

Руководство пользователя. RAMFOS

1. Введение

Одним из главных критериев эффективности программного обеспечения ПЭВМ является гибкая архитектура, широкий набор функциональных и сервисных возможностей, избавляющих пользователя от "мелкой" рутины. Эти принципы были положены в основу создания данного пакета программного обеспечения (ПО).

Структура ПО предусматривает три уровня работы:

Режим работы с файлами, обеспечиваемый операционной системой RAMFOS, являющейся функциональным аналогом известной оболочки для MS_DOS - "Norton Commander". Систему RAMFOS эффективно дополняет операционная система MX_DOS, организующая работу с файлами совместно с дисководом.

Режим работы с памятью и текстами, обеспечиваемый Отладчиком и Редактором. Прототипом Отладчика является программа, опубликованная в журнале "Радио", которая подверглась существенным доработкам, как в функциональном, так и в сервисном направлениях. Редактор новый, по своему уровню и возможностям соответствует профессиональным текстовым процессорам.

Режим обработки памяти и текстов, обеспечиваемый Ассемблером и Дизассемблером. Прототипам этих программ также послужили программы, опубликованные в журнале "Радио", но их возможности существенно возросли, как в функциональном, так и в сервисном планах. Ассемблер и Дизассемблер позволяют транслировать программы объемом свыше 16 кБайт.

2. Операционная система RAMFOS

Загружается вместе с BIOS в системное ОЗУ при включении ПЭВМ или по <сбросу>.

Характеризуется двумя режимами: 1 - получаемый при инициализации RAMFOS и 2 - при запросе (F824h) файла из программы пользователя. Оба режима практически равноценны по предоставляемым функциональным возможностям. Во втором режиме некоторые функции блокируются из-за их ненужности или нежелательности. При инициализации RAMFOS инициализируется BIOS, который проверяет наличие даты. Если ее нет (при первой инициализации), то BIOS запрашивает дату - необходимо ввести 3 параметра через запятую. После этого в верхней служебной строке выводится имя ПЭВМ и текущие режимы. Затем RAMFOS распечатывает файловую таблицу. В заголовке файловой таблицы отображается следующая информация: обозначение "диска" - RAM(DISK) или ROM(DISK); номер версии операционной системы RAMFOS (ПЗУЗ); после первой наклонной черты располагаются три символа, которые соответствуют типу файла, установленному из программы пользователя; после второй наклонной черты цифра (1...8) означает номер текущей страницы RAM - диска. На нижней служебной строке выводится список действующих функций:

F1-COPY F2-DELETE F3-MLOAD F4-MSAVE F5-EDIT F6-DOS F7-DEBU6

Адрес	Назначение
0000H 8FFFH	ОЗУ пользователя (36 кБайт)
9000H BFFFH	Экранная область (12 кБайт)
C000H C5FFH	Знакогенератор КОИ - 8
C600H D1EFH	BIOS_MX (3 кБайт); C800 - "окно" BIOS
D1F0H D29FH	Служебные ячейки
D2A0H D2FFH	Системный стек
D300H E7FFH	Системное ОЗУ
E800H F7FFH	Область загрузки Отладчика и Редактора
F800H FFBFH	ОС RAMFOS; F800 - "окно" RAMFOS
FFC0H FFDFH	"Окно" микропроцессора
FFE0H FFE3H	Порты клавиатуры (КР580ВВ55А)
FFE4H FFE7H	Порты "программатора" (КР580ВВ55А)
FFE8H FFEВH	Порты контроллера дисководов (КР1818ВГ98)
FFECH FFEFH	Порты таймера (К580ВВ53)
FFF0H FFF3H	Дополнительные порты контроллера дисководов
FFF4H FFF7H	Резерв
FFF8H	Порт контроллера цвета
FFFCH FFFВH	Резерв
FFFCH	Закрытие страниц RAM / ROM - дисков
FFFDH	Открытие страницы RAM - диска; D2...D0 - номер страницы
FFFЕH	Открытие страницы ROM - диска

Табл. 1.
План ОЗУ и адреса внешних устройств.

<COPY> - копирование, во втором режиме, выбранного курсором файла в ОЗУ по указанным в таблице адресам или по адресу, заданному с клавиатуры на нижней кромке файловой таблицы. В первом режиме загрузка в ОЗУ не происходит, копируется лишь только оглавление файла из себя в себя. В этом режиме можно вносить изменения в таблице адресов загрузки.

<DELETE> - удаление файла, выбранного курсором. Если курсор на пустой строке, то запрос на удаление даты.

<MLOAD> - чтение (сравнение) файла с ленты (ой) на RAM - диск.

<MSAVE> - запись файла с RAM - диска на ленту. После выполнения директивы сбрасывается признак несохранённости файла (инверсный цвет даты в таблице). Если курсор на пустой строке, то на ленту записывается весь список файлов одним пакетом.

<EDIT> - на пустой строке запускается редактор текстов Edit_MX, на строке с именем файла запускается директива редактирования имени файла.

<DOS> - при наличии дисководов и при вставленной дискете запускается операционная система .MX DOS, размещенная на дискете. В противном случае инициализируется ROMDISK. Во втором режиме инициализируется только ROMDISK.

<DEBUG> - в первом режиме запускается отладчик Debug_MX. Функция <MEREE> - присоединение текстовых, графических файлов, реализуется директивой <COPY> при указании адреса загрузки, равного конечному адресу предыдущего загруженного файла. <↑>, <↓> - перемещение по списку файлов. В таблице на экране помещается 18 файлов. Если их больше, то переход на следующую часть списка осуществляется курсором <↓> и обратно <↑>. <BK> - в первом режиме осуществляет загрузку выбранного файла с типом файла EXE и COM с передачей ему программного управления. Во втором режиме реализуется только функция копирования файла, тип которого задан программой пользователя. После выполнения команды управление возвращается программе пользователя, сделавшей запрос файла.

Формат файла на RAM - диске и ROM - диске:

- 3 байта - дескриптор (опознаватель) файла [коды D3H]
- 8 байт - имя файла [FILENAME]
- 1 байт - пробел []
- 3 байта - тип файла [EXE, ...]
- 1 байт - признак (не) сохраненности файла [код (8BH) 8CH]
- 3 байта - дата создания файла
- 5 байт - резерв (используется в MX_DOS)
- 2 байта - начальный адрес загрузки файла [мл.байт, ст.байт]
- 2 байта - конечный адрес загрузки файла [мл.байт, ст.байт]
- 2 байта - контрольная сумма загружаемой части [мл.байт, ст.байт]
- N байт - загружаемая часть файла
- 2 байта - указатель на начало файла (оглавление) [мл.байт, ст.байт]

Формат файла на «ленте»:

- заголовок 3 - 4 секунды
- 2 байта - длина оглавления файла [мл.байт, ст.байт]
- 15 байт - имя файла
- заголовок 3 - 4 секунды
- 2 байта - длина полного файла без указателя [мл.байт, ст.байт]
- N байт - полный файл с RAM - диска без указателя.

3. Базовая система ввод - вывода (BIOS)

Представляет собой широкий набор стандартных подпрограмм ввода-вывода и связи с периферией ПЭВМ, с широким набором сервисных функций. Функции BIOSa реализуются в двух формах: через стандартные точки "окон" BIOSa и RAMFOS, и через систему специальных служебных кодов (87H...BFH), реализуемых через подпрограмму вывода символа (строки) на экран.

RAMFOS	BIOS	Сохраняемые регистры	Функции
F800H	-	-	Инициализация RAMFOS
-	C800H	-	Инициализация BIOS
F803H	C803H	HL, DE, BC	Ввод символа с клавиатуры; A - код
F806H	C806H	HL, DE, BC	Ввод байта с ленты; A - код; если ошибка, то код 3FH и флаг переноса
F809H	C809H	HL, DE, BC, AF	C - код символа на экран (500 симв/с)
F80CH	C80CH	HL, DE, BC, AF	C - код байта на ленту, в A - помечается байт укорочения, старт - бита, число которого, умноженное на 15 соответствует числу тактов выполненных МП после предыдущего обращения к данной подпрограммы
F80FH	C80FH	HL, DE, BC, AF	A - код символа на экран
F812H	C812H	HL, DE, BC	Опрос нажатия клавиатуры; A=00H (FFH) - нет (да), 0FH=нуль (нет)
F815H	C815H	HL, DE, BC, AF	A - код байта в HEX виде на экран
F818H	C818H	DE, BC	HL - вывод строки символов на экран, до нулевого байта
F81BH	C81BH	HL, DE, BC	Ввод кода нажатой клавиши; A= код, иначе - FFH
F81EH	C81EH	DE, BC, AF	Запрос курсора; H - номер строки (00H...18H), L - номер позиции в строке (00H...3FH), начало отсчета от левого верхнего угла экрана
F821H	C821H	HL, DE, BC	Чтение байта с экрана; A - код в КОИ - 8
F824H	-	-	Запрос файла; флаг - нуль, если прервано, иначе файл загружен, тогда в HL - начальный адрес загрузки, DE - конечный адрес загрузки
-	C824H	HL, DE, BC	Чтение заголовка с ленты
F827H	-	-	Сохранение файла на RAMDISKe в диалоговом режиме; HL - начальный адрес, DE - конечный адрес, в итоге; флаг - нуль, если прервано (отменено), иначе выполнено
-	C827H	HL, DE, BC, AF	Запись заголовка на ленту
F82AH	C82AH	DE	Подсчет контрольной суммы в BC; HL - начальный адрес, DE - конечный адрес
F82DH	-	HL, DE, BC, AF	Установка типа файла; HL - начальный адрес, подпрограмма использует первые три символа
-	C82DH	DE, BC	Вывод нумерованной служебной нижней строки; HL - нач.адрес строки, Первый код строки - номер строки. Номер запоминается BIOS - ом и позволяет ему не допускать повторного вывода ранее выведенной строки. Номера 01H...2FH отведены для системных программ, Служебные строки можно выводить и общей подпрограммой C818H, но в этом случае строке присваивается нулевой номер (после команды - очистить нижний служебную строку)
F830H	C830H	DE, BC, AF	Запрос верхней границы памяти в HL
F833H	C833H	HL, DE, BC, AF	HL - установка верхней границы памяти
F836H	-	HL, DE, B, A	Чтение байта с RAM/ROM диска; в HL - задается адрес диска, в A - задается номер страницы диска тремя младшими битами (D0...D2), бит D7=1(0) - задает ROM/RAM диск, а в C - полученный код
-	C836H	BC	Ввод командной строки; в HL - задается начальный адрес буфера (00H...FFH), E - режим; при E=00H - перемещение курсора влево с очисткой знакоместа, а вправо - блокируется; при E=XXH - режим без очистки знакоместа, а при перемещении вправо на экран выводится содержимое буфера до не символического кода. Работа ПП прерывается клавишами с кодом 00H...1FH (в буфер не записываются). При этом код 1FH <CTP> сопровождается флагом нуля (и не переноса), код 0DH <BK> - флагом переноса (и не нуля), а остальные коды - флагом не перенос. По выходе из ПП в DE возвращается начальный адрес буфера, в HL - адрес текущей незаполненной ячейки буфера, в A - последний введенный код. Если код 0DH <BK> и если были введены символы, то присутствует еще и флаг минус, иначе плюс
F839H	-	HL, DE, BC, A	Запись байта на RAMDISK: задается в HL - адрес диска, в A - номер страницы диска (биты D8...D2, а D7=0), в C - код
-	C839H	BC	Преобразование командной символической строки в цифру (HEX слово): в DE - задается начальный адрес буфера, в HL - выдается результат преобразования, флаг переноса - означает завершение обработки командной строки (до кода 0DH или ошибки), флаг нуля означает обнаружение ошибки в строке символов и на экран выводится символ (?), флаги не нуль и не перенос означают завершение обработки фрагмента строки (до запятой или пробела)
F83CH	C83CH	HL, DE, BC, AF	Установка курсора: H - номер строки (00H...18H), L - номер позиции курсора (00H...3FH)
F83FH	-	-	Сохранение файла в бездиалоговом режиме: HL - начальный адрес, DE - конечный адрес, в итоге: флаг перенос - диск мал, иначе - выполнено.
-	C83FH	HL, DE, BC, AF	C - вывод кода символа на принтер
F842H	-	-	Коррекция начального адреса загрузки текущего файла: HL - адрес
-	C842H	HL, DE, BC, AF	A - вывод байта на принтер
F845H	-	HL, DE, BC, AF	Получение имени последнего загруженного файла: в HL - задается адрес буфера (12 байт) пользователя
-	C845H	HL, DE, BC	Ввод имени файла: флаг - нуль, если работа подпрограммы прервана (например, отмена сохранения файла)
F848H	-	HL, DE, BC, AF	Установка имени сохраняемого файла: в HL - задается адрес буфера пользователя
-	C848H	HL, DE, BC, AF	Формирование звука таймером; DE - период частоты (N*0.5 мкс), BC - длительность сигнала (N*10 нс.), если присутствует флаг переноса, то подпрограмма ожидает завершения предыдущего звукового сигнала, если - нуль, то подпрограмма ожидает завершения заданного сигнала
F84BH	-	-	Поиск файла, имя которого задано подпрограммой F848H
-	C84BH	HL, DE, BC	Установка текущего номера цвета фона и изображения в порт цвета - FFF8H, в A - код номера цвета
F84EH	-	-	Загрузка файла, найденного подпрограммой F84BH: в HL - задается начальный адрес загрузки файла
-	C84EH	HL, DE, BC, AF	Полное стирание экрана
F851H	-	-	Удаление файла, найденного подпрограммой F84BH
-	C851H	HL, DE, BC, AF	Генерация звука Монитор 2 (через порт клавиатуры)
-	C854H	HL, DE, BC	Сравнение регистров HL/DE, результат определяется флагами нуля, переноса
-	C857H	-	Включение/выключение режима эхопечати символов на экран/принтер
-	C85AH	-	Переключение плотности записи на ленту и типа принтера

Табл. 2.
Стандартные подпрограммы.

Адрес	Назначение
C60CH	Реальный адрес размещения таблицы знакогенератора
C60DH	
C60EH	Реальный адрес размещения таблицы кодов клавиатуры
C60FH	
C61CH	Заданный (FFBFH) и текущий конечный адрес RAM диска
C61DH	
C61EH	Заданный (0000H) и текущий начальный адрес RAM диска
C61FH	
C62CH	Признак КОИ: КОИ - 8 - 00H, КОИ - 7 - FFH
C62DH	Признак РУС/LAT: LAT - 00H, РУС - FFH
C62EH	Признак фиксированного регистра: верхний - 00H, нижний - FFH
C62FH	Признак включенного режима эхопечати на принтер: D2H, иначе - 00H
C63CH	Заданный номер (B1H) цвета фона и изображения
C63DH	Текущий конечный номер страницы RAM диска (00H...07H)
C63EH	Значение текущего положения курсора
C63FH	
C64CH	Заданная и текущая длительность стандартного звука
C64DH	Заданная и текущая частота стандартного звука
C64EH	Частота и длительность звука, формируемого подпрограммой Монитор 2
C64FH	
C65EH	Цвет черно - белого фона экрана
C65FH	
C65EH	Заданный предельный номер страницы RAM диска (07H)
C65FH	Номер плотности записи на ленту / тип принтера

Табл. 3.
Стандартные служебные ячейки.

Значение	Назначение
07H	Выдача стандартного звукового сигнала, формируемого таймером K580BI53
08H	Перевод курсора влево на одну позицию
0AH	Перевод курсора на строку вниз с установкой в начало строки, если курсор на нижней строке, то осуществляется сдвиг экрана вверх на строку, нижняя строка очищается, и курсор устанавливается в начало строки
0CH	Перевод курсора в начало экрана
0DH	Перевод курсора в начало строки
18H	Перевод курсора вправо на одну позиций
19H	Перевод курсора на строку вверх с установкой в начало строки, если курсор на верхней строке, то осуществляется сдвиг экрана вниз на строку, верхняя строка очищается, и курсор устанавливается в начало строки
1AH	Перевод курсора на строку вниз, если курсор на нижней строке, то устанавливает его в начало строки и сдвигает экран вверх на строку, нижняя строка очищается
1BH	Установка курсора (20H + номер строки), (20H + позиция курсора)
59H	
1FH	Первый вывод кода стирает экран между служебными строками, если следом еще раз, то стираются и служебные строки
87H	Восстановление кода цвета из "стека" (см. ниже)
88H	Запоминание кода цвета, по принципу стека, глубина 2 ячейки
89H	Инверсия кода цвета
8AH	Инверсия фона черно - белого экрана
8BH	Установка светлого фона черно - белого экрана
8CH	Установка темного фона черно - белого экрана
8DH	Запоминание курсора "строки" (название условно)
8EH	Восстановление курсора " строки"
8FH	Очистка верхней служебной строки
90H	Очистка нижней служебной строки
91H	Сдвиг экрана вниз от курсора
92H	Сдвиг экрана вверх от курсора
93H	Запоминание курсора "экрана" (название условно)
94H	Восстановление курсора "экрана"
95H	Вывод на экран имени файла, загруженного с диска
97H	Включение режима КОИ - 7
98H	Включение режима КОИ - 8
9AH	Сдвиг экрана вверх от нижней служебной строки до текущей строки курсора
A0H	Установка цвета изображения, младший полубайт - номер цвета
AFH	
B0H	установка цвета фона, младший полубайт - номер Установка цвета фона, младший полубайт - номер цвета
BFH	

Табл. 4.
Стандартные служебные коды.

4. Отладчик программ - Debug_MX

Загружается по запросу из RAMFOS "загрузчиком" в системное ОЗУ по адресам E800H...F7FFH. Служебные ячейки отладчика размечаются по адресам D2C0H...D39FH. Отладчик имеет два режима работы: первый - директивный, второй - отладочный. И соответственно две стандартные точки входа:

E800h - запуск с инициализацией первого режима.

E803h - запрос второго режима из отлаживаемых программ командой CALL.

В первом режиме на нижней служебной строке выводится список директив, разделенных на группы наклонными черточками [/], количество которых определяет число задаваемых параметров у следующей за ним группы директив:

? ,X,G,J \D,L,R,M \A,S,K,U,N \T,C,F,P,W > F7-FILE

? - просмотр информации о текущем состоянии второго режима, значений бита флажков и значений ловушек.

X - просмотр и изменение текущих значений регистров микропроцессора.

G - запуск второго режима из первого.

J - уход или возврат (например, после срабатывания ловушки и в обход ее) в отлаживаемую программу.

D/addrN/ - дамп (просмотр содержимого) памяти с адреса /addrN/.

L/addrN/ - просмотр содержимого памяти в символьном виде с адреса /addrN/.

R/addrN/ - реассемблирование участка памяти с адреса /addrN/.

Выполнение директив D, L, R приостанавливается клавишей <СТР>, <БК> - продолжение.

M/addrN/ - редактирование участка памяти с адреса /addrN/.

A/addrN/, /N/ - установка начального адреса отлаживаемой программы и скорости (N) ее выполнения (0-7).

S/addrN/, /addrK/ - запись области ОЗУ на RAM - диск.

K/addrN/, /addrK/ - подсчет контрольной, суммы.

U/N/, /LOV/ - установка номера (0...7) и значения ловушки [(0...5) - по адресу, (6) - по коду команд, (7) - по значений аккумулятора.

N/addrN/, /addrK/ - установка интервальной ловушки.

T/addrN/, /addrK/, /addrP/ - перемещение области ОЗУ.

C/addrN/, /addrK/, /addrC/ - сравнение содержимого двух областей ОЗУ.

F/addrN/, /addrK/, /код/ - заполнение области ОЗУ /кодом/.

P/addrN/, /addrK/, /код/ - поиск кода (байта) в заданной области ОЗУ.

W/addrN/, /addrK/, /слово/ - поиск двухбайтового слова в заданной области ОЗУ.

F7 - запрос файла с RAM - диска, загружается директивой <COPY>.

Во втором режиме в верхней служебной строке выводится список оперативных директив изменения флажков микропроцессора:

ИНВЕРСИЯ: F1 - переноса, F2 - нуля, F3 - знака \ F4 - вып. ПП,

где: F4 - автоматическое выполнение подпрограммы.

В нижней служебной строке выводится список оперативных директив изменения формата вывода информации на экран:

TRASIR: 1-COM, 2-SP, 3-CALL, 4-JMP, \ 5-STEP, 6-V.LOV, 7-P.LOV, 8-COMPACT.

1 - трассировка выполнения команд; при $\phi_1=1$ на дисплей выводятся сообщения о значениях регистровых пар и мнемоники выполняемой команды, при $\phi_1=0$ не выводится.

2 - трассировка содержимого вершины стека; $\phi_2=0$ информация выводится в одну строку, при $\phi_2=1$ в две строки. Во второй строке выводится значение указателя стека и три слова из его вершины.

3 - трассировка выполнения подпрограмм; при $\phi_3=1$ отображается вложенность подпрограмм, переход на ПП обозначается символом [→], а возврат [←].

4 - трассировка выполнения команд перехода, при $\phi_4=1$ переходы обозначаются [>].

5 - пошаговый режим выполнения команд; при $\phi_5=1$ выводятся сообщения о текущем состоянии программы без выполнения отображаемой команды. При каждом нажатии <VK> выполняется только одна команда. Нажатие клавиши <пробел> отменяет пошаговый режим на время нажатия клавиши. При $\phi_5=0$ отлаживаемая программа выполняется непрерывно со скоростью, заданной директивой A. Нажатие на любую клавишу, кроме <пробел>, временно приостанавливает выполнение программы.

6 - выбор объекта ловушек; при $\phi_6=0$ ловушка срабатывает при попытке выполнить как команду, так и чтение или запись в ячейку с адресом, совпадающим с ловушкой или попадающим в интервал значений ловушки. При $\phi_6=1$ со значениями ловушек сравниваются значения всех регистровых пар, адреса и операнды 3 - х байтовых команд.

7 - режим прохождения ловушек; при $\phi_7=1$ срабатывание ловушки приводит к прерыванию выполнения программы, при $\phi_7=0$ сообщение о срабатывании ловушки выдается, но выполнение программы не прерывается.

8 - режим однострочной выдачи информации при $\phi_8=1$.

Адрес	Назначение
0000H 0081H	Загрузчик
0082H 042FH	Знакогенератор
0430H 101FH	BIOS (3 кБайта)
1020H 17FFH	RAMFOS (2 кБайта)
1800H 1FFFH	Редактор (2 кБайта)
2000H 27FFH	Дизассемблер (2 кБайта)
2800H 37FFH	Отладчик (4 кБайта)
3800H 3FFFH	Ассемблер (2 кБайта)
4000H BFFFH	ROM диск пользователя (32 кБайта)

Табл. 5.
План ROM - диска.

5. Редактор текстов - Edit_MX.

Запускается из RAMFOS, при этом программа загружается "загрузчиком" из ROM - диска, а системное ОЗУ по адресам E800H...EFFFH, После запуска инициализируется режим ввода - редактирования и в нижней служебной строке выводится список основных директив:

F1-SLIT F2-(СТР)-DEL F3-FILE F4-SAVE F5-QUIT F6-ASM F7-DIZ

F1 - слияние строки, заданной курсором, с верхней. Курсор может быть в любом месте строки. После слияния курсор устанавливается на место стыка строк.

F2 - удаление символа из строки в позиции курсора, <СТР> - слева от курсора.

F3 - запрос файла из RAM - диска, загрузка осуществляется клавишей <BK>, если тип файла соответствует заданному редактором, иначе директивой <COPY>.

F4 - сохранение файла на RAM - диске. Выполняется при условии, если были сделаны изменения в тексте файла.

F5 - выход из редактора, если измененный файл не был ранее сохранен, то редактор предлагает это сделать.

F6 - вызов Ассемблера.

F7 - вызов Дизассемблера.

<←>, <→> - перемещение по строке текста.

<↑>, <↓> - перемещение курсора или текста на экране в режиме "реверсивный рулон".

<⌂> - перемещение курсора в начало страницы. Следующее нажатие - переход на страницу вверх.

<ПС> - перемещение курсора в конец страницы или переход на страницу вниз.

<ТАБ>- табуляция строки. Осуществляется с неравномерным шагом (4, 5, 3, 5, 3..5, 3).

<BK> - если курсор в начале или конце строки, то раздвижка текста для ввода новых строк, иначе разрезание строк.

Буфер строки рассчитан на ввод 126 символов. На экран выводится часть строки длиной 63 символа. В верхней служебной строке отображается, помимо имени программы, имя загруженного файла, а также положение курсора на экране: L - номер строки на экране, C - первый байт соответствует номеру символа в строке, второй байт означает заданную границу строки для автоперевода строки и для выравнивания правого края текста. При автопереводе последний введенный символ переносится в начало следующей строки, кроме пробела. Автоперевод осуществляется, если курсор находится в конце строки.

В редакторе имеется и группа дополнительных директив, иницируемых клавишей <AP2>, список которых выводится на нижней служебной строке:

AP2: F2-delete, L(R)-поиск, S(T/C)-маркер(move/copy), J, G, H, B, K

F2 - задание фрагмента текста и его удаление директивой <AP2> + <BK>.

L - задание фрагмента строки и его поиск - <BK>. Если после заданного фрагмента ввести знак [=] и задать новый фрагмент, то первый фрагмент будет заменен на второй.

R - продолжение поиска фрагмента, заданного ранее.

S - задание фрагмента текста и запоминание его директивой <AP2> + <BK>.

T - перенос маркированного фрагмента текста в позиций, заданию курсором.

C - копирование (многократное) маркированного фрагмента текста в позицию, заданную курсором.

G - запоминание границы строки по текущему положению курсора в строке. Если курсор в начале строки, задается максимальная граница в 128 символов, практически отменяющая автоперевод.

H - задание смещения текста влево; если позиция курсора меньше 65, то смещение задается половиной текущей позиции курсора, в противном случае равным 64. Эта директива позволяет обрабатывать тексты шириной в 126 символов.

J - аннулирование левого смещения текста.

V - просмотр кода (байта) символа в текущей позиции курсора.

K - выравнивание правого края строки по заданной границе строки.

<␣>, <ПС> - переход в начало или конец текста.

<←>, <→> - перемещение курсора в начало или конец строки.

6. Ассемблер - Asm_MX

Запускается из редактора Edit_MX, при этом предварительно программа загружается "загрузчиком" из ROM - диска в системное ОЗУ по адресам F000H...F7FFH. После этого в верхней служебной строке отображается имя программы и имя ассемблируемого файла, а на нижней служебной строке выводится список основных директив:

F1-LIS F2-LIN F3-EXE

F1 - трансляция с выводом полного протокола.

F2 - при запуске программа сразу запрашивает файл связей (тип файла "LIN"). Его находят курсором и подключают клавишей <BK>. При трансляции - на экран выводятся, кроме краткого протокола, строки всех внешних меток. При этом в начале этих строк выводятся уже согласованные значения этих меток.

F3 - трансляция с кратким протоколом; конечный рабочий адрес программы / конечный адрес размещения программы в буфере (0000H...OFFFH) / число обнаруженных ошибок.

В процессе трансляции выводятся так же строки, в которых обнаружены ошибки. При этом в начале строк указывается вид ошибки: р? - повтор метки, т? - метку не понял, к? - команду не понял, о? - операнд не понял.

Данный ассемблер предусматривает два режима работы:

ОДНОАКТНЫЙ - когда формируется программа объемом до 4 кБайт. Этот режим реализуется директивами F1, F3. Файл связей при этом сохранять не обязательно, отменяется клавишей <СТР>.

МНОГОАКТНЫЙ - когда программа объемом свыше 4 кБайт формируется согласованно по частям. Порядок такой: первый файл (любой) транслируется директивой F3 и на "диск" сохраняют полученный файл связей (файл EXE сохранять не надо). Последующие файлы транслируют в любом порядке директивой F2, каждый раз сохраняя получаемый файл взамен предыдущего. Получив конечный файл связей, цикл трансляций повторяют директивой F2 в любом порядке, но при этом файл связей сохранять не надо, а файлы EXE обязательно. Полученные файлы EXE объединяют в общую программу, если это, необходимо.

Метки могут состоять из русских (КОИ - 7) и латинских заглавных и прописных символов или цифр, начинаясь с буквы. Транслятор допускает метки длиной до 6 символов. Наиболее оптимальной рекомендуется длина 3 символа. Комментарии отделяются символом [;].

В тексте можно использовать следующие операторы (псевдокоманды):

ORG - задает своим операндом начальный рабочий адрес программы.

EQU - присваивает метке значение, заданное числом или меткой другого числа.

DB - занесение в память байтов или символов, выделяемых с обеих сторон символом ['].

DW - занесение в память слов, задаваемых числом или меткой.

DS N - резервирует N нулевых байтов в памяти.

END - указывает на конец программы, прерывает трансляцию.

В качестве операндов могут быть использованы константы, арифметические выражения, содержащие знаки [+], [-], имена меток и регистров микропроцессора, десятичные и шестнадцатеричные числа. Десятичные числа, кроме команд RST, обозначаются в конце символом [d], а шестнадцатеричные символом [h] или [H].

7. Дизассемблер - Diz_MX

Запускается из редактора, при этом предварительно программа загружается "загрузчиком" из ROM - диска в системное ОЗУ по адресам F000H...F7FFH. После этого в нижней служебной строке выводится список директив:

F1-ADR F2-MET F3-DIZ F4-LINK.

F1 - удаление адресов после расстановки меток.

F2 - расстановка меток; первый символ (буква) устанавливается программно, два других символа - двухзначное число 00H...FFH. При достижении количества меток до 256 первый символ метки заменяется на следующий по кодовой таблице.

F4 - при запуске программа сразу запрашивает дизассемблируемый файл. Его находят курсором и загружают директивой COPY по адресу 0000H. После загрузки программа переходит в режим ретрансляции. В текстовом буфере редактора в этот момент должен находиться файл данных.

F4 - расстановка меток с подключением файла меток. При запуске программа сразу запрашивает файл меток (тип "MET"). Его подключают директивой <BK>. Все директивы работают независимо друг от друга.

Данной дизассемблер предусматривает два режима работы:

ОДНОАКТНЫЙ - когда ретранслируется программа объемом до 4 кБайт. Порядок следующий:

- создать в редакторе или загрузить с диска файл данных;

- директивой F3 загрузить дизассемблируемую программу и ретранслировать ее.

Полученный текст можно просмотреть в редакторе, сохранить на диске или ленте. Строки дополнять комментариями на этом этапе нельзя, но их можно вставлять в виде отдельных строк.

- директивой F4 расставить метки.

- директивой F1 удалить адреса по мере необходимости.

МНОГОАКТНЫЙ - когда программа свыше 4К ретранслируется согласованно по частям: на первом этапе необходимо программу разбить на части по 2...3 кБайт. Для каждой части создать файл данных. Затем директивой F3 сформировать первичные тексты (без меток) и сохранить их на диске. На втором этапе необходимо сформировать общий файл меток. Первичный получают директивой F2, сохраняя его на диске (расстановку меток при этом надо отменять <СТР>). Последующие получают директивой F4 с подключением предыдущего и сохранением взамен предыдущего. На третьем этапе расставляют метки в первичных текстах в любом порядке директивой F4, отменяя при этом сохранение файла меток.

ФАЙЛ ДАННЫХ в виде символьных строк готовится в редакторе: каждая строка в самом начале содержит пару адресов через запятую (начальный и конечный). Первая пара адресов - область рабочих адресов программы. Все последующие пары - области данных. Их следует указывать в порядке возрастания адресов. Они не должны накладываться друг на друга. Если текстовые строки сосредоточены в одном блоке, то можно указывать парой адресов блок данных. Программа сама их разделит по нулевым байтам. Допускаются комментарии после [;], пустые строки. Максимальное число областей данных (блоков) для одного файла - 56. В конечном итоге все метки будут согласованы и расставлены во всех частях программы.

8. Мониторы - Magic_Ж2, SP580

Адаптированные мониторы отличаются от оригиналов скорректированными адресами обращений к внешним устройствам - портам клавиатуры, дисководу, таймера и программатора. А также дополненные: библиотеками директив (вызываемые директивой) и справочной информацией из двух экранных страниц; директивой адаптации работающих с ними программ; директивой ввода (коррекции) адресов портов, используемых в конкретных программах. При адаптации корректируются операнды следующих команд: LDA(3AH), STA(32H), LXI H(21H). В таблице приведены области адресного пространства портов используемые в мониторах и в RAMFOS.

Порт	Magic_Ж2	SP580	RAMFOS
Клавиатура	F800H...FFFFH	F000H...F7FFH	FFE0H...FFE3H
Программатор	F000H...F7FFH	E800H...EFFFH	FFE4H...FFE7H
Дисковод	E800H...EFFFH	-	FFE8...FFEBH
Таймер	-	E000H...E7FFH	FFECH...FFEFH
Цвет	-	-	FFF8H

Табл. 6.
Области адресного пространства портов.

9. Basic_MX

После запуска инициализируется директивный режим. При этом в верхней служебной строке выводится имя загруженной программы, а в нижней - список директив:

F1-HELP F2-AVN F3-FILE F4-SAVE F5-EDITS F6-LIST F7-RUN PC-CLIN

F1 - вызов дополнительных функций из программы, загружаемой в системное ОЗУ с адреса D300H, командой CALL D303H.

F2 - автонабор операторов.

F3 - запрос файла.

F4 - сохранение файла.

F5 - перевод в режим редактирования строки, выбранной на экране курсором или введенного с клавиатуры номера строки.

F6 - распечатка текста программы со строки, указанной курсором или с введенного номера.

F7 - запуск программы с введенного номера или с номера строки (курсора).

PC - раздвижка строк на экране для вставки новой.

10. ТЕСТ_МХ

Тестируется три объекта: основное ОЗУ, RAM - диск и клавиатура. Тестирование ОЗУ осуществляется в два этапа: быстрый - позволяет выявлять явно неисправные ИМС памяти; медленный - позволяет выявлять "хитрые" неисправные ИМС памяти. Результаты тестирования выводятся на экран (в цвете) как показано на рисунке. Расположение микросхем соответствует их расположению в базовой схеме. Режим тестирования сопровождается короткими звуковыми сигналами скользящего тона через порт клавиатуры. При обнаружении неисправной микросхемы выдается двухтональный сигнал. Тест зацикливается, а на экране эта микросхема маркируется светлым цветом ("дырка"). Если ОЗУ исправно, то запускается тест ROM - диска. Содержит один этап - медленный. По завершении цикла тестирования включается режим тестирования клавиатуры, который сопровождается одно - тональными звуковыми сигналами. Нажатие клавиши должно породить двухтональный сигнал. Клавиша <HP> прерывает тест клавиатуры и запускает тест ОЗУ.

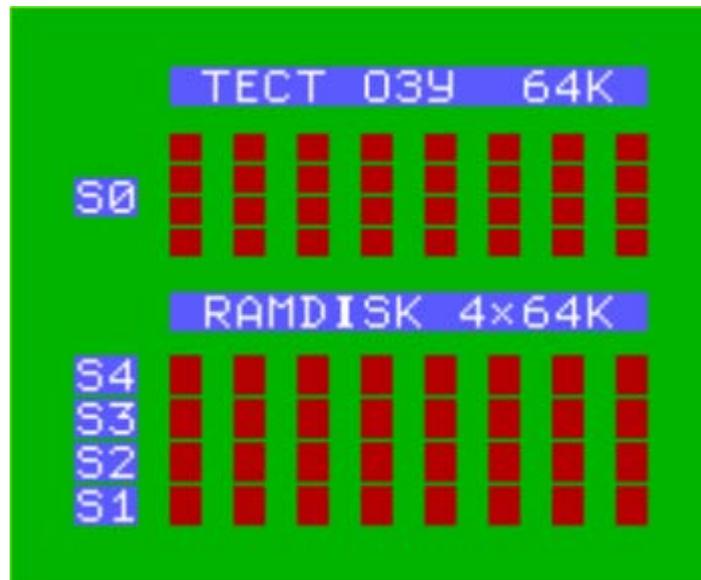


Рис. 1.
Результат работы ТЕСТ_МХ.