началом работы удаляет из памяти все переменные и очищает стек. Общее правило - если Вы применяете переменные, которые хранятся в памяти, но не хранятся в программе, т.е. не выставляются программой, то вместо RUN можно использовать только GO TO, в данном случае запуск произведете командой GO TO 10.

Второе золотое правило - всегда выгружайте такую программу только с номером строки автостарта - SAVE "имя" LINE n. Тогда после загрузки программа начнет работать со строки n так, как будто бы прошла команда GO TO n, а не RUN.

И, наконец, простейшая рекомендация - если какое-то число используется в программе многократно, то лучше обозвать его именем какой-либо переменной и дальше вызывать по этому имени. И последнее - имена переменных из одной буквы расходуют значительно меньше места, чем из нескольких, хотя это нарушает читаемость программ. Поэтому такое изменение надо вводить, когда программа уже полностью отлажена. При этом можно удалить из программы и строки REM.

---

<sup>1</sup> В оригинале пропущена строка (Прим. OCR)

# ADVENTURE LESSONS

Окончание, начало на с. 35,55

## 9. Книги, записки, объявления и пр.

Их, конечно, если они встречаются по ходу игры, можно взять и рассмотреть (LOOK) и/или прочитать (READ).

READ BOOK - прочесть книгу

READ NOTE - прочесть записку

READ ADVERT - прочесть объявление

Почти всегда в них находится какая-то информация, жизненно важная для прохождения программы. Хотя иногда фирма, выпустившая программу, вставляет в них ради шутки информацию рекламного характера.

## 10. Музыкальные инструменты.

Особый случай могут представлять музыкальные ноты. Их можно, конечно, прочитать (READ MUSIC), но можно и сыграть на музыкальном инструменте.

PLAY MUSIC - сыграть музыку.

Можно и указать на чем сыграть, например, PLAY PIANO - сыграть на фортепиано.

Для духовых инструментов обычно применяется команда BLOW.

Например:

BLOW WHISTLE - подуть в свисток;

BLOW HORN - подуть в рожок.

Такой командой можно разбудить спящего персонажа, оживить статую, вызвать коня и т.п.

## 11. Растения.

Понятно, что если перед вами дерево, то на него можно взобраться - CLIMB TREE (если это не священное дерево). Но особый и более интересный случай представляют менее значительные растения. И здесь есть несколько вариантов.

Во-первых, слабенький зеленый росток, очевидно, нуждается в поливке. Если у Вас есть вода, то полейте его - POUR PLANT - полить росток; POUR WATER - полить водой.

Посмотрите, что из этого выйдет. Вполне возможно, что на Ваших глазах он вырастет, распустится и даст какие-то плоды, которые можно где-то использовать. Кстати, вообще надо помнить, что в работе с растениями самым интересным могут быть не растения, а их плоды.

Во-вторых, буйная трава (очевидно не нуждавшаяся в поливе) может что-то скрывать. Можно обыскать это место командой SEARCH. Можно попытаться здесь копнуть, чтобы понять отчего это вдруг вокруг все нормально, а здесь выросла буйная трава. Слабонервным лучше этого не делать. Бывает, откапываются останки ("PRICE OF MAGIC").

## 12. Веревка, канат (ROPE).

С ними все просто. Их можно потянуть (PULL ROPE), а также привязать - TIE ROPE. По ним можно и взобраться наверх - CLIMB ROPE.

## 13. Лестница - LADDER.

Совершенно очевидное применение - CLIMB LADDER.

В заключение этой главы мы должны сделать еще пару замечаний общего характера об использовании различных предметов.

Есть очень хорошая команда - USE (использовать), если программа ее понимает. Предположим, что Вы найдете предмет, который вам совершенно не известен, непонятен и что с ним делать Вы не представляете, например Вы нашли ABCDEF. Попробуйте его

применить командой USE:

USE ABCDEF

Если это ни к чему не привело, но Вы убедились, что команду USE программа понимает, то попробуйте повторять ее в разных местах. Может быть, где-то Вы и получите неожиданный результат.

Конечно, при использовании разных предметов надо руководствоваться в первую очередь здравым смыслом, но можно их использовать и не по назначению. Очень простой пример. Во многих программах встречаются лабиринты. Хорошо, если лабиринт обычный. Рисуя карту, Вы пройдете его рано или поздно. Но встречаются, и нередко, очень иррациональные лабиринты. Сделав по нему хотя бы шаг, а затем вернувшись на шаг назад, Вы уже можете не попасть в исходную точку. Там могут быть и коридоры с односторонней проходимостью и переходы на другие этажи. В общем, возможны совершенно нелогичные вещи. Чтобы осилить такой лабиринт, надо использовать имеющиеся у Вас предметы. Бросили предмет, сделали шаг вперед, потом назад и посмотрите (LOOK). Если Ваш предмет лежит там же, то здесь пока все в порядке. Так шаг за шагом "раскалывают" любые лабиринты, а они, как показывает практика, совсем не так страшны, как кажется на первый взгляд.

И последнее. Кажущееся многообразие предметов и действий, которые с ними можно совершить, не очень должно Вас пугать. Дело в том, что хорошая программа так или иначе намекает Вам о том, чего она от Вас ждет. Как догадаться, что в заброшенной часовне надо поднять подгнившую половую доску, чтобы обнаружить тайник? - так ведь Вам говорили, что когда Вы вошли в дверь, доска скрипела.

Когда, закончив игру, Вы пройдете ее еще раз, то увидите, что на все (даже самые головоломные) действия в игре были подсказки и намеки. Ну, а уж если этого не было, то знайте, что именно здесь и проходит граница между великими шедеврами и посредственной подделкой.

Поскольку на рынке программного обеспечения есть несколько программ, позволяющих создавать адвентюрные игры автоматизированным путем (в порядке их появления - QUILL, G.A.C. - GRAPHIC ADVENTURE CREATOR и P.A.W.S. - PROFESSIONAL ADVENTURE WRITING SYSTEM), то неискушенные программисты часто используют их для быстрого создания неполноценной продукции, которая тысячами экземпляров заваливает фирмы, выпускающие программное обеспечение и редакции Синклер-журналов. Хотя, конечно, у нас нет и мысли опорочить программы QUILL, G.A.C. и P.A.W.S. Они являются прекрасным инструментальным средством и, конечно, они дали возможность многим начинающим программистам попробовать свои силы, создать что-то свое и благодаря им некоторые смогли впоследствии достичь высоких результатов.

Тем не менее надо помнить, что если с их помощью не требуется быть программистом, чтобы создать шедевр, то зато надо безусловно иметь писательский талант и нестандартную фантазию, а и то и другое требует многих месяцев и лет напряженной тренировки, так что сразу шедевр не получается.

## 5. Взаимодействие с другими персонажами.

Правильная работа с другими персонажами, также как и работа с объектами, может очень облегчить прохождение программы, в то время как неправильная может испортить все впечатления от программы.

Во первых, всех встреченных Вами персонажей (неважно людей или животных) можно условно разделить на четыре категории.

1. Друзья.
2. Враги.
3. Нейтральные.
4. Неопределенные.

Если с первыми тремя категориями все ясно, то к четвертой мы (то есть "ИНФОРКОМ") относим таких, которые по ходу развития событий могут стать как друзьями, так и врагами - все зависит от того, как Вы будете себя с ними вести.

Итак, рассмотрим основные команды, которые используются при работе с персонажами.

1. Возьмите себе за правило при встрече нового персонажа здороваться. Делается это с помощью команды SAY (сказать). Например, SAY "HELLO" - сказать "ПРИВЕТ".

Это не поможет, если встреченный персонаж Ваш убежденный враг, но в прочих случаях это позволит Вам получить ответ, а из него какой-то намек на то, в каком направлении вести дальнейшую беседу.

2. Предположим, контакт установлен и на Ваше приветствие встреченный персонаж не откусил Вам голову, а ответил что-нибудь или хотя бы пробурчал, следующий шаг - вытрясти из него информацию или еще что-нибудь, наиболее часто применяется команда:

ASK ABOUT ... - спросить о ...

Предположим, Вам надо переправиться через реку и вам встретился лодочник. После приветствий спросите его о лодке:

ASK ABOUT BOAT

Он может прямо запросить оплату, может сразу перевезти Вас, а может и сказать, что некогда ему здесь сидеть и отвечать на глупые вопросы.

Также можно спрашивать и об оружии у оружейника, о хлебе у булочника и так далее.

3. Как правило, за услуги надо платить. Платят чаще всего деньгами. Соответствующие команды - PAY (платить) или GIVE (давать). Например,

PAY MONEY - заплатить деньги.

GIVE COIN - дать монету.

Платить можно не только деньгами, можно расположить персонаж к себе с помощью подарка:

GIVE GOLD - дать золото

GIVE BOOK - дать книгу.

Самое интересное, что в мире сказочных приключений действуют совсем иные соотношения ценностей. Правильно подаренный рваный башмак может сыграть большую роль, чем мешок с золотом.

4. Иные взаимоотношения с врагами. С ними можно либо сражаться, либо убегать.

Убегать просто - RUN.

Сражаться - FIGHT.

Можно еще употребить команду ATTACK – атаковать.

В некоторых играх бывают ситуации, когда разных врагов лучше атаковать разным оружием. Например, против бегемота лучше применить топор (AXE), а против вампира - вязанку чеснока (GARLIC), в этом случае в команде указывают чем атаковать соперника:

ATTACK WITH DAGGER - кинжалом

ATTACK WITH SWORD - мечом

ATTACK WITH AXE - топором и т.п.

5. В некоторых играх общаться с другими персонажами можно и по телефону.

Если Вы хотите позвонить по телефону, можете использовать для начала команду USE PHONE - воспользоваться телефоном, это если из контекста программы само собой разумеется, кому Вы звоните.

Бывает, что по ходу программы Вам подсказывают какой-либо номер (например, Вы его прочитали в найденной в собственном кармане записной книжке). Набор номера выполняется командой DIAL.

Например:

DIAL NUMBER - набрать номер. Сам номер может потом набираться по запросу от программы.

Если Вам известно, кому Вы хотите позвонить конкретно, то можете проверить команду CALL. В принципе CALL по-английски это звать, вызывать, но позвонить по телефону это тоже CALL. Например, CALL PRESIDENT позвонить президенту.

## 6. Использование отгрузки программы на ленту.

Нормальная программа, предназначенная для длительной работы, обязана иметь возможность отгрузки состояния игры на ленту. Обычно это делается командой SAVE.

Естественное назначение этой команды - отложить незаконченную игру до следующего сеанса, в то же время, в адвентюрных играх эту команду применяют просто для закрепления достигнутого успеха.

Всегда, когда идете на рискованное дело, делайте SAVE, это позволит Вам в случае гибели не начинать все сначала, а загрузить отложенное состояние и сэкономить на этом много времени.

Адвентюрная программа, которая не имеет опции SAVE - не программа, имеющая право считаться адвентюрной.

## 7. Полезные практические приемы.

Здесь мы рассмотрим некоторые моменты, не вошедшие в основной текст нашей статьи по причине либо их нетипичности, либо нетрадиционности.

1. Есть очень хорошая команда - BOM. Введена в практику фирмой "LEVEL 9". Это сокращение от BACK ONE MOVE - ОДИН ХОД НАЗАД. Она позволяет взять один ход назад, чтобы избежать нежелательного исхода. К сожалению, она редко применяется в играх, но если кто-то из наших читателей, наигравшись досыта, начнет сам разрабатывать такие игры, то он должен знать, что за наличие такой команды в игре ему многократно скажут спасибо.

2. В программах, связанных с магией и чародейством, вам может быть придется иметь дело с заклинаниями. Заклинание по-английски SPELL. Сами заклинания и их назначение Вам придется найти по ходу игры. Они могут быть в старинных книгах, их можно купить, наконец они могут быть выцарапаны на стене, вышиты на ковре или вырезаны на картинной раме.

Произносятся заклинания командой CAST.

Приведем пример, Вы нашли на задней стенке шкафа надпись FLY (летать). Полагая, что это заклинание, Вы даете: CAST FLY или SAY FLY, после чего оказывается, что Вы можете теперь летать.

Диалог может быть и таким:

> CAST SPELL - произнести заклинание

- WHICH SPELL? - какое заклинание

> DIE - умереть

- O.K.

После этого Ваш соперник погибает.

3. При работе с программой никогда не используйте ругательных выражений, примеры мы по понятным причинам не приводим.

Умная программа их понимает. В лучшем случае она скажет, что джентльмены так не выражаются, а бывает сбрасывается совсем. Если даже она и не сбрасывается, то как знать, может быть в одну из ячеек памяти пошла информация о том, что Вы неприятный партнер и на что это в дальнейшем повлияет?

4. Если в работе с программой Вы зашли в безнадежный тупик, а поэкспериментировать в общем-то хочется, то дадим Вам несколько советов.

Первое - попробуйте с помощью дисассемблера (например MONITOR 48 или MONITOR 16) пробежаться по машинному коду программы в поисках ее словаря. Если он не зашифрован, то это поможет Вам установить список глаголов и существительных. Глаголы - это те команды, которые программа понимает, а существительные - возможные объекты, которые могут встретиться в ходе игры.

Есть интересный топорный прием, позволяющий поэкспериментировать с программой. Отгрузите игру в каком-то месте на ленту. Теперь снова подзагрузите ее, но при загрузке отложенного состояния прервите загрузку, остановив магнитофон. В девяти случаях из десяти загрузка не пройдет. Но иногда эффект бывает поразительным. Все предметы в программе путаются. Вы можете приобрести что-то, о чем и не знали, за

стойкой бара вместо бармена может оказаться инопланетянин, до которого еще надо было бы лететь на другую планету. У Вас может оказаться баснословное количество денег и теперь Вы можете закупить что угодно и все испытать, все попробовать. Как Вы понимаете, результат непредсказуем, но интересен, тем более, что не стоит особого труда, если получится.

Более тонкая химия - это взять отгруженный блок, а он всегда очень короткий, загрузить его в дисассемблер и поменять в нем несколько байтов, а затем посмотреть, что получилось. Правда, почти всегда этот блок бесхэддерный (без заголовка), и потому в дисассемблер загружаться не будет, но для такого дела можете "пришить" к нему заголовок от какого угодно другого блока или сделайте его сами. Дайте команду

```
SAVE "abcd" CODE m,n
```

m, n - почти что угодно. Например, m - 30000, а n - 500.

Когда пройдет первая порция широких полос и маленькая пачка (17 байтов) узких, выключите магнитофон, не выгружая всего остального - вот Вам и "хэддер"

Сшивание делайте в любом копировщике.

Кстати, копировщик COPY-COPY может загружать и бесхэддерные блоки и просматривать их командой LIST и вносить в них изменения командой POKE. Впрочем, мы об этом писали в нашей базовой разработке "Большие возможности Вашего Спектрума" еще полтора года назад и наши старые читатели это знают.

## ЗАКЛЮЧЕНИЕ.

Мы обещали, что уделив по одной странице в месяц, беремся обучить Вас за полгода. Уделив по две страницы, мы уложились за три месяца. Конечно не все команды (глаголы), которые понимают текстовые программы, мы сюда включили, но примерно 85% основных - наверняка. Начинать Вы уже можете. А если кто знает еще хорошие широко встречающиеся команды - пишите, мы напечатаем, ведь могли же мы что-то упустить.

Чтобы мы хотели получить взамен?

Только одно - заинтересованность. Чтобы те, кто сегодня начинают работать со "Спектрумом" не бежали как от чумы от текстовых программ. Помните, что через три года общения с компьютером для 75% пользователей этот жанр становится основным. Вот облегчить этот путь мы и постарались.

Вы приобретете:

1. Практическое применение знаниям английского языка.
2. Полезный и увлекательный досуг.
3. Новых друзей и знакомых.

Почти никому не удастся пройти мощную игру в одиночку. Только обмениваясь достижениями Вы сможете преодолеть нормальную программу средней силы.

# GRAPHIC ADVENTURE CREATOR

В мире программного обеспечения время от времени выпускаются программы, вносящие значительный вклад в написание игровых программ. Одной из таких программ была программа QUILL. Она открыла возможность самим создавать адвентюрные игры для тех, у кого хорошо развито воображение, есть терпение и совершенно нет навыков в программировании.

Конечно, большинство программ, созданных в системе QUILL, оставляют желать лучшего, но самые лучшие из них (например, TOWER OF DISPAIR) показывают каких успехов можно достичь.

Несмотря на то, что программа QUILL очень долго доминировала на рынке, у нее все-таки есть свои ограничения, и программа GAC (Graphic Adventure Creator), выпущенная фирмой INCENTIVE SOFTWARE явилась для нее серьезным конкурентом. Она предоставляет следующие дополнительные возможности:

- анализ сложных предложений;
- многозначные команды;
- компрессия текста;
- широкий спектр графических возможностей.

Пакет GAC включает в себя кроме самой программы еще и небольшой полезный файл Quickstart, являющийся как бы образцовой адвентюрной микроигрой.

Конечно, как и положено серьезной системе, к пакету должно прилагаться объемистое руководство по работе с ним, которого у нас в "Инфорконе" к сожалению нет, но если есть у кого-то из наших читателей, мы с радостью бы его приобрели. Поэтому нам приходится сегодня ограничиваться лишь обзорной статьей на эту тему, тем более, что многие читатели просят именно обзорных статей, для расширения общего кругозора по интересным программным продуктам.

Наше общение с любителями адвентюрных программ, показало, что при наличии упорства с этой программой можно разобраться и без учебника, а та информация, которую мы здесь дадим, поможет сделать первый шаг. Мы будем рады получать письма от тех, кто пробьется в этом направлении через первые трудности и поможет другим, ценным было бы также исследование возможности русификации этой программы. Общие соображения говорят о том, что это не должно быть очень трудно, поскольку в программе где-то есть свой знакогенератор.

После загрузки программы Вам выдается обширное меню различных опций. Самое мудрое, что здесь можно сделать - это загрузить прилагаемый файл "Quickstart".

Это даст полезную базу, от которой можно развивать работу. В этом файле уже есть небольшой словарь, включающий в себя такие совершенно необходимые в любой текстовой программе команды, как GET, LOOK, EXAMINE, INVENTORY и т.п. Там же есть и небольшая система программных сообщений типа:

- YOU CAN'T - Вы не можете.
- WHAT NOW? - Что дальше?
- YOU APE CARRING... - Вы несете...

Конечно их можно переделывать как хотите.

Словарь входит в программу в трех отдельных файлах:

- VERBS (глаголы),
- NOUNS (существительные),
- ADVERBS (наречия).

Процесс заполнения и редактирования словаря изумительно прост. Каждому глаголу, существительному или наречию присваивается номер от 1 до 255. Программа может теперь элементарно работать и с синонимами, поскольку Вы можете разным словам, имеющим одинаковое значение, присвоить и одинаковые номера.

Способность работать с наречиями существенно отличает GAC от QUILL, поскольку теперь Вы можете различать такие команды как:

EXAMINE BOX - проверить ящик и

EXAMINE BOX CAREFULLY - тщательно проверить ящик.

Ввод текстов сообщений и описаний обстановки тоже доставляет удовольствие. Вводя текст Вы можете не думать, как он расположится на экране, не будет ли разрыва в слове при переносе. Просто набирайте текст, и все. Программа сама все сделает за Вас. Правда, программа немного путается со знаками препинания. Считается общепринятым после знака препинания давать пробел, поэтому если знак препинания оказывается последним на строке, то в начале следующей строки этот пробел давать не надо. Здесь программа этого не учитывает и левый край текста может получиться неровным. На каждом шагу работы программа дает вам необходимые запросы, что очень упрощает работу. Так, например, когда Вы закончите описание какой-либо локации, программа запросит от Вас ввода соединений этой локации с другими. Например, если из данной точки можно пойти на север и попасть в пункт номер 15, то Вы вводите:

NORTH 15

Таким простым путем задается карта игры. Изменения в процессе работы вводятся тоже очень легко.

## ЛОГИКА.

Когда введен текст, описания локаций, сообщения, карта и имена объектов (существительные), можно приступать к разработке логики игры. Именно здесь и выявляется вся сила этой программы и именно здесь и наступают основные трудности для Вас.

Существуют три типа условных конструкций, для каждого типа есть свой файл и доступ к ним открывается из главного меню.

Условия высшего приоритета (High Priority Condition) проверяются, перед тем как играющий вводит команду.

Условия низшего приоритета, проверяются после того, как игрок команду подал.

Локальные условия относятся к тем специфическим действиям, которые могут происходить в данной локации, в которой сейчас находится игрок.

Логические операторы, которые могут быть при этом созданы и рассмотрены, очень широки и многообразны. Достаточно сказать, что Вы имеете возможность задавать и проверять 255 флагов (маркеров), с помощью которых программа отслеживает важнейшие события, такие например, как открыта дверь или закрыта. С их же помощью можно оценивать и более сложные вещи:

- сколько весит объект;
- есть он у Вас в наличии или нет;
- есть он в заданной комнате или нет и т.п.

Нижеприведенный пример служит для иллюстрации языка программы.

Предположим, что где-то по ходу игры Вам может встретиться гангстер NICK, вооруженный пистолетом (GUN). В этом случае программа в ответ на команду игрока:

TELL NICK TO GIVE ME THE GUN

(сказать Нику, чтобы отдал пистолет) программа должна, во-первых, дать нужное ответное сообщение и, во-вторых, добавить пистолет к списку объектов, которыми Вы владеете.

Условие в программе будет выглядеть так:

```
IF (VERB 24 AND NOUN 3 AND NOUN 5 AND RES? 6 AND HERE 3) SET 6 MESS 6 5 TO 3  
GET 5 END
```

Если это грубо перевести на русский язык, то получится следующее:

Если команда содержит глагол (VERB) номер 24 (tell) и существительное номер 3 (Nick), а также существительное номер 5 (GUN) и если при этом флаг номер 6 выключен (RESET) и Ник находится здесь же рядом с Вами (HERE 3), то включить флаг 6 (чтобы можно было забрать пистолет только один раз), напечатать нужное сообщение (MESSAGE 6), образовать пистолет как объект в данной локации (5 TO 3) и взять его, добавив к списку предметов, которыми владеет игрок (GET 3).

Если Вы зададите таким образом логику работы по отдельным командам типа

UNLOCK THE DOOR - отпереть дверь

OPEN DOOR - открыть дверь

GO INTO DOOR - войти в дверь,

то и многозначные команды типа: UNLOCK THE DOOR, OPEN IT AND GO IN - будет тоже выполняться!

Обратите внимание, что программа понимает местоимение IT и использует его вместо последнего встречавшегося существительного DOOR.

Чего правда не хватает программе - это она не умеет работать с командами DROP ALL и TAKE ALL.

## ГРАФИКА.

Адвентюрная игра может быть сделана с графическими иллюстрациями, любую картинку из загруженных Вы можете включить и выключить в любом месте игры по желанию.

Картинка изображается в окне, занимающем верхние 2/3 экрана, что пожалуй многовато, поскольку внизу остается недостаточно места для детального описания обстановки, и если описание слишком длинное, то оно начинает скроллироваться до того, как игрок успеет его прочитать.

Программа может рисовать эллипсы, прямоугольники, линии, точки. Можно заполнять фигуры гладким цветом или несложной текстурой. Есть очень хорошие редакторские возможности. Одним словом, создать можете любую иллюстрацию из тех, какие Вам позволят создать собственные таланты. Приличная картинка (не хуже, чем те, которые Вы можете увидеть в программе HOBBIT) занимает примерно 0.5 К.

Замечательной способностью программы является возможность "сливать" (MERGE) изображения. Поэтому Вы можете один раз изобразить "стандартное" дерево и всякий раз, когда надо, вызывать и накладывать его на картинку. То же самое часто используется для стилизованных орнаментов вокруг картинки.

Единственное ограничение - это то, что, как дерево нарисовано на экране, так оно всегда и будет изображаться - в одном и том же месте. Передвигать рисунки программа не умеет.

Заканчивая обзор этой замечательной программы, нам бы хотелось обратить внимание наших читателей на одно существенное обстоятельство.

Мы все время говорили о программе GAC как об инструментальном средстве для создания игр, в данном случае текстовых. Совсем не надо иметь богатую фантазию для того, чтобы сообразить, что это почти готовая система для создания обучающих программ, способных и дать материал с иллюстрацией и поставить вопрос и проанализировать ответ, а затем сделать вывод о том, чтобы похвалить/ пожуричь/ подсказать.

Причем работать с этой системой (создавать учебные комплексы) может преподаватель, хорошо знающий свой предмет, но не знающий программирования. Вы же, конвертировав GAC и сделав из него систему, можете наоборот не быть преподавателем и не знать никаких предметов. Вот Вам и путь к быстрому решению проблем компьютеризации образования. Нам часто пишут молодые люди, освоившие или осваивающие программирование и спрашивают совета куда приложить свои усилия. Возьмите GAC, разберитесь с ним так, чтобы ориентироваться в этой программе, как в своей, замените генератор шрифта на русский, адаптируйте GAC под другой круг задач, напишите достойную инструкцию по работе с ним, и Вы сделаете такой вклад в общее дело, который конечно же не останется незамеченным. Даже если у Вас из этого ничего не получится, Вы приобретете опыт и при этом у Вас появятся новые идеи. Дерзайте, и никакие Ваши усилия не пропадут даром, а мы дадим рекламу Вашим достижениям на страницах "ZX-РЕВЮ".

# SCRABBLE



Эта игра должна быть Вам хорошо известна. Поскольку впервые она была представлена в журнале "Наука и жизнь" еще в шестидесятых годах под названием "ЭРУДИТ", а в последующие годы было выпущено множество настольных вариантов.

Игра вполне оправдывает свое название, поскольку от играющего требуется умение комбинационно мыслить и иметь солидный запас слов.

Компьютерная версия игры, рассчитанная на тех, кто знает или осваивает английский язык, имеет особенную прелесть, поскольку вдвойне будет способствовать повышению Вашей эрудиции.

## Вступление.

Эта инструкция состоит из двух разделов, В первом описаны правила игры "SCRABBLE", а во втором - описание, как играть в нее на компьютере.

## I. Правила игры.

SCRABBLE - это буквенная игра. Количество играющих - от 1 до 4 человек. Цель игры состоит в построении взаимопересекающихся слов наподобие кроссворда. Игра ведется на специальной игровой доске, где размещаются фишки с буквами, имеющими различную ценность.

Каждый игрок стремится получить максимальное количество очков на своем ходе и должен для этого учитывать неодинаковую ценность различных букв, а также использовать премиальные клетки игровой доски (они выполнены цветными). Итоговый счет может быть в диапазоне от 500 до 700 очков, в зависимости от мастерства играющих.

## Игра.

Каждый игрок в начале игры имеет на руках 7 фишек. После каждого хода он получает дополнительно такое количество фишек, какое необходимо для дополнения их общего количества, находящихся у него на руках, до семи.

1. Первый играющий берет две или более букв, составляющих какое-либо слово и размещает их так, чтобы слово проходило через центр доски и читалось слева направо или сверху вниз.

Разнесение слов по диагонали не разрешается.

2. Игрок заканчивает свой ход подсчетом очков и объявлением счета, после чего он подбирает количество фишек до семи. (Конечно в компьютерной версии все за него сделает компьютер).

3. Второй игрок и все последующие игроки добавляют одну или более букв к уже выставленным на доску так, чтобы они составили слово. Все фишки, выставленные в один ход, должны быть выстроены в один вертикальный или горизонтальный ряд. Они должны составлять одно законченное слово. Если, в то же время, они соприкасаются с фишками в соседних рядах, то и с ними они должны составлять слова (по принципу кроссворда). Игрок получает очки за все построенные или измененные в результате хода слова.

4. Новые слова могут быть составлены:

- добавлением одной или более букв к слову или к буквам уже имеющимся на доске.
- размещением слова под прямым углом к уже имеющемуся слову. При этом новое слово должно использовать хотя бы одну букву ранее имевшегося слова.
- размещением слова параллельно имевшемуся, но при этом соседствующие клетки также должны составлять слова в перекрестном направлении.

5. Буквы после их установки перемещать нельзя.

6. В наборе букв есть две пустые фишки, которые могут быть использованы игроком по желанию вместо любой буквы. Выставляя пустую фишку, игрок должен указать какой буквой он ее назначает, после чего это назначение остается на все время игры и никем из играющих не может быть переопределено.

7. Каждый игрок может вместо хода произвести замену любого числа имеющихся у него букв. Сначала он сбрасывает ненужные ему фишки, затем вытягивает новые, после чего сброшенные фишки смешиваются с содержимым "котла". Дальнейший ход он делает в порядке очереди.

8. Разрешенными считаются словарные слова, за исключением имен собственных, слов иностранных языков (не английского), а также слов с апострофом, в случае сомнений в правописании слова справляйтесь по словарю. Любое слово может быть оспорено до того, как следующий игрок сделает свой ход. Если слово признается недействительным, игрок обязан забрать назад свои фишки и теряет ход.

9. Игра продолжается до тех пор, пока все фишки не будут вытащены из "котла" и один из играющих не выставит все имеющиеся у него на руках буквы или до тех пор, когда дальнейшие ходы станут невозможными для всех играющих.

### **Счет.**

10. Ценность букв неодинакова. Чем чаще данная буква встречается в словах английского языка, тем меньше ее ценность, т. е. тем меньше очков Вы получите за то, что нашли способ ее применения. Таблицу расценок букв для справки Вы всегда сможете вызвать во время игры, кроме этого, Вы можете вызвать и таблицу, показывающую распределение фишек по их количеству. Обратите внимание на то, что ценность пустой фишки равняется нулю.

11. Счет подсчитывается после каждого хода. Очки образуются суммированием оценок всех букв, входящих в построенное или измененное слово плюс премиальные очки за размещение фишек на премиальных полях.

12. Поля премиальных букв:

- голубые поля удваивают ценность букв, помещенных на них;
- синие поля утраивают ценность помещенных на них букв;

13. Поля премиальных слов:

- зеленые поля удваивают ценность всего слова, проходящего через них;
- красные поля, соответственно, ее утраивают;
- если в слове есть премиальные буквы, то сначала прибавляются премиальные очки, а только затем ценность слова удваивается или утраивается;
- если слово проходит через два зеленых или красных поля, то оно удваивается или утраивается дважды (т. е. увеличивается в 1 или 9 раз соответственно);
- обратите внимание на то, что центральное поле имеет зеленый цвет и следовательно, цена первого слова двойная.

14. Премиальные поля действуют только на тот ход, в котором они используются. В последующих ходах ценность букв и слов не увеличивается.

15. Если пустая фишка попадает на поле премиального слова, то ценность слова удваивается или утраивается, несмотря на то, что ценность пустой фишки равна нулю.

16. Если в результате хода образуются два или более слов, то их ценность суммируется. Ценность общей буквы (даже если она размещена на премиальном поле) учитывается для каждого слова независимо.

17. Если игрок в один ход выставит все свои семь фишек, то он получает дополнительные призовые 50 очков.

18. В конце игры производится окончательный подсчет очков. Финальный счет для каждого из игроков уменьшается на величину, равную сумме цен фишек, оставшихся несыгранными у него на руках. Если один из играющих израсходовал все свои фишки, то его очки увеличиваются на величину, равную сумме очков всех фишек, оставшихся несыгранными у других партнеров.

## II. Игра с компьютером.

После загрузки программы Вам надо ответить на ряд вопросов, чтобы определить условия игры.

1. Какой у Вас телевизор? (цветной - нажмите С, а если черно-белый - нажмите В.)

2. Не хотите ли Вы загрузить предварительно отложенную игру? Да - "Y", нет - "N".

3. Выберите количество играющих: от 1 до 4.

Теперь ответьте на вопросы для каждого играющего отдельно.

4. Не хотите ли Вы, чтобы в качестве этого играющего выступал компьютер? Да - "Y", нет - "N".

5. Если "Да", то на каком уровне сложности должен играть компьютер (нажмите от 1 до 4. Уровень 4 - самый сильный).

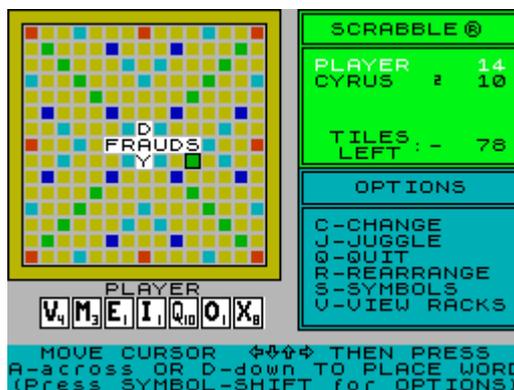
6. Введите имя игрока (максимум 8 букв для людей. 7 букв для компьютерных игроков).

И, наконец:

7. Хотите ли Вы видеть как компьютер думает? Нажмите "Y" или "N". Если Вы выберете "N", то пробные ходы компьютера не будут изображаться на экране, а также не будут изображены фишки, имеющиеся у него на руках.

8. Надо ли Вам, чтобы компьютер делал паузу между своими ходами? (Y/N)

После этого компьютер определит (случайным образом) кто из играющих будет ходить первым, нарисует игровую доску и раздаст (совершенно случайным образом) фишки.



На экране будет изображена доска, справа от нее написаны имена игроков и счет, а также список возможных действий.

Там же изображаются фишки игрока, который в данный момент должен делать ход.

### Главное меню.

#### V - VIEW RACKS

Этот выбор позволяет увидеть фишки, имеющиеся на руках у всех игроков. Возврат в главное меню выполняется одновременным нажатием SYMBOL SHIFT и 0.

#### S - SYMBOLS

По этой опции вызывается другой экран, на котором указаны премиальные поля. Возврат через SYMBOL SHIFT и 0.

R - REARRANGE - позволяет переместить фишки, находящиеся у Вас на руках. После нажатия "R" просто наберите их на клавиатуре в том порядке, в каком Вы хотите их видеть.

C - CHANGE - Вы можете поменять любые (или все) свои фишки вместо хода. После нажатия "C" Вы можете нажать на клавиатуре те буквы, которые хотите поменять. Если ошибетесь, пользуйтесь клавишей DELETE. В конце нажмите ENTER. Если хотите просто пропустить ход, нажмите ENTER без набора букв.

J - JUGGLE - это инструкция для компьютера - перемешать фишки ходящего игрока в случайном порядке.

T - TILE VALUES - по этой команде изображается алфавит с указанием ценности каждой буквы. Возврат через SYMBOL SHIFT и O.

D - DISTRIBUTION - изображает доступное количество каждой из букв в начале игры.

Q - QUIT - прерывание игры. После этого можно начать новую игру или выгрузить игру в промежуточном состоянии на чистую ленту.

### **Как делать ход.**

После того, как Вы придумали слово, Вам надо указать компьютеру, где его разместить.

Перемещайте курсор по доске с помощью курсорных клавиш 5,6,7,8 к полю, с которого начнется слово. Теперь нажмите "A", если слово пойдет слева направо, или "D", если слово пойдет сверху вниз. Далее выстраивайте свое слово простым нажатием клавиш на клавиатуре. Когда закончите, нажмите ENTER.

Если ход был выполнен по всем правилам, компьютер подсчитает очки и спросит, устраивает ли Вас это или попробуете другой ход. Ответьте "Y" или "N". Если Вы ответите "Y", компьютер выдает вам новые фишки взамен сыгранных. Теперь ход перейдет к следующему играющему.

### **Ход пустой фишкой.**

Чтобы пойти пустой фишкой, сначала нажмите пробел (SPACE). После этого нажмите любую букву, которая Вам нужна, на месте пустой фишки.

### **Запрещенные ходы.**

Если Вы сделаете ошибку или попытаетесь сыграть не по правилам или допустите грубую ошибку, компьютер сообщит об этом и ход можно будет повторить.

### **Оспаривание хода.**

После того, как Вы выставите слово, компьютер проверит его по своему словарю, в котором более 11 тыс. слов. Если в нем такого слова нет, то он попросит подтвердить правильность этого слова.

Да - "Y". Нет - "N". Если Вы нажмете "Y", то компьютер Вам поверит и слово будет засчитано.

Проверку может сделать и другой играющий. Он может запретить Вам нажимать "Y", если Вы ошиблись.

Вам проверять компьютер не надо. Его словарь был проверен экспертами.

### **Окончание игры.**

В конце игры компьютер подведет итог с учетом того, какие фишки остались на руках у игроков.

### **Откладывание игры.**

Незаконченную игру можно отложить. Воспользуйтесь выбором QUIT в главном меню, затем пользуйтесь указаниями на экране. Не забудьте о необходимости использования чистой кассеты. Выгружается не вся программа, а только блок данных.

### **Загрузка отложенной программы**

Отложенную игру можно дозагрузить после загрузки главной программы. Можно также загрузить отложенную игру после команды QUIT.

### **Стратегия.**

Компьютер в игре руководствуется определенной стратегией. Так, например, даже на самом сильном уровне (4) он может не выставлять максимально возможное слово для того, чтобы сберечь какую-либо полезную букву, например "S". В то же время, компьютер не применяет принципов негативной стратегии (например, он старается избегать открывания

для Вас премиальных полей). Он выиграет у Вас и без этих ухищрений.

### **Словарь**

Словарь компьютера содержит более 11000 слов. Это беспримерное достижение для сравнительно малой машины. Для того, чтобы он мог помнить максимальное количество высокоценных слов, многие простые слова, встречающиеся очень часто в повседневной жизни и хорошо известные Вам, были опущены. Поэтому пожалуйста не смущайтесь, если компьютер будет оспаривать очевидное слово. Просто подтвердите его нажатием клавиши "Y".

# FORUM

Мы рады сообщить нашим читателям, что объявленная нами атака на игру SABOTEUR-2, производства фирмы DURELL вылилась в самый настоящий коллективный штурм, о результатах которого мы Вам сейчас и расскажем. Мы получили десятки писем от энтузиастов и поклонников этой игры и можем сказать, что практически все они написаны с большим знанием дела. Приятно сознавать, что у сложной зарубежной программы нет шансов противостоять дружному натиску наших экспертов, но самое приятное - это то, что эти письма дополняют друг друга и сами эксперты из сегодняшнего обзора найдут для себя что-то новенькое, что не удалось обнаружить им и что нашли другие.

Лабиринт состоит из трех соединенных подвалами зданий. Он содержит три лифта и пульта управления к ним. Главное помещение лабиринта - туннель, проходящий две трети лабиринта и выходящий наружу на уровне земли.

По всем этажам лабиринта разбросаны ящики, в которых можно найти ключи, кинжалы, шурикены (метательные звезды ниндзя), а также фрагменты перфолент.

По комнатам зданий ходят охранники с пантерами. Они хорошо вооружены. У некоторых есть даже огнеметы. Для борьбы с ними Вы можете использовать найденное оружие или применять приемы каратэ (удары рукой или ногой).

В правом здании находятся ракеты и два пульта управления. Первый пульт - для загрузки перфоленты, второй - отключение электронной защиты.

При загрузке игры внимательно рассмотрите здание, изображенное на заставке, это поможет Вам ориентироваться в игре. Приготовьте карандаш и бумагу, желательно по ходу игры рисовать лабиринт.

После загрузки игры нажмите "ENTER" - перед вами главное меню программы. Вы можете выбрать себе нужные органы управления, а также миссию ("M"). Всего в этой игре, как сообщают читатели, до девяти миссий. После завершения очередной миссии Вам выдается пароль на следующую.

Вот расписание миссии, как оно составлено по письмам читателей.

1. ENTER - пройти по зданию, ликвидировать несколько охранников и покинуть здание на мотоцикле.

2. JONIN - пройти по зданию, ликвидировать несколько охранников и покинуть здание через туннель (пешком!).

3. KIME - найти два фрагмента перфоленты и покинуть здание на мотоцикле.

4. KUI KIRT - собрать пять фрагментов перфоленты, отключить электронную защиту, покинуть здание на мотоцикле.

5. SAIMENJITSU - то же, но собрать 9 фрагментов перфоленты.

6. GENIN - собрать 9 фрагментов перфоленты, ввести их в пульт управления ракетой, отключить электронную защиту и покинуть здание на мотоцикле.

7. MI LU KATA - собрать 11 фрагментов перфоленты, ввести в пульт управления и покинуть здание через туннель.

8. DIM MAK - собрать 14 фрагментов перфоленты, ввести их в пульт управления ракетой и покинуть здание через туннель на мотоцикле.

На каждом последующем уровне время, отведенное на выполнение задания, сокращается на 50%.

По всей видимости существует какое-то стандартное описание игры. Потому что до этого момента письма читателей в основном совпадают, но самое интересное наступает дальше.

Товарищ Скуратов из Муромы сообщает о девятом (!) уровне, пароль которого - SATORI (об этом же сообщает т. Каневский из Киева), а также о наличии в игре комнаты на нулевом этаже в правой части здания, после пребывания в которой Ваша энергия больше не расходуется. Можете падать с любых высот и смело сражаться с охраной. Сложность отыскания этой комнаты в том, что туда ведет невидимая лестница.

Товарищ Боярский из Ленинграда приводит и схему для отыскания этой комнаты:



Товарищи Подопригора из Днепропетровска и Шилин из Симбирска тоже отмечают сложность отыскания этой комнаты и приводят несколько POKES для облегчения выполнения миссий:

POKE 37122,0 (время)

POKE 61340,201 (энергия).

Товарищ Каневский дополняет эти данные следующими:

37121,0; 37122,0 - время

61338,182; 61382,182 - энергия,

Товарищ Дейна из Новгорода подчеркивает разницу в том, как завершаются миссии - уехать на мотоцикле (через левый выход) или уйти пешком - через правый. Когда прибывает мотоцикл - появляется сообщение BIKE ARRIVED.

Он же делится несколькими полезными приемами, которые могут пригодиться в игре.

1. Ввод перфоленты в пульт управления ракетой.

На высоких уровнях (с четвертого) появляется зеленый код. Это означает, что Вам надо добраться до правого здания, добраться до него можно по канату или пешком через нижние этажи. По канату надо проходить не останавливаясь. Если в длинном обходном пути через нижние этажи не хватает энергии, можно отдохнуть на дереве, в лифте или на ровной плоской площадке.

Когда переберетесь в правое здание, надо подойти к пульту с левой стороны у второго яруса ракеты, чтобы он загорелся и нажать FIRE. Если у Вас есть необходимое количество перфолент, то появится сообщение CODE ACCEPTED.

2. Полезный совет для рукопашного боя. Если у Вас мало сил и Вы не уверены, что Вам удастся убежать, сделайте следующий фокус: подойдите к краю экрана так, чтобы половина вашего туловища исчезла, развернитесь и сядьте. Теперь охранник не достанет Вас ни ножом, ни ногой и Вы можете сделать с ним, что хотите. Для пантеры это не проходит, но сидя ее легко убить. Правда сидя нельзя метнуть предметы.

Многokrатно пройдя игру, товарищ Дейна убежденно считает ее одной из лучших игр для "Спектрума" и сообщает, что есть непроверенные сведения о том, что где-то в программе есть комната, в которой можно остановить время. Ему ее найти не удалось. Но ведь должен же кто-нибудь что-то об этом знать?

Как мы и ожидали и, честно говоря рассчитывали, волну писем вызвала печать инструкций к программе "ELITE". Это не случайно, мы ведь сказали, что программа намного глубже, чем то, что написано в инструкции. Фирма специально оставила в ней неосвещенными многие вопросы и программа несколько лет держала в напряжении любителей этой игры. Ей посвящались рубрики в Синклер-журналах и систематические исследования.

Как оказалось, у нас тоже очень много межгалактических ассов, готовых немедленно ринуться в космические битвы. Правда, некоторые обратили внимание на наши слова о том, что фирменная инструкция имеет громадные размеры и мы дали ее в сокращении. Заподозрив нас в утаивании информации, они просят выслать им ксерокопии полной инструкции.

Конечно мы не ангелы и у нас бывают ошибки, но сознательный брак мы не делаем. Если мы что и выкинули из оригинального текста, так это только лирику (правда не всю, - нужно было бы еще больше).

Учитывая нынешний дефицит в услугах по копированию и очень высокие цены (для нас - от 75 к. до рубля за страницу) мы конечно никаких выкопировок для огромной массы желающих делать не можем, да это и не нужно. Давайте лучше вместе разбираться с этой программой.

Теперь об ошибках. В главе 13 выпали строки там, где описана стыковка с орбитальной станцией, что ввело в заблуждение т. Рутковского и некоторых других. Принося извинения, даем подробную консультацию на сей счет. После того, как Вы развернулись носом ко входу в станцию, установите сначала минимальную скорость сближения. Затем совместите оси корабля и станции с тем, чтобы во-первых они совпадали, а во-вторых корабль двигался строго перпендикулярно поверхности станции. Лучше всего оси совмещать парными импульсами. Нажали на клавишу "вверх" на долю секунды. Отпустили, подождали некоторое время и нажмите на клавишу "вниз" на такое же время. Когда оси совмещены, начинайте сближение. Когда приблизитесь достаточно, увидите, что входное отверстие имеет прямоугольную форму, этот прямоугольник вращается, т.к. станция вращается вокруг своей оси. Чтобы нормально пристыковаться, Вам надо тоже закрутить свой корабль вокруг его оси (клавишами "вправо/влево").

Если скорости вращения совпадут, то прямоугольное отверстие "застынет" на экране. Можете причаливать. Если Вы увидите, что полностью остановить прямоугольник нельзя, если он "бьет" после синхронизации вращения, значит Ваши оси не совпадают. Чем ближе к станции, тем сильнее биения. Подход надо повторить. Мы делали так. Почти у самого входа на станцию гасим скорость почти до нуля, делаем крутой разворот по тангажу на 180 градусов (клавишей "вверх" или "вниз"), переключаем экран на задний. Когда аппертура станции (приемное отверстие), появится в центре заднего экрана, останавливаем отработку тангажа подачей противоположного импульса и резко набираем скорость, удаляясь от станции, отойдя на приличное расстояние, опять сбрасываем скорость, разворачиваемся по тангажу на 180, переключаемся на передний экран, ловим аппертуру в центре экрана, сближаемся, тормозимся, отрабатываем угол крена ("влево"/"вправо"), синхронизируемся и, если все в порядке, стыкуемся, иначе - еще раз. Через неделю Вы будете это делать вдохновенно и артистично.

Если все это кажется Вам очень сложным, то пусть Вас утешает мысль, что это очень похоже на управление реальным космическим кораблем, и здесь Вы хоть чуть-чуть почувствуете как работают наши космонавты на ручной стыковке с "Салютом".

Теперь ответы на некоторые вопросы.

Товарищ Зайцев из Воронежа заслуженно получил ранг COMPETENT, но дальше дело не движется, ранг не меняется, а порой даже сбрасывается до "HARMLESS", в самом худшем подозревается собственный компьютер ("Балтика").

Что касается снижения ранга, то это что-то странное, а в остальном мы можем предположить, что видимо настало время проявить свои силы в исполнении специальных миссий. Откроем секрет, которого нет и не было в инструкции. По мере вашего роста в

глазах "Спектрума", он может Вам дать спецзадание. На "Сп. 48" их три, а в версии 128K их шесть. Первое задание - "Взрыв сверхновой" - Вам предложат спасти население некой планеты, которой угрожает гибель, и щедро за это расплатятся. Но миссии выдает компьютер сам, Вам их вызвать нельзя.

Ряд вопросов, на которые нет ответа:

Как поменять корабль на корабль другой марки? Как доставить "Таргон" в исследовательский центр? Как получить пилотскую лицензию? Если записано, что "корабли, отказывающие отвечать на запрос - пираты...", - то как послать запрос? И другие подобные. На все эти вопросы ответ - НИКАК. Эти вопросы возникли потому, что мы не всю "лирику" выкинули из фирменной инструкции. Это ведь только антураж, игра ведь аркадная, и эти события - ПОДРАЗУМЕВАЮТСЯ. Вывод Вы должны были сделать совсем другой. Если хотите строить в космосе честную карьеру и не заниматься пиратством, то при сближении с космическим телом не открывайте огонь первым. Ждите, когда корабль себя проявит как враг, тогда с ним и расправляйтесь.

Есть еще серия вопросов типа: Сколько энергетических отсеков можно купить одновременно? Одноразовая ли спасательная капсула? Можно ли энергетической бомбой уничтожить станцию? и другие подобные. Ответ на все эти вопросы должен быть таким - ПРОВЕРЬТЕ И УЗНАЕТЕ. Дело не в том, что нам жалко дать такую информацию, но на ход игры она влияния не оказывает, а существует очень хороший принцип компьютерных "хаккеров" - "ВСЕ, ЧТО МОЖНО ПРОВЕРИТЬ, НУЖНО ПРОВЕРИТЬ". Только так можно стать экспертом в любой программе - будь то примитивная "стрелялка" или новый язык программирования.

С интересной мыслью обратился наш читатель Довженко В.П. из Киева. Он предложил использовать в игре джойстик, что гораздо удобнее, для этого надо после загрузки программы не нажимать клавишу BREAK, а нажать подключенный джойстик влево. Если же в этот момент нажать клавишу "курсор влево", то можно будет управлять курсорными клавишами.

В нашей почте лежит обширнейшее письмо с килобайтами ценнейшей информации от т. Дейна. Если их все собрать вместе и обработать, то получился бы бестселлер под названием "О чем умолчал ИНФОРКОМ". Нам самим эта работа очень понравилась, и мы сообщаем адрес автора. От себя можем добавить, что если многое из того, о чем он пишет, не явилось более поздними "доработками" хакеров, то его сведения превосходят то, что вскрыли западноевропейские пилоты.

173024, Новгород, ул. Попова, дом 15, корп. 1, кв. 39. Дейна А. А.

Сообщаем также адрес еще одного эксперта по проблемам "Элиты", изъявившего желание вести консультирование и обобщать поступающую информацию:

325026, Херсон, пер. Комбайновый, д. 11, кв. 3, Жаров Р.М.

Если Вы, уважаемые читатели, разделяете нашу мысль о том, что "ФОРУМ" заработал и набирает обороты, то может быть начнем новые атаки? В план очередного штурма включены программы по письмам наших читателей:

TOTAL ECLIPSE (\*INCENTIVE\*)  
MOVIE (\*IMAGINE\*)  
BATMAN (\*OCEAN\*)  
THE GREAT ESCAPE (\*OCEAN\*)  
THEIR FINEST HOUR (\*CENT.COM.\*)  
SHADOWFIRE (\*BEYOND\*)  
STRIKE FORCE COBRA (\*PIRANHA/MACMILLAN\*)

Есть идеи? Пишите, только просьба: одно письмо - одна проблема. Иначе с письмами не управиться, не срабатывает принцип распределения обязанностей между сотрудниками при обработке почты.

Представьте себе, если в одном письме заказ на новые разработки, просьба о досылке какого-нибудь выпуска "РЕВЮ", утерянного по вине почты, вопросы по работе с какой-нибудь программой, предложение своих услуг и полезные идеи для "ФОРУМа". Нам очень трудно работать таким письмом, неясно кому поручить его обработку.

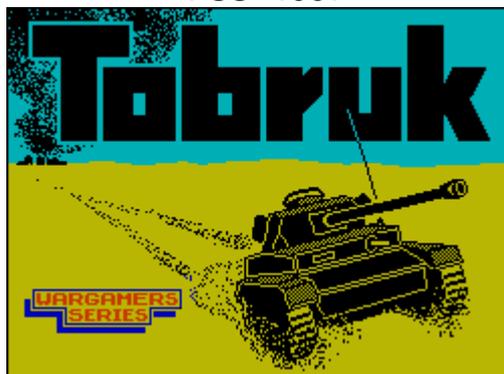
Настала пора подводить итоги конкурса, который мы объявили в конце прошлого года. Напомним, что это конкурс на лучшую тематическую подборку игровых программ для "Спектрума".

Воспользовавшись тем, что этот выпуск имеет двойной объем, и в рекламном разделе образовалось место, мы привели списки поступивших проектов в "Рекламном приложении" и представляем их на Ваш суд.

# STRATEGIC GAMES

## TOBRUK

"PSS" 1987



Танковые сражения в пустыне - вот тема стратегической игры TOBRUK. В основу игры положены реальные события боев второй мировой войны в Северной Африке.

Тобрук - маленький городок на Средиземноморском побережье Ливии. В январе 1941 г. он был занят английскими войсками и с апреля по декабрь 1941 г. выдержал длительную осаду объединенных немецких и итальянских войск.

Играть можно как вдвоем, так и против компьютера. В последнем случае компьютер исполняет роль союзных войск, а Вам предоставляется испытать себя в качестве германских вооруженных сил и попытаться захватить Тобрук.



Ко времени начала немецкого наступления союзные войска не успели закончить минирование подступов к городу. Тем не менее, даже незавершенная линия обороны поставила германское командование перед тройной проблемой. Что делать:

- прорывать оборону невзирая на потери?
- искать обходные пути и возможности для маневра?
- полагаться на длительную работу инженерных войск по подготовке прорыва?

Основной же целью их плана наступления являются шесть оазисов, через которые идет снабжение войск союзников.

Как и в большинстве стратегических игр, игра проходит в виде последовательности ходов. Во время хода войска могут перемещаться и атаковать армии противника. Если с помощью джойстика или курсорных клавиш установить курсор на боевом соединении, то на экране появляется окно, в котором показаны:

- наименование соединения;
- количество пехоты (INF);
- артиллерии (ART);
- бронетехники (AFV);
- условия снабжения (SUP);
- и фактор мобильности (MOB).

Фактор мобильности определяет на сколько полей может быть передвинуто это соединение за очередной ход. Естественно, на него оказывает влияние рельеф местности,

но также и качество снабжения.

Обе воюющие стороны очень сильно зависят от условий снабжения (тылового обеспечения). Снабжение осуществляется боеприпасами, горючим и подкреплениями. В принципе игра в Тобрук, в отличие от многих других стратегических игр - это в первую очередь игра в организацию тылового обеспечения.

Союзники получают свое обеспечение через сеть транспортных путей, опирающихся на шесть ключевых оазисов и вынуждены отступать по мере их потери. Германские же войска имеют два мобильных снабженческих подразделения, которые могут питать войска, расположенные от них на расстоянии не более шести игровых полей. Поэтому если Вы держите свои войска в компактных группировках, у Вас больше шансов выиграть сражение. Но имейте в виду, что Вы должны обеспечить самую лучшую защиту отрядам снабжения, иначе грузовики будут уничтожены и игра проиграна.

В отличие от некоторых стратегических игр в Тобруке есть командная стадия игры. На этой стадии Вы можете назначить одну из шести возможных диспозиций своим воздушным силам и инженерным подразделениям. Так, например, воздушным силам может быть отдан приказ на завоевание превосходства в воздухе или на поддержку наземных атак. В свою очередь инженерным подразделениям может быть поставлена основная задача на восстановление пораженной бронетехники или на расчистку минных полей или на организацию противодействия вражеским танкам. Каждая из диспозиций изображается цветным квадратом на экране, цвет которого меняется в зависимости от того, как идет выполнение данной задачи. Красный цвет означает, что в выполнении данной задачи противник Вас обошел, желтый - Ваши успехи примерно равны, а зеленый - успех на вашей стороне. Если в обеспечении наземных подразделений Вы вышли вперед, то имеете право на проведение дополнительной атаки или воздушного рейда против вражеского соединения. Если в расчистке минных полей Вы получите зеленый квадрат, то один участок будет освобожден от мин. Когда расчистите их достаточно, сможете порадовать противника неожиданной атакой.

Как и некоторые другие игры фирмы PSS, Тобрук имеет в своем сценарии аркадные вставки. Это как бы миниигра в большой игре. В ней Вы можете проверить свои силы в качестве командира танка на поле боя. Вы можете выбрать один из четырех постов управления танком и вооружением:

- общий обзор (карта);
- пулемет;
- орудие (башня);
- ходовая часть (движение).

Ваша задача подавить как можно больше целей, располагая ограниченным боекомплектom. Работа требует недюжинной реакции, поскольку постоянно требуется переключаться между четырьмя экранами и делать это надо очень быстро, пока Вас не разбили. Ваш успех в этой аркадной вставке определяет результат сражения между вашими соединениями и соединениями противника. Пока у Вас нет необходимого опыта, лучше избежать этих аркадных вставок, благодаря тому, что их можно отключать по желанию.

Даже и без них Тобрук выглядит одной из наилучших стратегических игр.

## LEGIONS OF DEATH

"LOTHLORIEN" 1987  
LEGIONS ◀ OF  
◀ DEATH ▶

Для сюжета этой интересной игры фирма выбрала один из наименее исследованных периодов военной истории. Действие происходит в Средиземноморье где-то между 264 годом до нашей эры и 146 годом до н.э. и относится к противостоянию флотов Рима и Карфагена. Те, кто увлекаются древней историей, уже поняли, что речь идет о Пунических войнах. Исторический аспект таков: Рим и Карфаген схватились в жестокой войне за обладание торговыми путями по Средиземному морю. К успеху в этой длительной и изнурительной войне придет тот, кто обеспечит контроль за основными проливами.

"Легионы смерти" - игра для двоих, но в нее можно играть и в одиночку против компьютера. Ваша задача - разгромить корабли противника, но в то же время, Вы должны обеспечить надлежащую охрану собственных портов. Успех в игре зависит от Ваших способностей проводить военные операции на обширных просторах, а также от умения принять правильное решение в сложных ситуациях, когда время на решение ограничено, а от правильности стратегии зависит многое.

Оригинальный подход, необычный для стратегических игр состоит в том, что Вам самому предлагается назначить условия победы. Это может быть количество захваченного золота, количество уничтоженных кораблей противника или размеры захваченных территорий. Благодаря такой возможности у Вас появляется способ регулировать продолжительность игры и уровень ее сложности.

"Легионы смерти" можно было бы с полным правом назвать "Легионами пиктограмм". Пиктограммы были изобретены, чтобы обеспечить пользователю простоту и наглядность ввода в игру информации, но когда их становится очень много, то работа с ними может порождать трудности. Впрочем, это не упрек в адрес "Легионов смерти", потому что программа очень и очень глубока и без них с ней не разобраться.

Игра состоит из нескольких стадий. На первой стадии Ваша задача - построить и оснастить флот, а также нанять экипажи. Для всего этого Вы имеете 1000 единиц золотого запаса. Выбор корабля выполняется из пяти предлагаемых - от быстрорходной, но слабозащищенной биремы до могучего и неповоротливого гептареса. После того, как корабль выбран, он оснащается парусами, надстройками и пр. Среди модификаций Вы можете выбрать даже и специальный прочный настил, идущий вдоль борта, предназначенный для упрощения десантирования на вражеский корабль в абордажном бою (CORVUS). Правда надо иметь в виду, что он точно также упрощает и вторжение на Ваш собственный корабль.



На этой стадии игры так много возможностей для выбора, что она может и затянуться, особенно если Вы сели в первый раз и еще не научились ловко порхать между пиктограммами.

На следующей стадии Вам надо расположить свои суда по Средиземному морю. Здесь все проходит просто, особенно если учесть, что в Вашем распоряжении две карты - одна крупная карта Средиземного моря и мелкая локальная карта зоны боевых действий.



После этого наступает самая трудная часть игры - сражение, и здесь опять Вы перемещаете свои суда, отдаете им приказы с помощью пиктограмм. Когда два судна вступают в ближний бой, то в нижней трети экрана детально изображается картина схватки.

Эта игра определенно доставит удовольствие тем, кто ищет разнообразие после многочисленных сухопутных, боев. Совершенно очевидный исторический дух присутствует в программе и доставит радость тем, кому надоели сюжеты Второй Мировой или Наполеоновских нашествий. Начинаям игра против компьютера может показаться скучноватой, в то время как опытные стратеги найдут в глубинах этой игры то, чего им давно не доставало. Управление пиктограммами поможет начинающему очень быстро "включиться" в игру и начать ее освоение, но по мере все более глубокого погружения техника их использования становится все сложнее и сложнее и игра требует уже профессионального подхода. В общем, в этой игре найдут свои положительные моменты любители всех категорий, а заодно это поможет пробудить интерес и желание узнать побольше об эпохе Пунических войн.

## BATTLEFIELD GERMANY

"PSS" 1987



Это полноэкранный имитатор войны, которая не должна никогда произойти. Речь идет о войне между НАТО и Вооруженными силами Варшавского Договора. История возникновения гипотетического конфликта достаточно интересна и поучительна, но имеет малое отношение к самой игре и мы ее приводить не будем.

Вы можете сами выбрать за какую сторону Вы будете играть, если игра ведется вдвоем. Если же Вы играете против компьютера, то придется вам воевать за НАТО, но не расстраивайтесь - все равно Вам не победить ВС ВД.

В игре задействовано невероятно большое количество соединений. Среди них есть и пехота и мотопехота и бронетанковые войска, а также парашютные, вертолетные части и даже горные стрелки. Все это перемещается по карте Европы, которая имеет возможность скроллить во всех направлениях. Очень удобно, что в правом верхнем углу размещено окно, в котором в укрупненном масштабе показана карта Европы с нанесенной на ней стратегической обстановкой. Соединения на этой карте показаны в виде мигающих точек. На основной карте соединения представлены пиктограммами. Информацию по каждому соединению можно получить, если ввести в его бокс курсор.



Игра разыгрывается в шесть или в двенадцать ходов. Каждый ход может занять до часа времени и состоит из двух этапов.

На первом этапе выполняется снабжение войск. Компьютер выполняет это автоматически, а на втором производится управление войсками - их перемещение и нанесение ударов. Цель - захват территории. Компьютер играет очень жестко и наказывает

за любые стратегические или тактические промахи. Если Вы не чрезвычайно опытный игрок в такие настольные игры, привыкший к шестигранным игровым полям, то у Вас очень мало или почти никаких шансов противостоять компьютеру. Эта игра - для экспертов, начинающим лучше сначала отточить силы на ТОБРУКЕ или чем-нибудь подобном.

## SAMURAI

"CRL" 1986



Это игра для одного игрока, имеющая три различных сценария и три уровня сложности. Вы управляете сборным отрядом из воинов различного боевого амплуа. Ваша задача - уничтожить вражеские силы, охраняющие замок.

Управляется игра методом пиктограмм и начинается с того, что Вы собираете под свои знамена войско, выбирая себе воинов четырех типов. Каждый из них чего-то "стоит", а ваши возможности расплаты ограничены.



В Вашем распоряжении:

Ашигари - легкая пехота.

Самураи – традиционные японские воины.

Конные самураи.

Смертоносные ниндзя.

Механика игры очень проста и состоит в перемещении и нападении. Особенно аккуратно надо распоряжаться с ниндзя, поскольку это единственные воины, способные наносить удар на расстоянии. Самая лучшая тактика их применения - диверсионная "бей и беги".



Игра настолько проста, что ее можно смело рекомендовать начинающим карьеру в стратегических играх. Во многом она напоминает игру SWORDS OF BANE фирмы CCS.

# BATTLE OF BRITAIN

"PSS" 1987



Эта игра относится к событиям Второй Мировой войны, а именно к 1940-ому году. Планируя вторжение на Британские острова, Гитлер отдал приказ командующему имперскими люфтваффе Герингу подавить Английские Королевские Воздушные силы и завоевать превосходство в воздухе.

Учитывая крайнюю скудность снабжения английских воздушных сил, почти любая тактика Геринга принесла бы успех, если бы она упорно проводилась в действие, но тому показалось, что гораздо надежнее переключить свой воздушный флот на ночные бомбардировки мирных городов. Конечно к подавлению английских ВВС это не привело по той простой причине, что "Спитфайры" и "Харрикейны" по ночам просто никогда не летали.



В игре Вы можете выбрать один из трех сценариев. Тренировочный режим поможет Вам освоить методику управления воздушными силами путем имитации легкого воздушного налета. В сценарии "Блицкриг" немецкая авиация бросает все, что есть в ее распоряжении, но только в течение одного дня. И, наконец, есть сценарий затяжной кампании, рассчитанной на тридцать дней.

В игре есть также аркадные вставки, в которых Вы можете попробовать посбивать "Мессершмиты", не вставая с собственного стула. Если Вы входите в такую аркадную вставку, то успех Ваш в стрельбе по "Мессершмитам" будет рассматриваться и как успех в локальных воздушных боях и будет отражаться на ходе всей кампании.

Конечно это хорошо для любителей аркадных "стрелялок", но настоящим любителям стратегических операций лучше этот режим сразу отключить, чтобы не мешал заниматься серьезным делом. Успех в игре зависит от Вашей способности быстро принимать ответственные решения в сложных ситуациях.

Как только появятся немецкие самолеты, Вам надо поднять в воздух свои эскадрильи. После боя или продолжительного патрулирования в воздухе эскадрильи надо посадить, чтобы они могли дозаправиться и довооружиться. Если Вы упустите момент, когда их надо возвращать, они расплывутся по ближайшим аэродромам и пройдет некоторое время, пока Вы сможете их снова собрать и привести в боеготовность.

Дополнительную сложность привносят погодные условия, т.к. в плохую погоду некоторые аэродромы могут быть закрыты для приема самолетов.

Игра несложна в работе, но чем-то напоминает искусство жонглирования. Каждый знает, что подбросить в воздух одновременно несколько мячей нетрудно, но вот как сделать, чтобы все они там и оставались?

# VULCAN

"CCS" 1987



Эта игра имитирует события боевых действий в Тунисе в 1942-1943 годах и является достойным образцом среди игр этого жанра.

Управление игрой - через систему меню.



В игре есть пять сценариев, рассчитанных на то, чтобы ее продолжительность составляла от получаса до шестнадцати часов. Вы можете выбрать за кого Вы будете воевать - за немецко-итальянские войска (Axis) или за англо-американские (Allied). Играть, как и в большинстве игр можно против партнера, а можно и против компьютера.

Существует и версия для 128-го компьютера, в которой есть дополнительные возможности:

- серьезный информационный блок;
- несколько исторических сценариев, разработанных по принципу "Что было бы, если бы...";

- а также устранена необходимость подзагрузки данных после каждого сеанса игры, что необходимо в модели 48K по причине нехватки памяти. В игре есть очень приятная особенность, которой лишены настольные аналоги военных игр. Здесь можно сделать перемещение войск противника скрытым от Вас. Вы обнаруживаете вражескую часть только буквально столкнувшись с ней.

Существуют три способа перемещения своих частей.

Нормальное перемещение - moving. Нападение - assault. Перемещение походной колонной - travel.

При нападении соединение идет до достижения своей цели, а при походном марше может двигаться только по дорогам, но зато с удвоенной скоростью. Правда в этих условиях

оно очень легко уязвимо для вражеских атак.

Если Вы не хотите перемещать часть, то можете выбрать один из двух режимов:

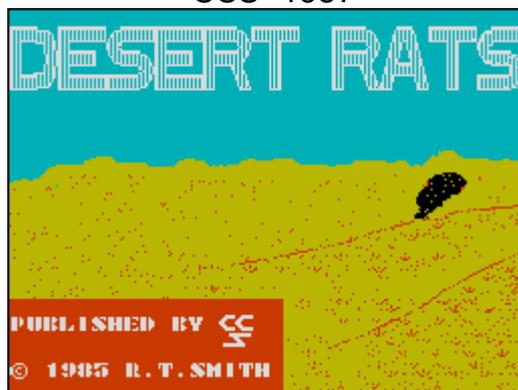
- обороняться (hold position);

- закрепляться (fortify). Рельеф местности играет в игре очень большую роль.

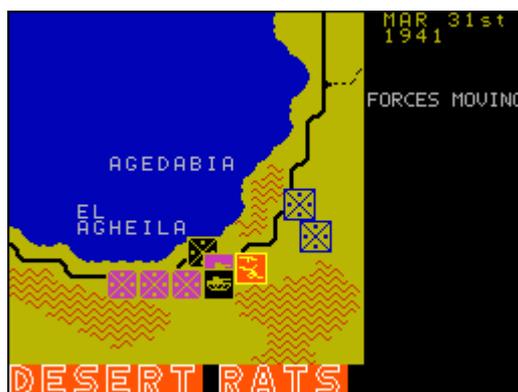
Естественно, он сказывается на скорости перемещения частей, но не только. От него во многом зависит насколько успешно пройдет атака или насколько успешно Вы атаку отразите. Успех каждого из многочисленных сражения между вошедшими в контакт противостоящими частями также зависит от многих факторов: от силы частей, от условий их снабжения и даже от погодных факторов. Снабжение своих войск - важнейший элемент Вашей стратегии. Необходимо также хорошо использовать авиацию как для нанесения ударов по врагу, так и для решения разведывательных задач. Эта игра может быть рекомендована вниманию даже самых искушенных игроков, она всем доставит удовольствие.

## DESERT RATS

"CCS" 1987



Программа "Крысы пустыни" больше подойдет тем, кто ставит игровые аспекты выше боевых. Это скорее шахматы, в которых вместо фигур - танковые полки, а вместо доски - разбитая на клетки карта ливийской пустыни. Карта очень гладко скроллирует, предоставляя Вам игровое пространство внушительных размеров. Можно отметить очень четкую графику.



В игре есть шесть сценариев, относящихся к боям в Северной Африке 1941-1942 годов, начиная с высадки дивизий Роммеля в Триполи и кончая битвой за Эль Аламейн.

Успех в битвах зависит от правильной расстановки сил и от умения обеспечить на должном уровне снабжение боевых частей.

Это не игра в режиме реального времени. В ней реализован двухшаговый метод. Сначала Вы выдаете каждой части приказ на данный ход, а потом сидите и смотрите, что из этого вышло. На ваших глазах перемещаются Ваши полки и полки противника, входят в столкновения, стирают друг друга с лица земли.

Играть можно как против партнера, так и против компьютера. Конечно поначалу у Вас почти нет шансов как-то противостоять Роммелю, но если Вы играете с другом, то может быть Вам и удастся сбросить его войска в море.

Для тех, кто знаком с такими играми, эта игра не испортит общего впечатления от жанра, а для тех, кто еще не начал, она могла бы стать прекрасным началом.

# INVASION

## "MASTERTRONIC" 1987



Если предыдущие стратегические программы, рассмотренные в данном разделе проработаны нами на уровне обзора, то нижеследующее описание является полноценной инструкцией по работе.

Эта стратегическая программа необычна тем, что боевые действия в ней ведутся между фантастическими вооруженными силами.

История конфликта примерно такова.

Колонизируя новую планету, поисковый отряд столкнулся с инопланетной цивилизацией, которая занимаясь тем же, установила здесь станцию управления погодой и занимается тем, что создает условия жизни совершенно неприемлемые для землян. Ваша задача - разгромить войска противника, разрушить эту станцию и, по возможности, живым и невредимым вернуться домой.

Основные факторы в игре - стратегия, скорость и тактика. На каждый Ваш ход пришельцы ответят так, чтобы помешать Вам разрушить их станцию управления погодой и выиграть сражение. Вам необходимо тактически правильно спланировать кампанию и все время быть начеку. Противник только и ждет Вашей ошибки и зачастую второго шанса у Вас не будет.

Под Вашим управлением находятся различные боевые части. Используйте их рационально, поскольку у противника тоже есть соответствующие аналоги и вскоре они их проявят.

### **Ваши боевые части**

#### **1. Радары.**

Они управляются клавишей R и являются как бы Вашими "глазами". Карта на экране изображает то, что находится вокруг радара. Радаров у Вас четыре и, нажимая клавишу "R", можно между ними переключаться.

#### **2. Минные части.**

Они представляют из себя дистанционно управляемые ядерные взрывные устройства. Взрывное устройство, взорванное рядом с космическим кораблем или заводом полностью выводят их из строя. Поражаются и все находящиеся в смежных полях боевые части. Но для того, чтобы взорвать вражескую станцию управления погодой, необходимо взорвать два рядом стоящих взрывных устройства.

#### **3. Воздушная пехота.**

Это очень быстрые и маневренные соединения, имеющие среднюю защиту и огневую мощь. Оно может атаковать только смежные с ним части, но может преодолевать кислотные озера и минные поля.

#### **4. Парящая броня.**

Эти части имеют примерно те же характеристики, что и воздушная пехота, но у них радиус действия больше.

#### **5. Страйдеры.**

Это медленные части со средней степенью защиты. Они могут атаковать только

смежные с ними позиции.

## Боевые части инопланетян

### 1. Тяжеловооруженные части.

Это медленно перемещающиеся войска, но с тяжелым вооружением, мощной броней и довольно большим радиусом действия.

### 2. Части быстрого реагирования.

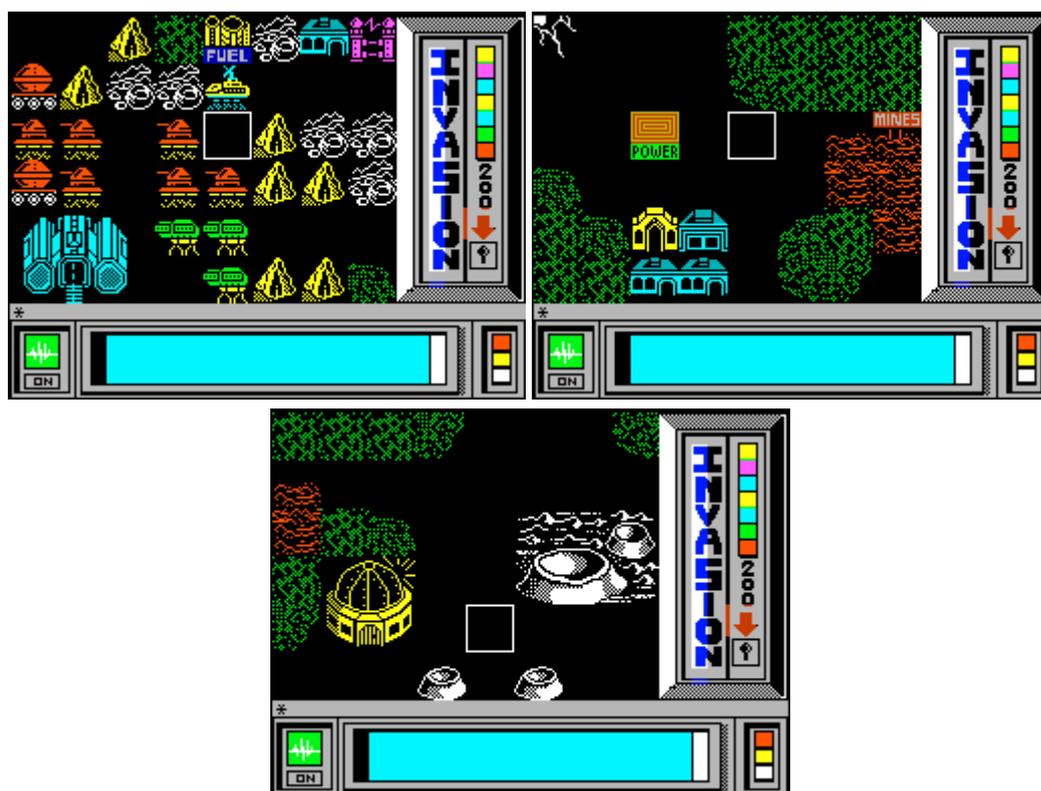
Быстрые воздушные части с низкой защищенностью и малой огневой мощью. Они имеют возможность преодолевать кислотные озера и минные поля.

### 3. Дроиды.

Аналог земных частей "Парящая броня".

## Игра

Главная цель - уничтожить станцию управления погодой. Для этого Вам надо взорвать два из трех Ваших взрывных устройств в прилегающих к ней позициях. Если Вы сумеете это сделать, то вам надо как можно быстрее покинуть планету на одном из двух возможных космических кораблей, прихватив с собой как можно больше своих частей. Если Вы оставите второй корабль неразрушенным при оставшихся в целости вражеских войсках, то инопланетяне стартуют вслед за Вами и собьют вас, поэтому надо либо уничтожить все вражеские части, либо взорвать вражеский корабль последним оставшимся взрывным устройством.



Инопланетяне могут уничтожить космический корабль, поместив на соседние поля три боевых части и дав им команду на самоликвидацию. Если оба корабля и станция управления погодой уже уничтожены, то Вам остается только закончить игру или вести войну до полного уничтожения.

## Дополнительные возможности

1. Боевые части могут скрываться на заводах и фабриках и вести огонь оттуда.
2. Части могут частично восстанавливать свои повреждения, если их поместить на поле с топливным устройством.
3. Войска могут погружаться на космический корабль только с полей, смежных с дверьми.
4. При взрыве минного устройства уничтожаются все смежные с ним части - не только

вражеские, но и Ваши. При поражении вражеским огнем они также взрываются.

### **Полезные советы**

1. Защищайте свои минные части. Без них Вы не сможете победить. Особенно тщательно оберегайте их от дальнего огня противника.
2. Следите за сообщениями телепринтера.
3. Постоянно следите за собственными частями. Небрежность может превратить их в артиллерийскую мишень.
4. Всегда имейте в виду основную цель.

### **Счет**

В конце игры Вы получите итоговую таблицу. Наибольшие очки даются за уничтожение станции управления погодой или вражеского корабля противника. Те же очки можно получить и за уничтожение собственного корабля, если Вам удастся улететь на вражеском.

Начисляются очки также и за каждую уничтоженную вражескую часть и вывезенную на корабле свою.

### **Управление игрой**

Если поместить курсор на свою часть и нажать "огонь", то в специальном окне на экране появится сообщение о ее состоянии. Если текущее положение Вас устраивает, нажмите "огонь" еще раз, в противном случае переместите курсор в новое место и вновь нажмите "огонь". После этого курсор автоматически укажет на новую часть, хотя Вы можете ее и пропустить нажатием клавиши N.

Ваши части защищаются от огня самостоятельно и сами выбирают цель для стрельбы. Вы не можете выбирать цели.

### **Расписание клавиатуры**

- N - переход к следующей части;
- R - переход на следующий радар;
- F - переключатель "Пауза";
- V - выгрузка игры на ленту;
- E - подрыв взрывного устройства;
- T - запуск космического корабля, если на нем установлен курсор.

# ТЕХНИЧЕСКОЕ ПРИЛОЖЕНИЕ

ЭТОТ РАЗДЕЛ ВЕДЕТ НАУЧНО-ТЕХНИЧЕСКИЙ КООПЕРАТИВ "ПЛЮС",  
СПЕЦИАЛИЗИРУЮЩИЙСЯ В ВОПРОСАХ АППАРАТНОГО ОБЕСПЕЧЕНИЯ СИНКЛЕР-  
СОВМЕСТИМЫХ КОМПЬЮТЕРОВ И СХЕМОТЕХНИЧЕСКИХ РЕШЕНИЙ ПЕРИФЕРИЙНЫХ  
УСТРОЙСТВ  
137566, МОСКВА И-566, НТК "ПЛЮС"

## ВНИМАНИЕ

В программе поддержки ZX-Модема обнаружены опечатки. Первым об их обнаружении сообщил товарищ Ключник из Полтавы.

Строка 46:

Напечатано: LET a(1) = FN(b\$)

Должно быть: LET a(1) = FN A(b\$)

Строка 202:

Напечатано:

F940 01 32 8E FC 3E 02 32 A3 09

Должно быть:

F940 01 32 8E FC 3E 02 32 A3 0B

Строка 210:

Напечатано:

F980 0B F9 AF C9 CD AF F9 3E AB

Должно быть:

F980 0B F9 AF C9 CD AF F9 3E A8

Строка 238:

Напечатано:

FA60 0A 2B CD 18 1C F5 11 80 13

Должно быть:

FA60 0A 28 CD 18 1C F5 11 80 13

## БУФЕРИЗАЦИЯ

Многие наши читатели являются убежденными поклонниками Синклер-совместимых компьютеров. Хорошо понимая и осознавая ограниченность этих машин, они также понимают, что в настоящее время нет никакой другой разумной альтернативы домашнему компьютерному творчеству. В нем они находят источник постоянного самосовершенствования и самоутверждения.

Совершенствуя себя, развиваясь и образовываясь, многие задаются вопросами совершенствования и своего компьютера, ведь как бы хорош он ни был, пытливый человек всегда найдет как сделать что-то лучше, иначе, не так как у всех. Одному из направлений возможного совершенствования Вашей машины и посвящена сегодняшняя статья "Технического приложения".

Фирменные и всевозможные варианты схем самодельных Спектрумов за редким исключением имеют общий существенный недостаток. Этот недостаток заключается в том, что микропроцессор Z80 - сердце Спектрума - не защищен никакими шинными формирователями, усилителями и другими буферными элементами. В фирменном компьютере это обусловлено малым количеством микросхем, подключенных к шинам микропроцессора (всего около 20), но и там любая фирменная периферия, подключаемая к краевому разъему платы, вынуждена на входе иметь буферные элементы.

Самодельные компьютеры и без того используют предельные по нагрузочной способности режимы микропроцессора, а если встает вопрос о дополнительной периферии, то тут уже без усилителей шин не обойтись.

Кстати, с перегрузкой шин микропроцессора в самодельных компьютерах очень часто связаны всевозможные неполадки. Иногда не удается запустить правильно собранный из исправных элементов компьютер из-за разброса параметров даже для исправных микросхем. Например, Вы вынимаете микропроцессор из панельки и втыкаете его в исправный компьютер, он оказывается в порядке, а на Вашей плате - "не тянет".

Иногда компьютер работает, но часто "сбоит", сбрасывается, зависает, то ли после прогрева, то ли от помех по сети, то при нажатии клавиш. (Кстати, если работа компьютера нарушается при нажатии клавиш, то по ним можно вычислить перегруженную шину и буферизировать хотя бы одну ее). В общем, весь этот "полтергейст" может быть связан с превышением предельного уровня нагрузки микропроцессора. Разработчики фирмы Синклер руководствовались, видимо, соображениями дешевизны, когда не заботились о периферии.

#### СПРАВКА ИНФОРКОМА

Разработчики фирмы "СИНКЛЕР" руководствовались соображениями эффективности технологии производства, ставя перед собой задачу всемерного повышения надежности главной платы в автоматизированном производстве. Поскольку надежность микросхем обеспечивалась автоматизированным входным контролем, надежность готовой платы зависела в основном от количества паяных и тем более непаяных контактных соединений. Были приняты все меры, чтобы свести их к минимуму. В этом случае ответственность за надежность работы машины совместно с периферией перекладывалась на плечи разработчика периферии.

Хотя конечно же такой подход дал возможность значительно снизить цены на готовую машину.

Интересно отметить, что в результате стоимость самой примитивной периферии стала сравнимой со стоимостью компьютера.

Например, стоимость нехитрого интерфейса джойстика на одной микросхеме составляла до 20% стоимости компьютера, но его делал не Синклер. Зато его компьютеры оставались самыми дешевыми в мире.

Пользователи же самодельных Спектрумов, особенно те, кто не слишком озабочен их серийным производством, могут себе позволить "обвесить" компьютер буферными элементами и затем вешать на него всю мыслимую периферию, схемы которой сыплются на головы пользователя, в том числе и стараниями специалистов из НТК "ПЛЮС".

Кстати сказать, когда много лет назад, в 1984 году разрабатывался один из первых вариантов самодельных Спектрумов, впоследствии окрещенный "Московским" вариантом, то в схему и плату компьютера были заложены буфера на K555АП6. Эти буфера можно по желанию устанавливать при работе с периферией или обходиться без них, заменив их перемычками. Они не окружают микропроцессор, а работают только на внешний разъем.

Этот подход мы и рассмотрим в нашей статье. Он подходит нам потому, что не потребует вмешательства в плату и схему компьютера и обращается лишь к внешнему разъему. Правда при этом внутренние шины компьютера не будут буферизованы.

Для буферизации внутренних шин компьютера можно было бы посоветовать следующий способ. Удалить микропроцессор и заменить его на небольшую плату, на которой установлен как сам микропроцессор так и буферные элементы, например 4 штуки K555АП6. Если у Вас запаяна панелька под микропроцессором, то такой эксперимент провести будет не очень сложно. Правда есть одно проверенное правило - лучшее враг хорошего. Поэтому если Ваш компьютер работает вполне устойчиво, не сбоят и вообще не вызывает устойчивых подозрений своей работой, можно обойтись без "хирургического вмешательства", а для подключения разветвленной периферии воспользоваться внешней буферизацией.

Схемы подключения буферных элементов для конкретного микропроцессора достаточно стандартны и изобретать велосипед мы не стали, а взяли эти схемы из зарубежной печати. При этом мы исходили из доступной отечественной базы.

Шины микропроцессора делятся на входные, выходные и двунаправленные. Проще всего с входными шинами, такими как /WAIT<sup>1</sup>, /NMI, /INT, /RESET и т.д. Их вообще не нужно буферизировать.

Выходные сигналы, требующие усиления - это 16 линий шины адреса - A0-A15, а также сигналы управления, такие как /RD, /WR, /IORQ, /MREQ и т.п.

Немного сложнее обстоит дело с двунаправленной шиной данных D0-D7, потому что для этого требуется применение двунаправленных микросхем и управление переключением направления передачи данных.

На рис.1 и 2 приведены схемы буферов шины адреса и управления, выполненные на распространенных микросхемах K589HP12. Здесь и далее обозначения сигналов со штрихом относятся к усиленным сигналам. Номиналы резисторов не критичны и могут быть в пределах 10-20 Ком.

Более сложная схема буфера шины данных приведена на рис. 3.

Правда, если заменить доступные микросхемы K589HP16 на более дефицитные, например, на восьмиразрядный буфер K555АП6, то схема будет по крайней мере выглядеть более простой, да и количество паек уменьшится. Сложность управления буферами объясняется стремлением достичь полной универсальности в их работе.

Здесь приняты меры к тому, чтобы не было конкуренции на шине данных "внутри" компьютера между "внутренними" источниками и буфером, включенным на ввод данных по сигналу "на запись" (-WR без инверсии) и вовремя не переведенным в высокоимпедансное состояние.

Для этого служит сборка 8И-НЕ, выполненная на K555ЛА2. Сборка "отлавливает" те моменты, когда внешние устройства (устройства находящиеся за буфером) не выбраны, т.е. находятся в третьем состоянии и отключает буфера от шины. Кроме того, этот "диспетчер шины данных" отключает /CS на буферах еще и тогда, когда от внешних устройств поступает байт FF (все единицы, ни одного нуля на шине данных), что вообще говоря неправильно, но в этом случае роль байта FF выполняет высокоимпедансный выход буфера и резисторы R1-R8. Кстати, эти резисторы можно не ставить потому, что они как правило уже есть в компьютере. Чтобы это проверить, нужно прочитать данные из незадействованного адреса внешнего порта, и если они равны FF(255), то это говорит о подтянутости шины данных к плюсу.

Далее по схеме на рис.3. Если переключатель E запаян, то буфер полностью универсален. При удалении переключателя буфер становится зависим от сигнала /IORQ. Т.е. буфер будет работать только с внешними устройствами, адресованными как порты ввода/вывода. От переключателей ABCD зависит то, какие сигналы используются в управлении буфером - непосредственно от микропроцессора (в таком случае желательно в качестве инверторов применить микросхемы из серии ТТЛШ для уменьшения нагрузки на микропроцессор) или взятые после буфера сигналы управления.

Если Вы используете буфер шины управления, то есть смысл использовать усиленные сигналы /RD и /WR. Опыт показывает, что возникающая при буферировании задержка около 30 Нс в данном случае не существенна.

Оставшиеся незадействованными два элемента 2 ИЛИ-НЕ можно использовать для выработки сигнала очень полезного при работе с периферией, связь с которой устанавливается по прерыванию. Этот сигнал INTA (аналогичен сигналу INTE микропроцессора KP580ИК80 (I8080)) и служит для подтверждения полученного прерывания. У микропроцессора Z80 нет такого сигнала, но сигналы /M1 и /IORQ свидетельствуют о выборке вектора прерывания с шины данных, поэтому их комбинация может служить в этом качестве.

На рис.4 представлена схема дешифратора адресов портов ввода-вывода. Это также необходимая вещь, если Вы присоедините к компьютеру периферию. Дело в том, что в фирменном и почти во всех самодельных компьютерах Спектрум очень грубо дешифрируется адресное пространство ввода/вывода, а именно - все четные адреса (при

---

<sup>1</sup> Инверсные сигналы обозначены префиксом / (Прим. NUK)

A0=0) задействованы под внутренний порт с адресом 254, который "занимается" бордюром экрана, магнитофоном, звуком и клавиатурой. Кроме того, по концепциям фирмы под периферию, выпускающуюся для Спектрума заняты адресные линии A5 и A6.

Некоторые пользователи жалуются на неработающую периферию, в том числе и сделанную по нашим схемам (НТК "ПЛЮС"). Часто причина этого в том, что в схеме уже стоит какой-либо грубо адресованный порт, и в результате возникают конфликты на шине, т.е. выбирается одновременно более одного источника.

Если Вы хотите избежать этих трудностей при Ваших экспериментах, используйте более точный дешифратор адресов. Дешифратор, изображенный на схеме 4, рассчитан на 8 периферийных устройств и при желании может быть расширен дальше. Следующий дешифратор K555ИД7 можно присоединить, используя в качестве сигналов выбора вывод 8 микросхемы DD3 и адресную линию A10 и т.п.

Адресные линии A5, A6 не используются, чтобы не конкурировать с наработанными периферийными устройствами с "кемпстоновской" адресацией.

На рис.5 приведена для примера буферная схема для шин адреса, данных и управления, выполненная на микросхемах KP589BA86. Адресную шину можно буферизировать полностью, добавив еще один элемент аналогично DD1.

Буфер в шине управления на DD3 тоже не нуждается в комментариях. Особенностью этой схемы является управление переключением направления буфера данных на DD2.

В связи с тем, что направление данных в микросхемах KP589BA86 нельзя изменять во время их активного состояния, как это делалось в K589АП16, в схему введен RS триггер.

Как Вы могли заметить, данная схема буферизации не универсальна, она рассчитана на работу с портами ввода/вывода и управляется сигналами выбора /CS тех внешних устройств, которые подключены к вашему компьютеру.

Если таких устройств несколько, то для объединения их сигналов выбора /CS можно применить сборку, например 8И-НЕ, как показано на рисунке. Теперь при адресации к любому из внешних портов любой из сигналов /CS переведет буфер шины данных, выполненный на микросхеме DD2 из высокоимпедансного состояния в активное. Направление передачи данных через буфер в это время удерживается RS триггером, включенным сигналом /WR или /RD в одно из двух состояний. Заметим, что так как на сборку сигналов CS заводятся сигналы выбора только от устройств, подключенных после буфера, то при выборе какого-либо внутреннего порта буфер будет находиться в высокоимпедансном состоянии и таким образом не возникает конфликта на шине данных.

Приведенные здесь схемы вполне работоспособны, но мы ставили своей целью осветить общие подходы к буферизации шин микропроцессора. Пользователи Спектрума могут использовать эти подходы при разработке собственных схем внешних устройств. Применяемые Вами в качестве буферных элементов микросхемы могут быть очень разнообразными, в зависимости от возможности их приобретения, потребляемого тока, размеров и т.д.

Кроме упомянутых в статье микросхем, в качестве буферов часто применяются K155ЛН1, 2, 3, 4, 5 - буфера без выводов разрешения; K155ЛН6, K155ЛП7, ЛП8, ЛП10 - буфера с выводом разрешения; K531АП3, АП4, K155ИП6, ИП7, K555АП5, АП6 - двунаправленные буфера с выводами разрешения и т.д. и т.п.

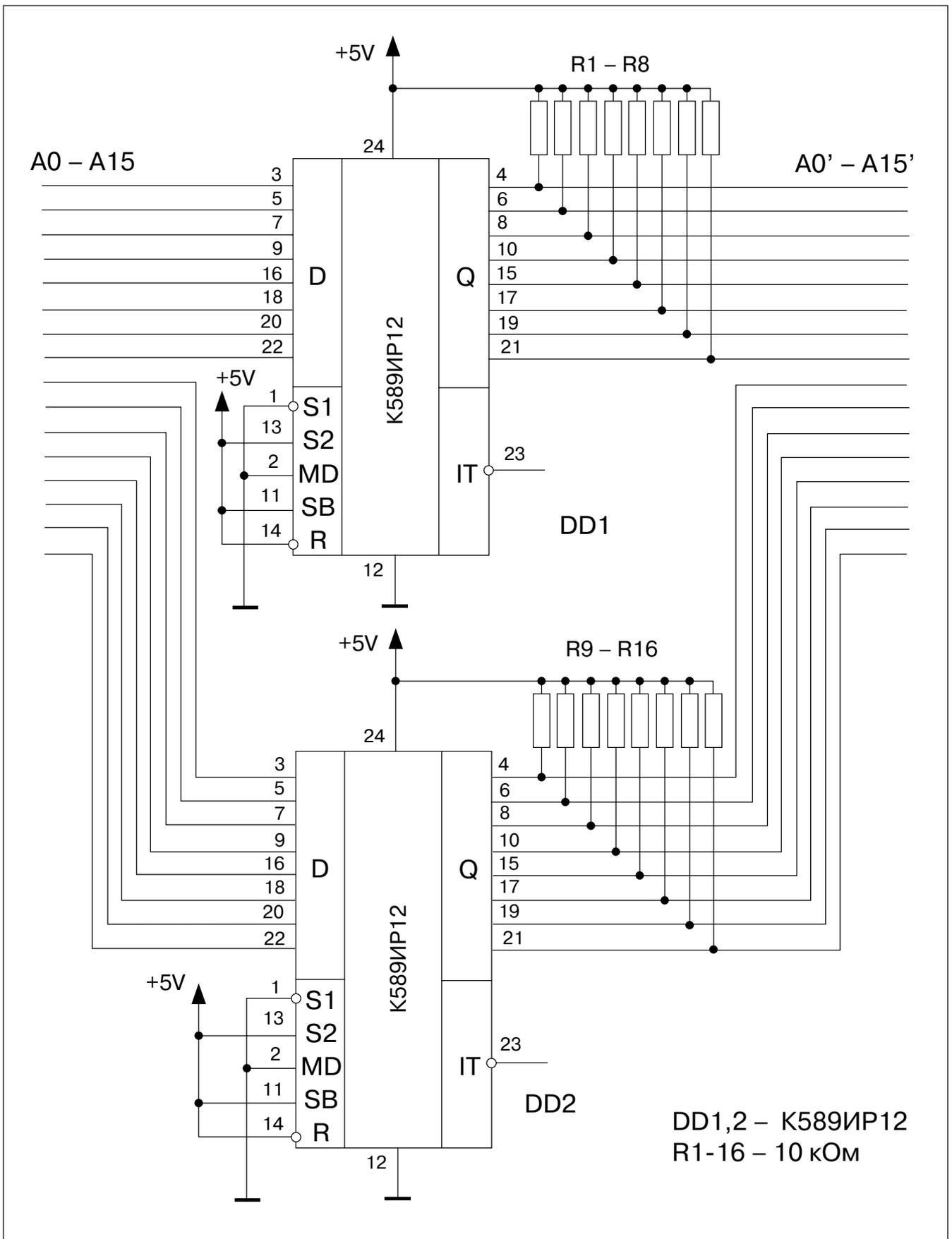


Рис. 1. Буфер шины адреса A0 - A15

В заключение рассмотрим одну важную особенность фирменных компьютеров SPECTRUM, которая в последние годы стала иногда использоваться разработчиками фирменных программ для Спектрума, что привело к проявлению неполной совместимости самодельных компьютеров с оригиналом. Речь идет об устройстве шины данных компьютера. Мы уже обсудили тот факт, что при чтении данных из несуществующего порта (или из неиспользуемого адреса порта), мы получаем данные FFH (или 255D). Это связано с тем, что тогда, когда не выбрано ни одно устройство, подключенное к шине данных, то все они находятся в высокоимпедансном состоянии. Но шина данных через резисторы 10-20 К подключена к плюсу источника питания. Поэтому все разряды шины данных принимают состояние логической единицы, что в результате дает данные FF(255).

Т.е.

```
10 PRINT IN 1  
20 GO TO 10
```

даст результат:

```
255  
255
```

```
...
```

```
255
```

```
SCROLL?
```

Однако если эту программку запустить на фирменном Спектруме сразу же после включения или сброса, то мы увидим, что среди результата 255 иногда будет "проскальзывать" результат 56.

Оказалось, что это число зависит от атрибутов экрана. Например, если задать цвет фона (PAPER) черный, а цвет символов (INK) - белый, то это число, иногда считываемое в программе, будет равно 7, т. е. это именно байт атрибутов, считываемый из области 768 байтов атрибутов, которая расположена после области 6144 байтов графической информации экрана (Мы исходим из того, что карта памяти компьютера Вам хорошо знакома, хотя бы по книге ИНФОРКОМА "Большие Возможности Вашего СПЕКТРУМа").

Остается выяснить, каким образом, считывая данные из порта с адресом 1 (этот адрес взят для примера, можно взять любой адрес порта, не используемый в данной конфигурации), мы получаем в результате данные из области атрибутов экрана.

Для этого нужно рассмотреть устройство внутренней шины данных фирменного компьютера Спектрум. У фирменного Спектрума непосредственно к шине данных подключены: микропроцессор, ПЗУ, поле памяти 16К (нижнее), дополнительное поле памяти 32К (верхнее) и внутренние регистры микросхемы ULA - программируемой логической матрицы, выполняющей в Спектруме функции контроллера дисплея, генератора тактовой частоты процессора и порта ввода/вывода с адресом 254 (а на самом деле с любым четным адресом), для ввода и вывода данных магнитофона, вывода звука, ввода данных от клавиатуры, вывода цвета бордюра.

Причем ULA работает автономно совместно с областью памяти 16К, в которой как раз и выделено место для памяти экрана 6912 байт (растровая графика плюс атрибуты). Периодически, в соответствии с частотой кадров и строк телевизионной развертки, ULA вырабатывает адреса области экрана и считывает 32 раза в течение каждой строки по два байта - один байт из области растровой графики и один из области атрибутов экрана, так как микропроцессор также имеет доступ к этой области памяти, то, если не принять специальные меры, может возникнуть конфликт на шине данных.

Чтобы этого не произошло, в фирменном компьютере приняты следующие меры. При одновременном обращении к полю памяти 16К со стороны ULA и микропроцессора, ULA останавливает тактовую последовательность, поступающую к микропроцессору Z80 и, таким образом, Z80 "ждет" возможности работать дальше, т.е. ULA имеет приоритет на работу с этими микросхемами памяти. В то же время, независимо от работы ULA с памятью экрана, микропроцессор может продолжать работать с дополнительной памятью 32К, с ПЗУ и внешними портами.

Чтобы при этом не возникал конфликт на шине данных, шина данных разделена. ULA и поле памяти 16К отделены от остальных пользователей шины данных резисторами 410 Ом.

При этом ULA и микропроцессор по отдельности пользуются каждый своим "куском" шины данных, не слишком мешая друг другу. Теперь вернемся к работе нашей программы.

Когда микропроцессор считывает данные из порта 1, то состояние шины данных на участке, "принадлежащем" микропроцессору, определяется резисторами 8.2 Ком, которыми линии подтянуты к плюсу источника питания. В то же время, к микропроцессору через резисторы 470К поступают данные, считываемые ULA из памяти экрана при его регенерации и при совпадении моментов считывания эти данные воспринимаются микропроцессором, как данные из порта 1. Т.е. в фирменном компьютере при считывании с внешнего порта с неиспользуемым адресом можно получить данные из области атрибутов экрана.

Это обстоятельство используется в некоторых фирменных программах возможно для более точной синхронизации события программы и изображения на экране, во всяком случае, то обстоятельство, что в самодельном компьютере шина данных полностью развязана и на ней всегда присутствует лог."1" по всем линиям при невыбранных устройствах, является причиной несовместимости некоторых программ.

Например, здесь нужно упомянуть третий выпуск ZX-РЕВЮ (раздел "ФОРУМ"), в котором наш читатель рекомендовал как сделать работоспособной программу ARKANOID-1. В самодельных компьютерах с отдельным полем памяти для достижения подобного результата можно было бы еще посоветовать соединить входы и выходы микросхем памяти поля 16К с помощью резисторов 470 Ом.

Ввиду большого разнообразия разработок самодельных компьютеров трудно дать рекомендации всем пользователям Спектрума, но мы думаем, что своими публикациями мы хотя бы чуть-чуть поможем Вам разобраться в некоторых тонкостях работы Вашего компьютера.

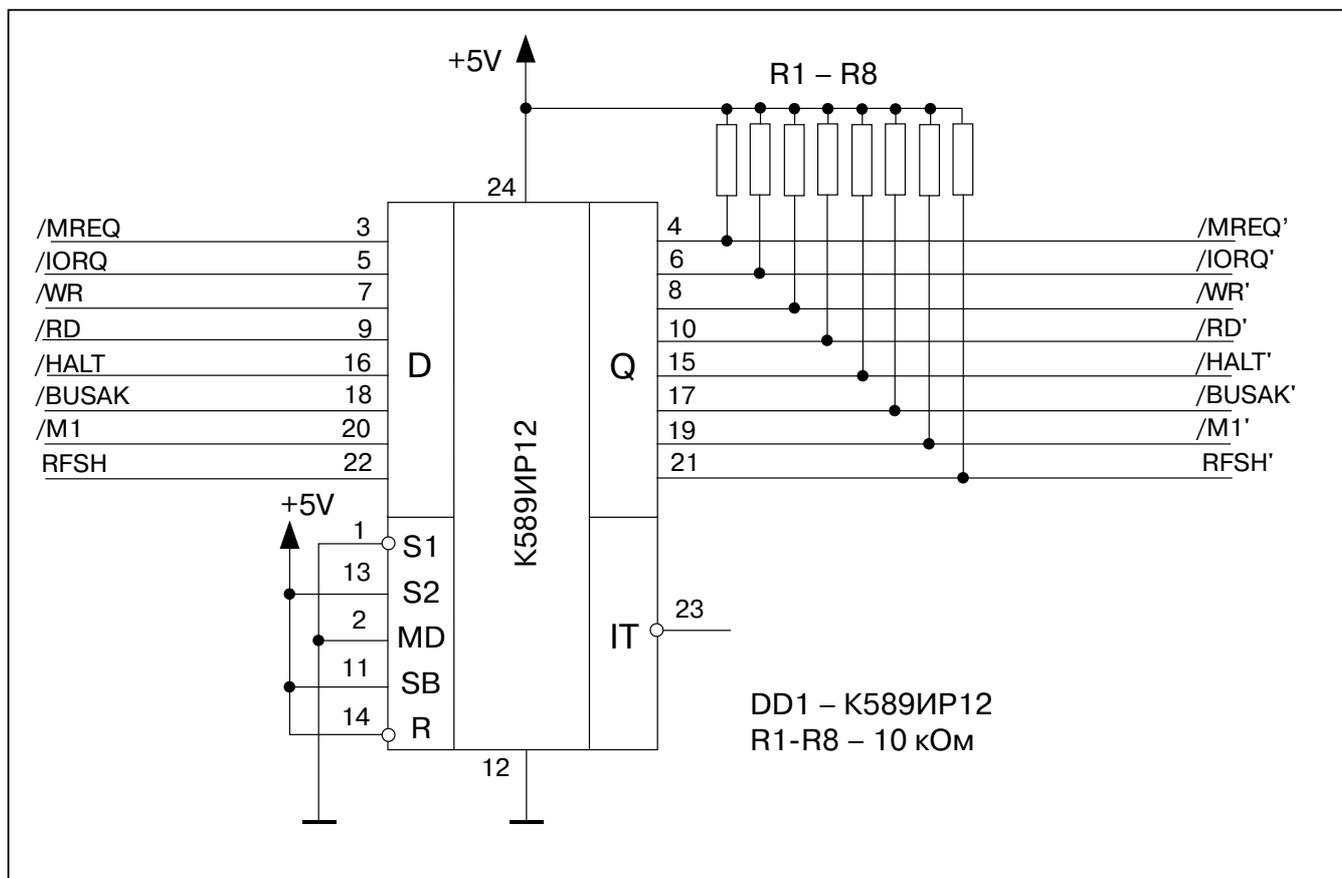


Рис.2. Буфер шины управления.

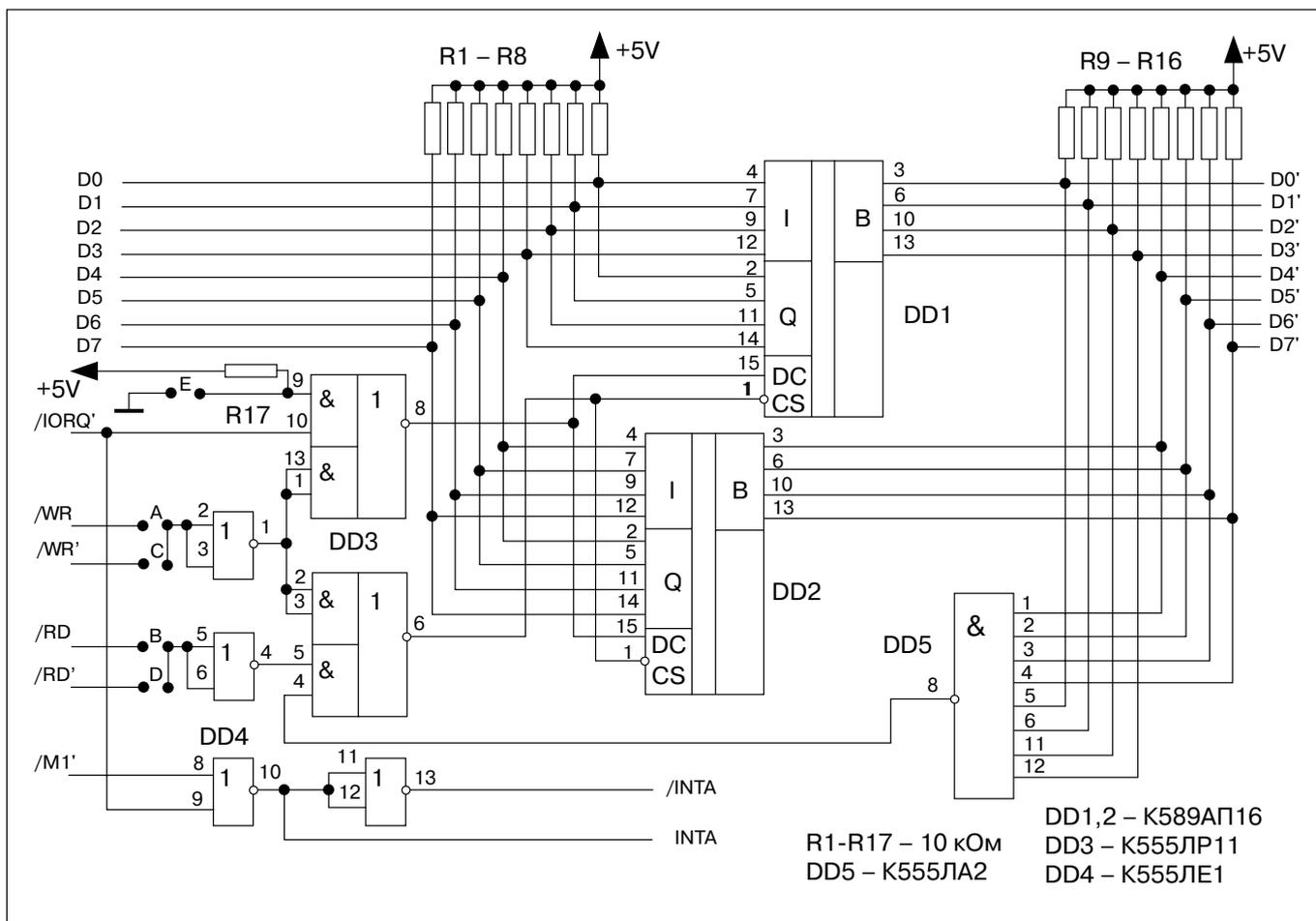


Рис.3. Буфер шины данных.

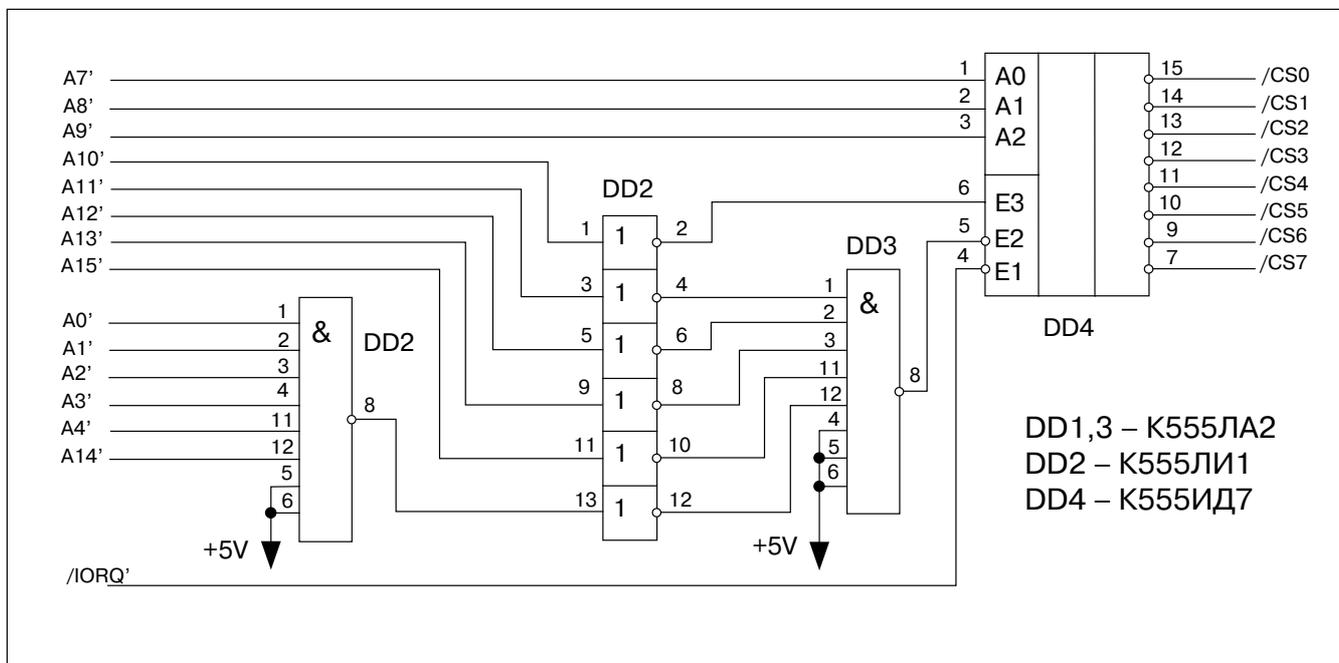


Рис. 4 Дешифратор адресного пространства ввода/вывода

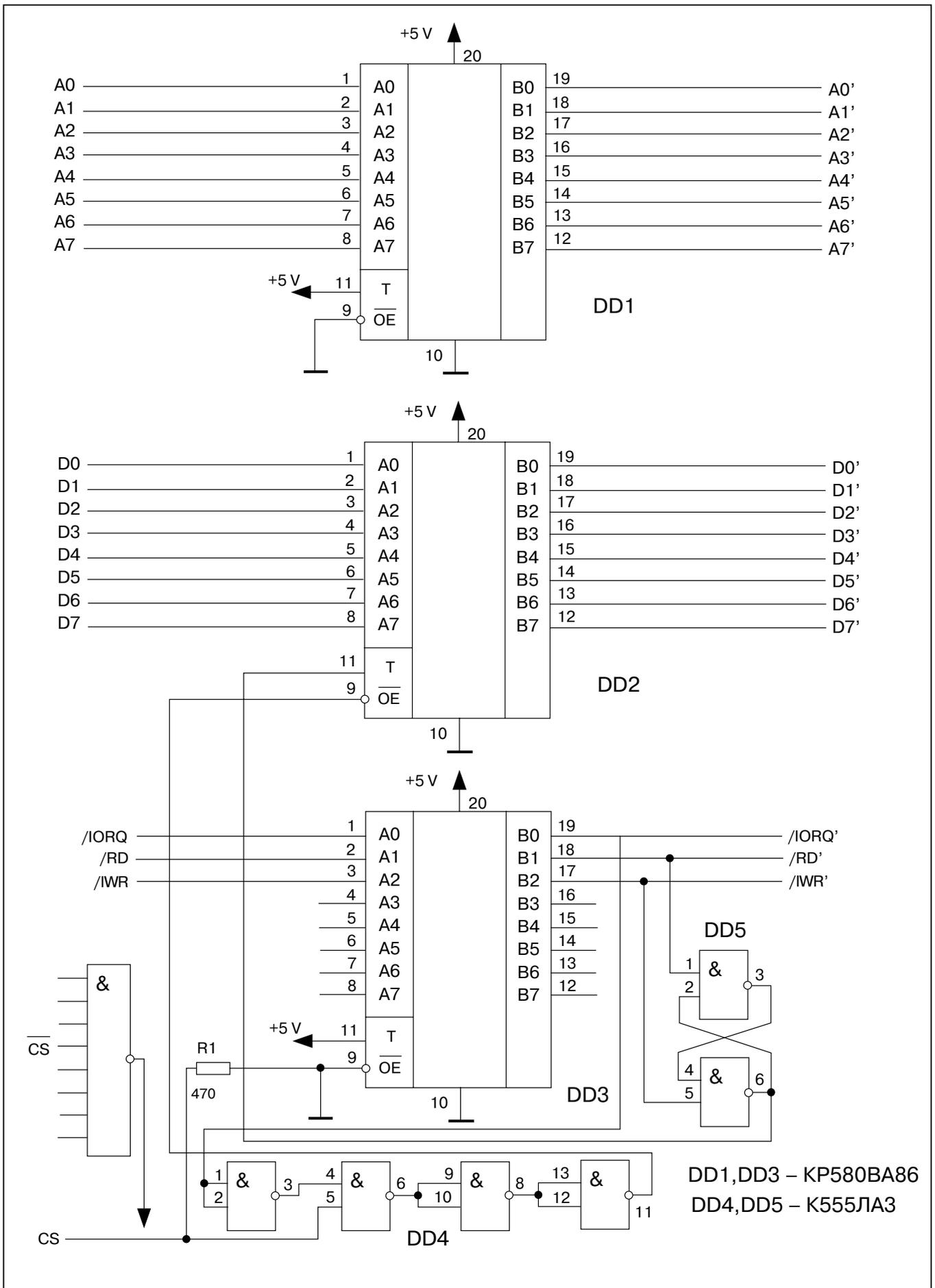


Рис.5. Универсальный буфер

## ZX-РЕВЮ РЕКЛАМНОЕ ПРИЛОЖЕНИЕ АПРЕЛЬ-МАЙ 1991

НТК "ИНФОРКОМ" не может принимать на себя никакую ответственность за точность и достоверность сведений, содержащихся в объявлениях читателей.

Имею 1000 программ для ZX-Spectrum.  
Предлагаю обмен. Всем желающим вышлю каталог.

614002 Пермь, ул.Фонтанная 9-105  
Носкову Владимиру Анатольевичу.

Предлагаю:  
Запись на ГМД размер 5.25" различных  
игровых и прикладных программ.  
ГМД - мои или заказчика. Каталог вышлю.

326520, Херсонская обл, г.Цюрипинск,  
пер.Зеленый- 2  
Алексеев А.А.

Обменяю системные, прикладные,  
сервисные, учебные и игровые  
программы.

Телефон в Бердске: 5-32-62. Адрес:633190  
Новосибирская обл, г.Бердск,  
ул.К.Маркса, 66 кв.86 Кобылинский Юрий.

Ищу принтер с интерфейсом для ПК  
"Спектрум".

334264 Крым, г.Гурзуф, ул.Соловьева 6  
кв.33 Сорокину В.В.

Предлагаю в обмен на кассеты,  
дискеты и радиодетали более 800  
системных и игровых программ к ПК типа  
"ZX SPECTRUM". Запись на кассетах  
заказчика.

Обращаться по адресу:  
460048 Оренбург, пр.Победы д.144"А"  
корп.2 к.572 Щедринов.

Предлагаю ППЗУ (573РФ5) с  
тестами для проверки ОЗУ и отладки плат  
"Синклер-48К" (30 руб.) и "Синклер-128К"  
(35 руб.).

Обращаться по адресу: 617740  
г.Чайковский, ул.Кабалевского 28-75  
Дубинец Игорь Николаевич

Всем владельцам 43-х корпусного  
варианта ПК "Spectrum" (вариант Зонва)  
предлагаю несложную аппаратную  
доработку. Она превратит его из обычной  
игрушки в очень нужный и серьезный  
прибор.

Доработка позволяет:

1. Подключить к компьютеру помимо  
джойстика, "естественную" периферию -  
принтер и контроллер дисководов.

2. Подключить любые разработки  
НТК "ПЛЮС" (программатор, все типы  
джойстиков, световое перо, интерфейсы  
RS232 и параллельный для принтера.)

3. Кроме того, в распоряжении  
пользователя останется еще 32700 портов  
ввода-вывода, т.е. неограниченное  
количество аппаратуры, которая может  
быть подключена к шине данных  
компьютера.

Все вышеперечисленное в  
оригинале невозможно. Стоимость схемы  
и описания (включая пересылку) - 27  
рублей.

Оплата наложенным платежом.

Гарантирую четкую и безотказную  
работу всем приславшим заказ.

Письма-заявки присылать по адресу:  
460040 Оренбург, пр.Гагарина д.23, кв.24  
Чалову Юрию Анатольевичу.

Куплю принтер к ПК "SPECTRUM".

Обращаться по адресу: 338042 Донецкая  
об. г.Горловка, ул.Бессонова 33/310  
Павловскому Г.В. (тел.3-96-08)

Запишу программы к ПК "Спектрум".  
Имею около 400 фирменных программ.  
Качество гарантирую. Вышлю каталог.  
Возможен обмен программами.

Адрес: 330001, г.Запорожье-1, а/я  
160. Юра.

Предлагаю на обмен программы 87-89 г.г. для компьютера ZX-Spectrum.  
Куплю джойстики - аналог VS-125.  
Адрес: 678300 ЯСССР, п.Сангар, ул.Алексеева 17-9. Нечетайленко А.В.

Вышлю наложенным платежом печатные платы ПК "Синклер".  
Варианты: ленинградский, московский, "Пентагон"  
- 90 руб. за шт.  
Микросхемы, проверенные в ПК КР565РУ5  
-180 руб. за 8 шт.  
Документация к печатным платам 10-15 руб.  
Обращаться по адресу: 606400 Нижегородская об. г.Балахна-2, до востребования Пухову Юрию Викторовичу.

Предлагаю программы на кассетах SONY, BASF, TDK по 40 руб. за кассету с 20-ю программами. Кассеты вышлю наложенным платежом. Каталог вышлю бесплатно.  
Обращаться по адресу: 364052 ЧИССР. г.Грозный, гор-к Иванова дом 106 кв.10 Кирилину Андрею Николаевичу.

Предлагаю ежемесячное издание "48Кбайт". Стоимость годовой подписки 60 рублей, объем издания 22-30 страниц. Запись игровых и системных программ, (в коллекции около 2 тысяч программ), большой выбор описаний игровых и системных программ, в т.ч. и авторских. Каталоги и рекламные листки высылаются бесплатно.  
Обращаться по адресу: 164413. Архангельская об, п/о Катунино, ул.Катунина 4-20. Костюченко Ю.К.

Предлагаю программы "BIORITMICS", "TESTS", "CHIROMANT". Авторские разработки. Объем каждой около 40К.  
Адрес: 640020 г.Курган, ул.Климова д.56, кв.17. Сорокин Алексей Вячеславович.

Продаю компьютеры "ZX-SPECTRUM" (вариант "Ленинград-2" с кодером: RGB, BRIGHT, BEEP СЕКАМ-6 канал) с техописанием по ценам ниже рыночных.  
Гарантийный срок - 3 года.  
Продаю программы, сгруппированные по тематике, на 20 кассетах МК-60.  
256300 Киевская обл. г.Борисполь, ул.Котовского д.3. кв.77  
Кысиль Григорий Николаевич.

Продам Спектрум-128 с любой периферией (монитор, дисковод, принтер) и программным обеспечением. Форма оплаты любая.  
Обращаться по адресу: 117463, Москва, В-463, а/я 97 Шаликов В.В.

С высоким качеством, оперативно и дешево запишу на кассету заказчика или свою описания игровых и системных программ и программы.  
Обращаться по адресу: 626440 Нижневартовск. ул.Чапаева 85"Б"-188. Карша С.Р.

Продаю, покупаю и обмениваюсь программами ZX SPECTRUM. Обмен и продажу обеспечиваю своими кассетами.  
Обращаться по адресу: 245110 Сумская об, г.Шостка, ул.Франко, 24. Луговому Владимиру Викторовичу.

Продам ИС серий: 142; 155; 555; 561; 580; 1810. Налаженные платы "ZX-Spectrum" (6 штук). Чистящие дискеты (15 штук).  
Обращаться по адресу: 634001 Томск, пер.Островского 8-60. Тристан С.В.

Предлагаю обмен программами владельцам "Спектрума" и других совместимых ПК, проживающим на Дальнем Востоке и в Приморье.  
Адрес:692900, г.Находка-18, ул.Постышева д.49-22 Евтушенко В.А.

## НАШ КОНКУРС

Сегодня мы подводим итоги ранее объявленного конкурса на лучшую тематическую подборку игровых программ.

На конкурс мы получили 53 сборника от 30 читателей. К сожалению, большинство или невнимательно читали его условия или не смогли их правильно выполнить. На Ваш суд выносятся только 11 проектов 7-ми авторов.

Напоминаем условия конкурса:

1. Списки должны содержать программы, объединенные единым тематическим содержанием.

С этим условием как-то справились все, поскольку абсолютного критерия нет. А уж насколько успешно это получилось - судить Вам.

2. Количество программ в подборке должно быть таким, чтобы полностью заполнять 90- минутную кассету. С этим не справился только один конкурсант из п. Сельстрой Ленинградской обл.

3. Программы, вошедшие в список, должны иметь фирменное название, название фирмы-производителя и год выпуска.

На этом "срезалось" большинство. Очень многих подвел Bill Gilbert и другие подобные "хакеры" и пиратские фирмы, ворующие чужое программное обеспечение и оставляющие на нем свою "марку".

Многие написали названия фирм не для всех программ, включенных в сборники, поскольку взять их было негде, а может быть просто забыли.

4. В списки не должны включаться программы, ранее объявленные в наших сборниках "Детский Альбом-1", "Детский Альбом-2" и "HURRY UP".

Несколько человек, включивших в свои сборники программы типа Bruce Lee, Robin in the Wood, River Raid, Batty и т.п. не были допущены к конкурсу по этой причине. Правила есть правила и надо было внимательней читать условия.

Огромным подвохом для многих оказалась игра Green Beret. Эта программа оригинально разработанная японской фирмой KONAMI и конвертированная для "Спектрума" фирмой IMAGINE пользуется по-видимому очень большой популярностью среди "пиратов", поскольку каждый, кто включал ее в список, приводил самые невероятные названия фирмы-разработчика.

Призом в конкурсе является контракт на разработку инструктивного буклета к сформированной подборке программ. Стоимость 1-го контракта 1200 руб., второго -1000 руб., третьего - 800 руб.

Многие читатели указывают, что далеко не всегда можно в программе точно установить фирму производителя и год выпуска, и это ограничение очень сокращает возможный круг поиска. Это действительно так, но вот вопрос включать или не включать такую программу в список решали Вы сами.

Нашей задачей было найти людей, умеющих добывать информацию, а не просто имеющих и любящих игровые программы.

Но и мы вынесли для себя кое-что полезное. Теперь мы лучше представляем сколько же ходит по стране "изувеченных" программ с неграмотно переделанным названием и с указанием сомнительного "авторства".

Чтобы как-то помочь в преодолении такого положения, а заодно поощрить участников, мы поступим так:

Всем участникам конкурса, в том числе и тем, чьи предложения не были допущены к заключительному этапу, в начале сентября мы бесплатно вышлем готовящийся сейчас каталог-справочник по программному обеспечению, включающий около 5000 программ. Он окажет Вам необходимую помощь в Вашей работе с компьютером. Будет он также чрезвычайно полезным для проведения взаимных обменов программами и систематизации своей библиотеки.

Итак, представляем присланные на конкурс сборники, они приведены под теми девизами, которые им дали сами авторы. В тех случаях, когда такого девиза не было, мы будем писать NO NAME 1, NO NAME 2 и т. д.

В обсуждении итогов могут принять участие все желающие. Надо только написать нам и указать какой сборник Вы ставите на 1-е место, какой на второе, а какой на третье. За первое место будет начислено 5 очков, за второе - 3, за третье - 1.

Если найдете в списках "дезинформацию", то можете начислить этому автору 10 штрафных очков.

Мы допускаем, что все-таки где-то в представленных списках "дезинформации" просочилась, поскольку мы тоже многого не знаем, хотя что смогли, мы постарались отсеять.

В целом мы оцениваем предварительные результаты первого этапа как крайне неудовлетворительные.

Возможно, что условия оказались слишком жесткими для основной массы любителей. Посмотрим, как он будет развиваться дальше, когда Вы отберете три лучших сборника и мы предложим их авторам написать серьезный буклет описаний в обоснование своего проекта.

Не удовлетворившись такими итогами, мы начинаем новую кампанию - набор в заочные творческие группы "ЭКСПЕРТОВ" и "ОБОЗРЕВАТЕЛЕЙ" с очень высоким уровнем оплаты но и очень высокими требованиями. Желающие попробовать свои силы могут нам написать.

Подробности в следующем объявлении "Рекламного Приложения".

## ЗВЕЗДНЫЕ ВОЙНЫ

1	Elite	FIREBIRD	1884
2	Cybernoid 2	HEWSON	1988
3	Task Force	PREMIER	1989
4	Star Wars	DOMARK	1987
5	Phantis 1	DINAMIC	1987
6	Dark Star	DESIGN DESIGN	1986
7	Zynaps	HEWSON	1987
8	Xecutor	ACE	1987
9	Sabotage	ZEPPELIN	1987
10	Uridium	HEWSON	1987
11	Empire	DOMARK	1988
12	Virus	TELECOM	1988
13	Nemesis	KONAMI	1987
14	StarFight	HYBRID	1987
15	Cyberun	A. C. G.	1985
16	Death Star	SYSTEM 3	1985
17	EarthLight	TELECOM	1988
18	Unitrax	DOMARK	1987

## ПРИКЛЮЧЕНИЯ АСТРОНАВТОВ 1

1	Freddy Hardest 1	DINAMIC	1987
2	Freddy Hardest 2	DINAMIC	1987
3	Dan Dare	VIRGIN GAMES	1986
4	Quartet	ACTIVIZION	1986
5	Defender of Earth + 3 levels	ENIGMA VARIATIONS	1990
6	Return of Jedi	DOMARK	1987

7	Side Arms	CAPCOM	1987
8	Zub	MASTERTRONIC	1987
9	Forgotten Words + levels	U. S. GOLD	1987
10	North Star	GREMLIN	1988
11	Dark Fussion + 4 levels	GREMLIN	1988
12	The Ice Temple	BUBBLEBUS	1986
13	Dynatron	MASTERTRONIC	1987
14	Hypper Active	SPECIAL FX SOFT	1988

## ПРИКЛЮЧЕНИЯ АСТРОНАВТОВ 2

1	Alien 8	ULTIMATE	1985
2	Venom Strikes Back	GREMLIN	1988
3	Enigma Force	BEYOND	1985
4	Gun Runner	HEWSON	1987
5	People from Sirius	U. S. GOLD	1987
6	Omega One	MASTERTRONIC	1987
7	Shadowfire	BEYOND	1985
8	Blind Panic	MARTECH	1988
9	Catch 23	MARTECH	1987
10	Phantis 1	DINAMIC	1987
11	Phantis 2	DINAMIC	1987
12	Crosswize	FIREBIRD	1989
13	Prodigy	ELECTRONIC DREAMS	1986
14	Future Knight	GREMLIN	1986
15	Amaurote	MASTERTRONIC	1987
16	Nodes of Yesod	ODIN	1985
17	Antiriad	PALACE	1986
18	Equinox	MIKROGEN	1986

## ВОЗДУШНЫЕ СРАЖЕНИЯ

1	Ace	CASCADE	1986
2	Ace 2	CASCADE	1987
3	Cobra Force	PLAYERS PREMIER	1989
4	A.T.F.	DIGITAL INTEGRATIONS	1988
5	Sanxion	THALAMUS	1989
6	Thunder Blades + 5 levels	U.S. GOLD	1988
7	Flyer Fox	GAME GEMS	1984
8	Tomahawk	DIGITAL INTEGRATIONS	1985
9	Top Gun	OCEAN	1986
10	Skyranger	BE BE SOFT	1986
11	Operation Hormuz	DURELL	1985
12	Air Wolf	ELITE	1988
13	Typhoon + levels	OCEAN	1989
14	1943 of Midway	CAPCOM	1988

## СПОРТИВНЫЕ ИГРЫ

1	Pro Ski	CODEMASTERS	1987
2	World Games + 8 levels	ЕПУХ	1987

3	Uchi Kata 2	MARTECH	1988
4	Indoor Soccer	CODEMASTERS	1988
5	Supertest 1	OCEAN	1986
6	Show Jumping	ALLIGATO	1986
7	Basket Master	DINAMIC	1987
8	Passing Shot	IMAGE WORKS	1989
9	Xeno	A&F	1986
10	Basketball	IMAGINE	1985
11	Skate of Die	KINETIC DESIGN	1989
12	International Soccer	AUDIOGENIC	1988
13	Frank Bruno's Boxing	ELITE	1988
14	Play for Your Life	IMAGINE	1988

## КОСМИЧЕСКИЕ ВОЙНЫ

1	Ad Astra	GARGOYLE GAMES	1984
2	Code-Name Mat	MICROMEGA	1984
3	Code-Name Mat 2	DOMARK	1985
4	Death Star	SYSTEM 2	1985
5	Dark Star	DESIGN	1989
6	Elite	FIREBIRD	1986
7	Exolon	HEWSON	1987
8	Lunar Jetman	ULTIMATE	1983
9	Jetpac	ULTIMATE	1983
10	Luna Crabs	MICROMEGA	1983
11	Mercenary	NOVAGEN	1987
12	Planetoids	PSION	1982
13	Quazatron	PSION	1985
14	Space Wars	HEWSON	1983
15	Trantor	PROBE	1987
16	Venom	GREMLIN	1988
17	Zub	M. A. D. GAMES	1986
18	Zaxxon	U. S. GOLD	1986
19	Academy	CRL	1986
20	D.N.A. Warrior	ARTRONIC	1989
21	R-Type	IREM CORP.	1987
22	Thunderceptor	OCEAN	1989

## БЕГИ И СТРЕЛЯЙ ВО ВСЕ, ЧТО ДВИЖЕТСЯ

1	Attic Attack	ULTIMATE	1983
2	Android-1	VORTEX	1983
3	Android-2	VORTEX	1983
4	Avenger	GREMLIN	1986
5	Black Magic	PARAGON	1987
6	Batman	OCEAN	1986
7	Elevation Act	TAITO	1987
8	Kane	BYTE	1986
9	Eagle Nest	PANDORA	1986
10	Fred	INDESCOMP	1985
11	Gunfright	ULTIMATE	1985
12	Ghost'n Goblin	JAPAN CAPSULE	1986

13	Game Over	DYNAMIC	1985
14	Ice Palace	ELITE	1988
15	Ikari Warriors	ELITE	1986
16	Laser Tag		1987
17	Saboteur-1	DURELL	1986
18	Saboteur-2	DURELL	1987
19	Tutankhamoun	MICROMANIA	1983
20	Dan Dare	VIRGIN	1986
21	Cosmo Police	OCEAN	1986
22	Martianoids	ULTIMATE	1987

### ЛАБИРИНТ-1

1	Alien 8	ULTIMATE	1985
2	Knight Lore	ULTIMATE	1984
3	Pentagram	ULTIMATE	1986
4	Batman	OCEAN	1987
5	Nosferatu	MACMILLAN	1986
6	Chimera	FIREBIRD	1985
7	Demon's Revenge	FIREBIRD	1989
8	Heartbroken	DYNAMIC	1989
9	Sheer	K.J. BEZANT	1983
10	Panama Joe	SINCLAIR/HUDSON	1986
11	Gauntlet + 21 block	GREMLIN	1985
12	Three Weeks in Paradize	MIKROGEN	1986
13	Pyjamarama	MIKROGEN	1986
14	Nipper 2	GREMLIN	1987
15	Solomon's	PROBE	1987
16	Starquake	S. J. CROW	1985
17	Black Magic	PARAGON	1987

### ПИЛОТ БУДУЩЕГО

1	Academy	PETE COOKE	1986
2	Death Star	ATARI CORP.	1985
3	Elite	FIREBIRD	1987
4	Freddy 1	IMAGINE	1988
5	Freddy 2	IMAGINE	1988
6	Chronos	GREMLIN	1987
7	Equinox	MICRO-GEN	1986
8	Dan Dare	VIRGIN	1987
9	L. Force	FTL	1986
10	Trantor	PROBE	1988
11	Amaurote	MASTERTRONIC	1987
12	Zub	MASTERTRONIC	1987
13	Exolon	HEWSON	1987
14	Marsport	GARGOYLE	1985
15	Star Raiders 2	ATARI	1987
16	Earthlight	TARZAN BOY	1989

## NO NAME-1

1	Mr. Wimpy	OCEAN	1984
2	Space Raiders	PSION	1982
3	Falcon Patrol 2	VIRGIN	1985
4	Cosmic Pirate	B. C. G.	1983
5	Zipper Flipper	R. E. D.	1984
6	Equinox	MICRO-GEN	1986
7	The Fall Guy	ELITE	1986
8	Roller Coaster	ELITE	1985
9	Rockman	MASTERTRONIC	1985
10	Painter	A+F SOFTWARE	1984
11	Buck Rogers	O. S. A.	1985
12	PSST	A. C. G.	1983
13	Penetrator	C. B. S.	1982
14	Zoom	IMAGINE	1983
15	Trail Blazer	Mr. CHIP S.	1986
16	Twister	SYSTEM 3	1986
17	Hunchback	OCEAN	1983
18	The Devils Crow	MASTERTRONIC	1985
19	Magico la ta Peto	TURBO	1985
20	Ball Breaker	CRL GROUP	1987
21	Crystal Castle	ANDROMEDA	1986
22	PI-Balled	AUTOMATA	1984
23	Armageddon	OCEAN	1983

## ЛАБИРИНТ-2

1	Eagles Nest	PANDORA	1988
2	Spirits	TOPO SOFT	1987
3	Great Escape	OCEAN	1986
4	Robocop	OCEAN	1987
5	Game Over 1	DINAMIC	1985
6	Rat Race	OCEAN	1985
7	Scooby Doo	ELITE	1986
8	Atic Attack	ULTIMATE	1983
9	Sabre Wolf	ULTIMATE	1984
10	Knight Lore	ULTIMATE	1984
11	Alien 8	ULTIMATE	1985
12	On the Run	ULTIMATE	1986
13	Chimera	FIREBIRD	1983
14	Droid	FIREBIRD	1986
15	Rygar	U.S. GOLD	1988
16	Hysteria	SOFTWARE PROJ.	1987
17	Dan Dare	VIRGIN	1986
18	Olli & Lisa	ISI	1986
19	Dynamite Dan	MIRRORSOFT	1985

ДЛЯ РАБОТЫ ПО ТЕМАТИКЕ ИГРОВОГО  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ "ИНФОРКОМ" ПРИГЛАШАЕТ К СОТРУДНИЧЕСТВУ  
"ХАККЕРОВ" ДВУХ КАТЕГОРИИ

ЭКСПЕРТОВ и ОБОЗРЕВАТЕЛЕЙ

1. Задачи ЭКСПЕРТА.

Экспертом может быть, любой человек, способный с глубоким знанием дела разобраться с интересной программой, литературно изложить ее цели и задачи, порядок ее настройки, порядок работы с ней, дать полезные советы и подсказки. Одним словом - задача ЭКСПЕРТА - дать глубокий всесторонний анализ игры. Понятно, что не всякая программа заслуживает такого подхода. Подобрать ее - тоже задача эксперта. ЭКСПЕРТ отвечает за качество самой программы. Исследование неинтересной "стрелялки" может оказаться малонужным. Особым приоритетом будут пользоваться программы, требующие специальных знаний - адвентюрные, аркадно-адвентюрные, стратегические, менеджмент, имитаторы сложных технических устройств, аналоги традиционных настольных игр и т.п.

2. Задачи ОБОЗРЕВАТЕЛЯ.

Обозревателем может быть только человек, имеющий доступ к самым новым появляющимся в стране программам. В отличие от ЭКСПЕРТА, от него не требуется глубокого уровня проработки программ, а только рецензия (аннотация), написанная интересно и на достаточном художественном уровне, но программа, в отличие от ЭКСПЕРТА, обязательно должна быть новой.

В отличие от ЭКСПЕРТА, ОБОЗРЕВАТЕЛЬ не несет ответственности за качество программы, которую он освещает, то есть его обзор может иметь и негативное содержание, поскольку отрицательная рецензия - тоже рецензия.

3. Порядок выполнения работ.

Вы пишете нам письмо, в котором указываете чем Вы хотите заняться - обзором или экспертной проработкой. Указываете также какие программы Вы берете на себя. Обобщив полученные сведения, поступившие от разных людей, чтобы избежать, дублирования, мы дадим Вам "добро", указав какие из предложенных Вами программ Вам поручаются.

Получив Вашу законченную работу, мы либо принимаем ее в свой портфель и высылаем Вам купон на оплату, либо не принимаем и тогда возвращаем ее Вам с замечаниями которые позволят Вам быстрее набрать необходимый опыт и включиться в производительную работу.

4. Система оплаты.

Оплата и для ЭКСПЕРТОВ и для ОБОЗРЕВАТЕЛЕЙ одинакова - 15 рублей за одну машинописную страницу материала, напечатанного через 1,5 интервала (40 строк) с возможностью ее повышения до 20 рублей по мере приобретения опыта.

Если Вы не имеете возможность присылать материал в машинописном виде, то мы примем и разборчивую рукопись, но уровень оплаты на 30% ниже.

Если Вы имеете доступ к IBM-совместимой технике, то можете присылать материал на дискетах. В этом случае уровень оплаты - на 30% выше. Кодировка символов русского алфавита - альтернативная. Дискеты мы конечно вернем.

5. Система связи.

Связь только, по почте. Для ускорения – ставьте пометку на конверте "О" или "Э". Не посвящайте письмо разнородным вопросам. Принцип "одно письмо - одна проблема" - и оно попадет в нужные руки без блужданий по многочисленным столам.

6. Примеры.

В "Игровом приложении" данного выпуска Вы найдете цикл, посвященный

стратегическим играм. В этом цикле мы рассматриваем игры TOBRUK, LEGIONS OF DELATH, VULCAN, SAMURAI и др. как обзоры. Это примерно то, что мы хотели бы получать от ОБОЗРЕВАТЕЛЯ. Более широко рассмотренная программа, INVASION - это примерно то, что мы хотели бы получать от ЭКСПЕРТА.

#### 7. Нежелательные контакты.

Если Вы располагаете описаниями, распространяемыми в среде любителей, и рассчитываете использовать их в своей работе, то будет лучше, если Вы воздержитесь от контактов с нами. Нас интересуют только Ваши авторские разработки и у нас очень мало времени на переписку.

#### 8. Наши интересы.

"ИНФОРКОМ" никогда не занимался достаточно серьезной работой в области игрового программного обеспечения. В то же время, у нас есть многочисленные обращения читателей с просьбой о проведении обзоров новых игровых программ и глубокой проработки известных. Обращаясь к Вам с таким предложением, мы рассчитываем сделать ZX-РЕВЮ содержательнее и интереснее, а также подготовить определенный контингент соавторов для длительной работы. Может быть те из Вас, кто сумеют захватить целые направления, в дальнейшем получат постоянную рубрику.

Мы также рассчитываем, что сделанные по нашему заказу и оплаченные нами разработки не появятся нигде, кроме как у нас, причем это может быть не обязательно будет ZX-РЕВЮ. При достаточном материале это может быть особый сборник.

#### 9. Персональные приглашения.

Объявление касается всех желающих, но мы хотели бы особо отметить, сотрудничество некоторых товарищей и пригласить их персонально:

Богачева В. В. , Брунова О. В. , Дейну А. А. , Жукова К. А. , Климова Д. В. , Котина Д. , Кочнева С. В. , Линькова И. П. , Сипухина Р. И. , Сорокина А. В. , Татевосяна С. , Фокина А. М. , Хахонова А. Н. , Шилина Д. П.

И особо пригласить в качестве ОБОЗРЕВАТЕЛЯ Деменина В. А.

## ПЛЮС

### НАУЧНО-ТЕХНИЧЕСКИЙ КООПЕРАТИВ "ПЛЮС" ПРЕДЛАГАЕТ ВАШЕМУ ВНИМАНИЮ СВОИ УЧЕБНО-МЕТОДИЧЕСКИЕ РАЗРАБОТКИ

#### Методическая разработка (МР) N1.

Любители компьютерных игр найдут в ней несколько вариантов схем подключения джойстика. Наличие нескольких вариантов позволяет Вам выбрать наиболее приемлемый по наличию элементной базы.

Те, у кого есть принтер, смогут подключить его с помощью одной из схем последовательного или параллельного интерфейса принтера. Приведенные программы поддержки принтера (драйверы) позволят Вам распечатывать как текстовую, так и графическую информацию. Схемы интерфейсов и программы поддержки предназначены для работы с любыми принтерами, оборудованными одним из стандартных интерфейсов: CENTRONICS (ИРПП-М), ИРПП (IFSP), RS-232C (V.24).

Для занимающихся компьютерной графикой и живописью предназначена схема светового пера и распечатка работающего с ним графического редактора LP48K.

Схема программатора для ПЗУ с УФ стиранием позволит любителям технического творчества самостоятельно программировать такие микросхемы как 2716, 2764, 27128, 27256, 573 РФ2, 5, 4, 6, 8.

Объем МР1 - 34 страницы, стоимость - 30 рублей.

Со времени начала ее распространения (с сентября 1990г.) в адрес НТК не прекращается поток писем с положительными отзывами. Все схемы, входящие в разработку уже повторены многими пользователями, которые отмечают простоту изложения материала, доступность элементной базы и высокую повторяемость схем.

#### Методическая разработка (МР) N2.

Вторая методическая разработка МР2 состоит из двух частей. Первая часть посвящена контроллеру накопителя на гибких магнитных дискетах НГМД. Контроллер полностью совместим с фирменной системой BETA-DISC INTERFACE. Он выполнен на базе микросхемы 1818ВГ93 (WD1793) и поддерживает работу до 4-х НГМД - 3-х или 5-ти дюймовых, односторонних и двусторонних, 40 и 80 дорожечных. Разработка содержит подробное описание схемы, сигналов, принципов работы интерфейса, рекомендации по сборке и наладке, систему команд дисковой операционной системы TRDOS.

О преимуществах работы с НГМД по сравнению с работой с магнитофонной лентой знают все, это новый уровень работы с компьютером, и мы надеемся, что наша разработка поможет Вам перейти на этот уровень.

Вторая часть МР2 посвящена стыковке компьютера с бытовыми телевизорами. Наиболее качественное цветное изображение получается при стыковке компьютера по R-G-B входу телевизора. При стыковке по антенному входу с кодировкой по системе SEKAM или PAL происходит некоторая потеря качества цветного изображения, но зато телевизор не требует переоборудования. В некоторых случаях может удовлетворить стыковка через видеовход с чернобелыми или цветными телевизорами без передачи информации о цвете, т.е. в черно-белом изображении.

Вторая часть МР2 содержит схемы всех перечисленных способов стыковки компьютера с телевизорами, причем схемы предназначены не только АЛЯ полупроводниковых, но и для ламповых телевизоров.

Стоимость МР2 - 30 руб.

#### Методическая разработка (МР) N3.

Третья методическая разработка (МР3) подготовлена по многочисленным письмам и пожеланиям пользователей "Спектрума". В нее входит принципиально новый способ русификации "Спектрума", выгодно отличающийся от известных способов.

Знакогенератор кириллицы размещается в неиспользуемой области ROM, там же

размещены некоторые подпрограммы, которые добавляют в операционную систему "Спектрума" два новых регистра клавиатуры R и N к уже имеющимся K, L, C, E, G. Таким образом, в любой строке могут быть представлены буквы как русского, так и латинского алфавита одновременно, т.е. например, операторы Бейсика на английском языке, а текст на русском.

Также приведены дополнения к программам поддержки интерфейсов принтеров, опубликованных в нашей МР1 для вывода на печать смешанного русско-латинского текста.

Кроме этого, в МР3 по просьбам пользователей включены схемы интерфейсов джойстиков, рассчитанных на двух игроков - INTERFACE 1 и SINCLAIR, а так же схема программируемого джойстика, позволяющая запрограммировать все положения рукоятки и кнопки в соответствии с клавишами клавиатуры и более простой вариант - механически программируемый (путем перестановки перемычек) джойстик.

Стоимость МР3 - 30 руб.

Все программы, входящие в разработки, приведены в виде листинга и блоков шестнадцатеричных кодов.

Свои заявки на приобретение МР1-МР3 Вы можете направлять по адресу:

127566, Москва, И-566, НТК "Плюс". Оплата производится, только наложенным платежом.

Просьба при всех обращениях к нам прикладывать к письму конверт с обратным адресом. В заявке необходимо указать свой полный адрес, фамилию, имя, отчество.

СКАЖИТЕ, ВЫ ЧУВСТВУЕТЕ, КАК С ПРИОБРЕТЕНИЕМ КОМПЬЮТЕРА ВАША ЖИЗНЬ  
СТАЛА СПОКОЙНЕЕ И ВЫ ПОЧУВСТВОВАЛИ СЕБЯ УВЕРЕННЕЕ?

НЕТ?

ТОГДА ВАМ НУЖЕН МАЛЕНЬКИЙ АУТОТРЕНИНГ.

Внимательно прочитайте следующие строки.

\* \* \*

Я чувствую себя прекрасно. Мой кругозор сильно расширился. У меня появились интересные друзья и знакомые. Мне не надо думать о завтрашнем дне, работа с компьютером решит все мои проблемы.

На работе ожидается сокращение? Это прекрасно! Тем, кто останется, повысят зарплату. Меня оно не коснется, все знают, что я один из тех, кто приобщился к таинственному компьютерному знанию.

Пусть беспокоится тот, кто забыл, когда последний раз освоил что-нибудь сам, без указки начальства.

Жизнь становится дороже день ото дня? Наверное так и должно быть, но я спокоен. Мои проекты, мои идеи тоже дорожают день ото дня.

Нет уверенности в завтрашнем дне? А зачем она мне? И завтрашний и послезавтрашний дни у меня давно спланированы. Я думаю о двадцать первом веке, а завтрашний день пусть сам подстраивается под мои планы.

Нет финансовой стабильности? Пусть об этом беспокоятся те, кто вкладывает средства, в сбербанк или в акции каких-нибудь предприятий. Я вкладываю их в знания, а их не обменяют как ставшие ненужными купюры, да и процент они дадут побольше.

Жизнь становится суетливее? Возможно, но когда я часами трясусь в переполненном автобусе, моя голова работает над новыми идеями и дает отдачу не хуже, чем дома, в мягком кресле.

Нечем себя занять, по телевизору нечего смотреть, с семьей некуда пойти? Мне бы Ваши проблемы. Мы с семьей можем по сорок восемь часов в день что-то осваивать, над чем-то работать. Дети, правда увлекаются игрушками, но через пол-года это пройдет, они поймут, что самая интересная игра - это игра в сыщиков. Они начнут пытливо разбираться с работой программ, и все секреты будут им открыты. Прекрасное будущее им гарантировано и без армии грабителей-репититоров.

\* \* \*

НУ КАК, ПОМОГ ВАМ АУТОТРЕНИНГ?

Неужели нет? Тогда с Вами все ясно!

Вы просто не приобрели наш трехтомник для изучения программирования в машинных кодах и на языке АССЕМБЛЕРА. Очень напрасно. Тысячи читателей уже это сделали, а кому же хочется быть в последних рядах на спринте в 21-ый век?

А ведь мы рассчитываем делать новые книги, в которых будем опираться на эти как на базовые.

Заплатив шестьдесят (пока) рублей Вы приобретаете не только доступный учебник, Вы приобретаете уверенность в завтрашнем дне. Это дорого? Да, недешево. Знания стоят денег. Только незнание ничего не стоит, но платить за него придется дороже.

Будут меняться поколения компьютеров. Сокращая по обходной дуге разрыв в программировании между нами и американцами, мы будем все быстрее и быстрее осваивать новые технологии. Сегодня СИНКЛЕР, завтра IBM, послезавтра что угодно. Но первый шаг ведь делать надо! И поверьте, это шаг именно по той тропинке, которая сократит в десятки раз сложный путь по освоению новых компьютерных систем будущего.

Пишите, адрес Вам известен.

С наилучшими пожеланиями,  
ИНФОРКОМ.

## Содержание

<b>РАЗДЕЛ ДЛЯ НАЧИНАЮЩИХ.....</b>	<b>1</b>
<b>МЕГА-БЕЙСИК.....</b>	<b>4</b>
УПРАВЛЕНИЕ ВЫПОЛНЕНИЕМ ПРОГРАММЫ .....	4
<i>Процедуры.</i> .....	4
REPEAT-UNTIL.....	4
<i>Процедурный стек</i> .....	4
ОТЛАДКА И РЕДАКТИРОВАНИЕ ПРОГРАММ.....	5
<i>Операторы BRANCH_, TRON_, TROFF_ и SPEED.</i> .....	5
<i>Команды AUTO и DELETE.</i> .....	5
<i>Команды BRON_ и BROFF_.</i> .....	5
ЗВУК.....	5
PLAY_n,l,s,d,f.....	6
ISG.....	6
МАШИННЫЕ КОДЫ И МЕГА-БЕЙСИК .....	6
<i>Команды DOKE и CALL.</i> .....	6
FRONT-PANEL.....	7
НОВЫЕ КОМАНДЫ.....	7
<b>ПРОФЕССИОНАЛЬНЫЙ ПОДХОД .....</b>	<b>9</b>
ОРУДИЯ ПРОИЗВОДСТВА.....	9
1. Редактор/Ассемблер.....	9
2. Монитор.....	10
3. Внешняя память.....	10
4. Принтер.....	10
5. Книги по Z-80.....	11
<b>СЕКРЕТЫ ПЗУ .....</b>	<b>13</b>
ПРОГРАММЫ, ОБСЛУЖИВАЮЩИЕ РАБОТУ С МАГНИТОФОНОМ.....	13
<b>FORUM .....</b>	<b>30</b>
<b>МАЛЕНЬКИЕ ХИТРОСТИ .....</b>	<b>36</b>
<b>ADVENTURE LESSONS .....</b>	<b>37</b>
9. Книги, записки, объявления и пр. ....	37
10. Музыкальные инструменты.....	37
11. Растения.....	37
12. Веревка, канат (ROPE).....	37
13. Лестница - LADDER.....	37
5. ВЗАИМОДЕЙСТВИЕ С ДРУГИМИ ПЕРСОНАЖАМИ.....	38
6. ИСПОЛЬЗОВАНИЕ ОТГРУЗКИ ПРОГРАММЫ НА ЛЕНТУ.....	40
7. ПОЛЕЗНЫЕ ПРАКТИЧЕСКИЕ ПРИЕМЫ.....	40
ЗАКЛЮЧЕНИЕ.....	41
<b>GRAPHIC ADVENTURE CREATOR .....</b>	<b>42</b>
ЛОГИКА.....	43
ГРАФИКА.....	44
<b>SCRABBLE .....</b>	<b>45</b>
ВСТУПЛЕНИЕ.....	45
I. ПРАВИЛА ИГРЫ.....	45
<i>Игра.</i> .....	45
<i>Счет.</i> .....	46
II. ИГРА С КОМПЬЮТЕРОМ.....	47
<i>Главное меню.</i> .....	47
<i>Как делать ход.</i> .....	48
<i>Ход пустой фишкой.</i> .....	48
<i>Запрещенные ходы.</i> .....	48
<i>Оспаривание хода.</i> .....	48
<i>Окончание игры.</i> .....	48
<i>Откладывание игры.</i> .....	48
<i>Загрузка отложенной программы.</i> .....	48

Стратегия.....	48
Словарь.....	49
<b>FORUM.....</b>	<b>50</b>
<b>STRATEGIC GAMES.....</b>	<b>55</b>
TOBRUK.....	55
LEGIONS OF DEATH.....	56
BATTLEFIELD GERMANY.....	58
SAMURAI.....	59
BATTLE OF BRITAIN.....	60
VULCAN.....	61
DESERT RATS.....	62
INVASION.....	63
<i>Ваши боевые части.....</i>	<i>63</i>
1. Радары.....	63
2. Минные части.....	63
3. Воздушная пехота.....	63
4. Парящая броня.....	63
5. Страйдеры.....	63
<i>Боевые части инопланетян.....</i>	<i>64</i>
1. Тяжеловооруженные части.....	64
2. Части быстрого реагирования.....	64
3. Дроиды.....	64
<i>Игра.....</i>	<i>64</i>
<i>Дополнительные возможности.....</i>	<i>64</i>
<i>Полезные советы.....</i>	<i>65</i>
<i>Счет.....</i>	<i>65</i>
<i>Управление игрой.....</i>	<i>65</i>
<i>Расписание клавиатуры.....</i>	<i>65</i>
<b>ТЕХНИЧЕСКОЕ ПРИЛОЖЕНИЕ.....</b>	<b>66</b>
БУФЕРИЗАЦИЯ.....	66
<b>НАШ КОНКУРС.....</b>	<b>77</b>
ЗВЕЗДНЫЕ ВОЙНЫ.....	78
ПРИКЛЮЧЕНИЯ АСТРОНАВТОВ 1.....	78
ПРИКЛЮЧЕНИЯ АСТРОНАВТОВ 2.....	79
ВОЗДУШНЫЕ СРАЖЕНИЯ.....	79
СПОРТИВНЫЕ ИГРЫ.....	79
КОСМИЧЕСКИЕ ВОЙНЫ.....	80
БЕГИ И СТРЕЛЯЙ ВО ВСЕ, ЧТО ДВИЖЕТСЯ.....	80
ЛАБИРИНТ-1.....	81
ПИЛОТ БУДУЩЕГО.....	81
NO NAME-1.....	82
ЛАБИРИНТ-2.....	82