

## Архитектура компьютера Sprinter.

### Введение.

Данное описание предполагает наличие определенных знаний читателя, а именно знание архитектуры компьютера ZX-Spectrum и их разновидностей, в частности Pentagon-128 и Scorpion-256, а так же знание языка BASIC и некоторое знакомство с языком ассемблера Z80.

Здесь я буду называть конфигурацией машины - конкретную реализацию конкретной схемы в перепрограммируемой логической микросхеме (ППЛМ). Это означает, что машина имеет множество конфигураций, каждая из которых имеет свою схему.

Я так же использую понятие КЭШ-ОЗУ. Это не КЭШ в формальном смысле, а быстрое ОЗУ, в котором процессор может работать на высокой частоте без ожидания. КЭШ-ем это ОЗУ называется только по традиции, подобно КЭШ-у на KP537PY10 в компьютерах Pentagon-128.

### Краткие данные компьютера Sprinter.

Процессор ..... Z84C15  
Тактовая частота ..... 21MHz/3.5MHz  
ОЗУ ..... 4096Kb  
КЭШ ОЗУ ..... 64Kb  
ПЗУ ..... 128Kb  
Видео-ОЗУ ..... 256Kb(512)  
Контроллер дисков ..... Kp1818BG93  
Поддержка 1.44Mb формата ... 3.5" диска  
Контроллер винчестера ..... IDE/AT  
Контроллер клавиатуры ..... 101key/AT  
Контроллер мыши ..... MS-Mouse  
Два слота ..... стандарт ISA-8  
Аппаратная эмуляция AY-3-8910 стерео-OUT  
COVOX ..... 8bit x 4chanel  
Видео-режимы: ..... Spectrum standart  
GRAF 320x256x256, 640x256x16, TXT 80x32  
Выход видео на TV или CGA монитор, RGB

### Техническая реализация.

Ядром машины являются процессор Z84C15 и ППЛМ EPF10K10QC208. Кроме них на плате присутствуют микросхема ПЗУ, 72х-пиновый SIMM на 4Mb, 256Kb видео-ОЗУ, 64Kb КЭШ-ОЗУ, схема контроллера дисководов на БИС KP1818BG93, буферы для подключения джойстика, магнитофона, принтера, клавиатуры, дисководов, винчестера, мыши, буферные микросхемы шины ISA-8 и еще одна ППЛМ фирмы ALTERA - EPM7032LC44. Эта ППЛМ не меняет своей конфигурации и предназначена для обеспечения синхронизации и начального запуска компьютера. На плате так же предусмотрена возможность подключения CMOS часов на основе микросхемы DALLAS. Кроме периферии и буферов имеются микросхемы дешифрации, входы которых подключаются к процессору через ППЛМ. Это позволяет легко менять адресацию устройств без какого либо изменения разводки печатной платы.

### Возможности архитектуры машины.

Схема компьютера основана на большой перепрограммируемой логической микросхеме. Подключение периферийных устройств через ППЛМ позволяет получить высокую гибкость машины по конфигурациям.

Программирование ППЛМ осуществляется в момент включения, а так же при перезагрузке, что позволяет кардинально менять схему в ППЛМ непосредственно во время работы. Это сильно выделяет архитектуру компьютера из ряда существующих компьютеров и поэтому многие понятия, присущие обычным машинам, меняют свой смысл. Фактически компьютер имеет изменяемую архитектуру, в которой возможны изменения во многих частях схемы. Так, например, нельзя говорить о конкретных адресах портов подключения периферии, так как они могут быть изменены в одну секунду путем перепрограммирования ППЛМ и данных в ОЗУ, отвечающих за конфигурацию портов. Конкретные адреса появляются только в конкретных конфигурациях, например, такой как конфигурация ZX-Spectrum.

Перепрограммируемость схемы дает довольно большую свободу фантазии программиста по конфигурации машины. Задумывая конкретную работу программист может определить в какой конфигурации ее можно сделать лучше, а, возможно, и придумать свою конфигурацию, которую затем можно реализовать в ППЛМ и включить перед запуском этой программы.

Примерная блочная схема компьютера Sprinter изображена на рис. 1

Для простоты некоторые буферы и дешифраторы на схеме не указаны. Количество проводов в шинах так же условны. Часть сигналов управления устройств с ППЛМ объединены с адресами SIMM-а.

Дальнейшее описываются конкретные конфигурации.

В данный момент Sprinter имеет несколько конфигураций, две из которых записаны в ПЗУ, а остальные могут быть подгружены с дискеты или винчестера. Постоянно ведется совершенствование конфигураций и разработка новых.

#### Конфигурация Sprinter-1.

Включает в себя конфигурацию Spectrum-128/256, распределение памяти до 4Mb, расширенный экран с режимами Spectrum, Text-80x32, Graf-320x256 256 цветов на точку (палитра 16 млн. цветов), контроллер дискового, контроллер IDE винчестера, контроллер клавиатуры AT, подключенной как ZX-Keyboard, 8-bit COVOX.

Эта конфигурация максимально приближена к конфигурации ZX-Spectrum и позволяет работать на обычных спектрумовских программах и постепенно менять их под расширенные режимы экрана и памяти, а так же для работы с новыми устройствами.

#### Конфигурация Sprinter-2.

Включает в себя конфигурацию Spectrum-128/256, распределение памяти до 4Mb, расширенный экран с режимами Spectrum, Text-80x32, Graf-320x256 256 цветов на точку (палитра 16 млн. цветов), контроллер дискового, контроллер IDE винчестера, контроллер клавиатуры AT, подключенной как ZX-Keyboard, ускоритель операций с ОЗУ( далее акселератор) .

Эта конфигурация, также как и Sprinter-1, приближена к спектрумовской, но имеет более жесткие требования к программам по совместимости. Акселератор повышает скорость пересылки блоков данных и заполнения ОЗУ одним байтом до физического предела скорости основного ОЗУ.

#### Конфигурация ZX-Spectrum-256/AY.

Эта конфигурация максимально приближена к ZX-Spectrum-128/256 и включает в себя схему музыкального сопроцессора AY-3-8910. В этой конфигурации отсутствуют расширенные режимы экрана.

Сейчас схема AY включает в себя три генератора голосов, генератор шума и регуляторы амплитуды. Генератор огибающей отсутствует. Так же отсутствует возможность чтения из портов данных сопроцессора.

Далее предполагается данные недостатки исключить.

#### Конфигурация Sprinter-3.

Конфигурация отвязана от конфигурации ZX-Spectrum. Полностью отключается ПЗУ и все адресное пространство разбито на четыре окна по 16k, в каждое из которых подключается любая из 256-ти страниц ОЗУ. Отсутствует спектрумовский экран, графический экран такой же, как в конфигурациях Sprinter-1 и Sprinter-2. Имеет дополнительные функции акселератора. Позволяет производить операции AND, OR и XOR с блоками данных. Имеет 8-bit COVOX.

В дальнейшем предполагается подключение в этой конфигурации звуковой карты.

#### Конфигурация Game-1.

Похожа на конфигурацию Sprinter-3. Акселератор не имеет логических функций, а для вывода звука имеет COVOX-Blaster - COVOX с буферным ОЗУ, позволяющим выводить звук поблочно и освобождать процессорное время для другой работы. Конфигурация ориентирована на использование в играх для Sprinter-a.

#### Схема распределения памяти.

Схемы распределения памяти в конфигурациях Sprinter-1 и Sprinter-2 одинаковы и достаточно прозрачны. Фактически она представляет собой схему распределения памяти компьютера Scorpion, с наложенной на нее дополнительной схемой, которая позволяет произвольно устанавливать все страницы памяти, как ПЗУ, так и ОЗУ.

Каждая страница ПЗУ или ОЗУ имеет свой порт, в котором указывается действительный номер страницы из 256-ти страниц всех 4Mb. Страницы, проецируемые в различные окна адресного пространства процессора имеют свои собственные порты. Т.е. Страница, включаемая в адреса #4000..#7FFF, и страница номер 5 обычного Spectrum-овского распределения памяти, включаемая в адреса #C000..#FFFF имеют отдельные порты.

Всего таких портов страниц памяти - 32. Из них 16 портов отвечают за номера страниц ОЗУ, подключаемые к адресам #C000..#FFFF. Еще три порта отвечают за подключение страниц ОЗУ к адресам #0000..#3FFF, #4000..#7FFF и #8000..#BFFF. Восемь портов используются для подключения различных страниц ПЗУ. Один порт - для подключения страницы КЭШ-а вместо ПЗУ. И один порт - это порт системного ПЗУ, подключаемого на место ПЗУ сразу после сброса машины по клавишам Ctrl+Alt+Del.

Оставшиеся 3 порта страниц памяти остаются на данный момент в резерве. Схема распределения памяти позволяет подключить в адресное пространство процессора не только ОЗУ или ПЗУ, но и порты и память ISA карт, вставляемых в слот.

При подключении в адреса #C000..#FFFF скорпионовских расширенных страниц ОЗУ, на их место можно переадресовать слоты. Для этого надо просто записать в порт одной из этих страниц значение, соответствующее ISA-слоту, к которому необходимо произвести обращение. Это значение так же указывает к чему ведется обращение, к портам или памяти.

Схемы распределения памяти Sprinter-a изображена на рис. 2

В других конфигурациях схема распределения памяти упрощается для освобождения ресурсов ППЛМ. Могут отсутствовать порты #1FFD и #7FFD, а так же упрощается схема работы с устройствами, отображаемыми на память.

#### Схема распределения портов.

Sprinter имеет две обособленные группы портов. Первая группа, это внутренние порты процессора Z84C15, вторая - внешние порты. Адресация портов первой группы не может быть изменена, так как эти порты на одном кристалле с

процессором. Вторая группа подключается через ППЛМ и их адреса могут изменяться как угодно, с единственным условием, непересечения с адресами первой группы.

О том какие порты имеются на кристалле Z84C15 можно прочитать в документации по этому процессору и здесь я упомяну некоторые из них. Один из последовательных портов используется для ввода данных с активной мыши. Один из параллельных используется для вывода данных, на второй параллельный порт заведены сигналы прерываний и запросов прямого доступа со слотов ISA. Параллельный порт процессора Z84C15 устроен таким образом, что на нем возможна организация прерываний по сигналам приходящим через параллельный порт. Фактически второй параллельный порт используется как контроллер прерываний.

Схема распределения портов второй группы имеет свою особенность. Главной идеей было получение возможности быстро изменять конфигурацию портов без перегрузки ППЛМ. Это достигнуто путем применения карты распределения портов, располагающейся на специальной странице ОЗУ.

При появлении цикла обращения к порту сначала происходит обращение к ОЗУ карты портов. В карте портов записано какой именно порт подключен к данному адресу. Далее происходит внутренняя дешифрация по байту из карты портов и обращение к выбранному порту. В режиме нетурбо это происходит без каких либо задержек, а в режиме турбо процессору выставляется сигнал WAIT в зависимости от необходимой длины цикла обращения к порту.

Для подключения к какому либо адресу или отключения от него какого либо порта достаточно открыть карту портов и вписать в нужное место один байт.

В странице карты портов содержится четыре карты, которые могут переключаться через системный порт. Таким образом можно осуществить быстрое переключение конфигурации портов, что может быть полезно при работе Spectrum-овских программ совместно со Sprinter-овским BIOS-ом.

Конкретные адреса портов, используемые в Sprinter-e.

Здесь я приведу адресацию портов для конфигураций Sprinter-1 и Sprinter-2. Сразу отмечу, что эти адреса легко могут быть изменены простой программой, в случае появления такой необходимости.

Стандартные порты.

#FE - RD\_KBD - порт клавиатуры

#FE - WR\_BRD - порт бордюра

#7FFD - порт расширения ZX-Spectrum 128k

#1FFD - порт расширения Scorpion ZS-256

#1F,#0F - RD\_KEMPS - порт джойстика. В конфигурации Sprinter-1 порт #1F аппаратно переадресуется на порт #0F

#BFFD,#FFFD - AY-PORTS - порты AY-сопроцессора (ZX-Spectrum-256/AY)

Не совсем стандартные порты.

#FB,#4F - порт COVOX-a.

Дополнительные 8-битные порты Sprinter-a.

#82 - PAGE0 - страница ОЗУ, подключаемая вместо ПЗУ через порт #1FFD

#A2 - PAGE1 - страница ОЗУ, подключенная по адресу #4000

#C2 - PAGE2 - страница ОЗУ, подключенная по адресу #8000

#E2 - PAGE2 - страница ОЗУ, подключенная по адресу #C000

Здесь надо отметить особо, через порт #E2 можно изменить любую из 16-ти страниц скорпионовского распределения памяти.

#89 - PORT\_Y - вертикальная координата точки на графическом экране

или страница VIDEO-RAM для спектрумовского режима

#C9 - RGMOD - порт режима экрана. Переключает страницы режима экрана.

#3C,#7C - SYS\_PORT - системный порт трогать не рекомендуется

#10..#1F,#EE,#EF,#F0,#F1,#F4 - внутренние порты Z84C15

Порты страниц ОЗУ открыты как на запись, так и на чтение. Это позволяет легко выполнять программы, использующие переключение страниц, а затем возвращать эти страницы назад. При работе BIOS-a все страницы сохраняются.

Дополнительные 16-тибитные порты Sprinter-a.

#xx50..#xx55 - порты HDD - использовать внешними программами не рекомендуется. Функции работы с HDD записаны в ПЗУ.

Скрытые порты Sprinter-a.

Скрытыми являются порты которые недоступны в конкретный момент времени, но могут стать доступными после проведения изменений в карте портов. Их адреса не указываются, так как они могут быть выставлены в любое место.

Порт ПЗУ BASIC48

Порт ПЗУ BASIC128

Порт ПЗУ TR-DOS

Порт ПЗУ EXPANSION

Порт ПЗУ SYSTEM

Через эти порты можно установить новые прошивки ПЗУ. Для этого их достаточно записать в ОЗУ с номерами страниц меньше #80 и записать в соответствующий порт номер этой страницы. При таком подключении страницы Эти страницы будут защищены от записи.

Частично скрытыми, так же являются и порты #7FFD,#1FFD в обычном состоянии они доступны только на запись, но значения, записываемые в эти порты можно прочитать, открыв соответствующие порты на чтение.

В других конфигурациях может отсутствовать часть портов или присутствовать новые порты.

#### Организация видеопамяти и видеорежимов.

Видео-ОЗУ Sprinter-а составляет 256 килобайт. В дальнейшем предполагается его расширение до 512 килобайт, для получения более высоких режимов разрешения.

В режиме Spectrum-овского экрана вся организация такая же как в стандартном ZX-Spectrum. В остальных режимах включается Sprinter-овский экран, структура которого включает в себя Spectrum-овский экран как часть схемы.

#### Устройство экрана.

Весь экран разбит на квадраты, размером в стандартное Spectrum-овское знакоместо. Для каждого квадрата устанавливается свой собственный режим вывода, а так же адрес видео-ОЗУ, откуда производится вывод в этот квадрат.

В каждом знакоместе может быть задан свой собственный режим вывода. В данный момент можно устанавливать такие режимы:

ZX-40 - обычный Spectrum-овский режим с одним битпланом и одним атрибутом на знакоместо.

ZX-80 - Режим, похожий на Spectrum-овский по строению символов, но в каждом знакоместе оказывается два символа, сжатые по горизонтали.

G256-8 - Графический режим. Квадрат представляет собой массив 8x8 точек. В каждой точке задается один из 256-ти цветов, выбираемых из палитры 16 миллионов цветов. Квадраты могут иметь разные палитры. Таких палитр для режима G256-8 - четыре.

G16-16 - Графический режим. Квадрат представляет собой массив 16x8 точек. Каждая точка имеет один из 16 цветов, выбираемых из палитры 16 миллионов цветов. Так же, как и G256-8 в квадрате может быть установлена одна из 4-х палитр. Палитры графических режимов пересекаются друг с другом. Палитра 16-цветного режима это первые 16 цветов из палитры 256-цветного.

BORDER - В квадрат выводится цвет бордера.

BLANK - Квадрат гасится - становится черным.

Объем данных режима квадрата составляет 2 байта, поэтому изменение режима всего экрана сводится к перезаписи 2.5 килобайт данных в видео-ОЗУ.

Подобная структура экрана позволяет легко производить скроллинги как всего, так и частей экрана по знакоместам.

Режим экрана устанавливается при включении, а так же с помощью функций BIOS-а. Функции BIOS-а позволяют открывать на экране графические и текстовые окна в нужных местах и нужного размера.

В BIOS-е имеются функции открытия графического экрана на весь экран 320x256 точек. После открытия этого режима экран представляет собой, набор из 256-ти линий, длиной по 320 байт. Соседние точки в линии - это соседние байты. Переключение линий производится через PORT\_Y, в котором устанавливается номер линии, выводимой на экран. Номера линий считаются сверху экрана, начиная с нулевой.

Для вывода в графический экран так же требуется открыть соответствующую страницу основного ОЗУ. В этой странице будет содержаться копия видеоизображения.

Видео-ОЗУ является теневым ОЗУ, поэтому информация, находящаяся в основном ОЗУ, под которым находится видео-ОЗУ не обязательно будет совпадать с информацией, находящейся в этом видео-ОЗУ. Запись видео-данных может производиться и без перезаписи данных в основном ОЗУ, что оказывается полезным при работе, например, со спрайтами. Для работы со спрайтами так же предусмотрен режим записи в видео-ОЗУ с прозрачным цветом. В этом режиме информация, передаваемая в видео-ОЗУ проверяется на наличие байта #FF. Если этот байт обнаруживается, то цикл записи пропускается и на экране в этом месте остается те данные, какие были ранее. Таким образом на экране можно быстро прорисовывать спрайты, представляющие из себя прямоугольные картинки с "прозрачными" цветами.

Пример программы вывода прямоугольной картинки на экран:

```
PAGE3 EQU #E2
RGADR EQU #89
LD A,#50 ; страница графического видеозкрана
OUT (PAGE3),A ; установить в PAGE3
LD HL,Picture ; адрес картинки (прямые данные)
LD DE,#C040+HorPlace ; положение картинки на экране по горизонтали
LD A,VerPlace ; положение картинки на экране по вертикали
OUT (RGADR),A
LD B,VerSize ; высота картинки
LOOP: PUSH DE ; запомнить положение на линии
PUSH BC ; запомнить счетчик высоты
LD BC,HorSize ; длина картинки
LDIR ; копировать линию
```

```

POP BC
POP DE
INC A      ; следующая координата по Y
OUT (RGADR),A
DJNZ LOOP  ; повторить нужное количество раз

```

Управление режимом вывода на экран (включение вывода с прозрачными цветами, отключение копирования в основное ОЗУ) осуществляется через младшие биты порта страницы графического экрана.

Ускоритель операций с ОЗУ.( Акселератор)

Акселератор предназначен для ускорения операций по пересылке данных или по заполнению ОЗУ одним байтом. Акселератор присутствует в чисто Sprinter-овских конфигурациях и поэтому никак не мешает работе обычных Spectrum-овских программ.

Основой акселератора является быстрое внутреннее ОЗУ в ППЛМ. Операции по пересылке данных производятся путем записи блока данных в это внутреннее ОЗУ, а затем копировании его в нужное место памяти из этого ОЗУ. После одной записи копирование может производиться несколько раз и таким образом можно производить заполнение экрана текстурами.

Для заполнения экрана одним цветом используется другой режим акселератора. В нем вместо копируемого блока данных из внутреннего ОЗУ производится запись данных с шины процессора, которые в этот момент не изменяются.

Блок данных, записываемый в ОЗУ акселератора может иметь различную длину из диапазона 1..256 байт.

Управление акселератором производится непосредственно из программы. Для этого используются команды процессора, которые фактически являются операциями типа NOP.

Это команды LD A,A; LD B,B; LD C,C; LD D,D; LD E,E; LD H,H; LD L,L

Назначение команд следующее:

LD B,B - выключить акселератор.

LD D,D - включить акселератор в режим приема байта размера блока далее следует команда типа LD A,dat, где dat и будет новым размером блока. Если размер блока был установлен ранее, его можно не устанавливать.

LD C,C - операция Fill - заполнение одним байтом. Последующая команда типа LD (HL),A приведет к заполнению указанного ранее количества байт значением A

LD E,E - операция Fill для графического экрана - заполнение вертикальных линий.

LD H,H - rezerved

LD L,L - копирование блока. Последующая команда типа LD A,(HL) приведет к заполнению ОЗУ акселератора данными из адреса (HL),а команда типа LD (DE),A приведет к перезаписи данных из ОЗУ акселератора в основное или видео-ОЗУ.

LD A,A - копирование блока для графического экрана подобна команде LD L,L, но работает с вертикальными линиями экрана.

Пример использования акселератора:

; Считаем, что экранная страница уже открыта по адресу #C000

LD HL,#C040 ; адрес начала линии первого экрана

LD DE,#C180 ; адрес начала линии второго экрана

LD BC,#140 ; длина экрана по горизонтали

DI ; запретить прерывания для работы с акселератором

LD D,D ; включить акселератор на установку размера блока

LD A,0 ; установить размер блока - 256 байт

LD A,A ; установить акселератор на копирование

; вертикальных линий.

LDIR ; копировать

LD B,B ; выключить акселератор

EI ; включить прерывания

Этот отрезок программы произведет копирование всего экрана с одного экрана на другой. Время его исполнения составляет примерно 1.2 INT-а.

Дополнительные функции акселератора появляющиеся в конфигурации Sprinter-3 работают подобным же образом. Для выполнения логических функций используются команды XOR (HL); OR (HL); AND (HL).

Пример кодирования блока в 256 байт.

LD HL,ADRES\_1

LD DE,XOR\_DAT

DI

LD D,D

LD A,0 ; число байт, которые надо кодировать.

LD L,L

LD A,(DE) ; взять блок в ОЗУ акселератора

XOR (HL) ; произвести операцию XOR с данными акселератора

LD (HL),A ; запомнить в ОЗУ результат операции

LD B,B

EI

Скорость работы акселератора ограничивается только физической скоростью работы основного ОЗУ. Определить время работы команды с акселератором можно по такой примерной формуле:

Время работы = время работы команды без акселератора + время работы акселератора

Время работы акселератора = число пересылаемых байт / 7000000 (секунд)

Отключение прерываний во время работы акселератора необходимо, так как в этот момент частично меняется система команд процессора и программа на прерывании не сможет работать нормально.

Мультизадачность и многозадачность.

Здесь я буду подразумевать, что мультизадачность - это работа нескольких программ одновременно, а многозадачность - это возможность быстрого переключения между задачами, но работа только одной.

Мультизадачность требует наличия либо нескольких процессоров, либо механизма разделения процессорного времени. Но в разговорах об этом пропускается еще одно требование, которое всегда выполняется на старых машинах. Это неизменность конфигурации компьютера.

Изменение конфигурации в Sprinter-е это довольно длительный процесс, который занимает несколько десятых долей секунды. Это исключает реальную мультизадачность в глобальном смысле, но не запрещает ее реализацию в пределах одной конфигурации. Более того, какая-то конкретная конфигурация может быть специально разработана для работы в мультизадачном режиме, в режиме разделения времени так что бы минимизировать потери времени на переключение между задачами.

С изменением конфигурации может быть реализована многозадачность. Для такого режима работы время изменения конфигурации несущественно, так как оно сравнимо с временем нажатия на клавишу переключающую задачи.

Работа с расширенным ОЗУ.

Работа с расширенным ОЗУ ведется через функции BIOS-а, но программисту дается полная свобода при прямой работе с портами.

Установка страницы и считывание номера установленной страницы производится с помощью операций IN и OUT в восьмибитные порты, а BIOS предоставляет специальные функции, с помощью которых можно легко определить объем свободного пространства ОЗУ и получить номера передаваемых программе страниц ОЗУ.

BIOS имеет свою область переменных, в странице 0FEh, в которой находится информация о распределении памяти в виде RAM Allocation Table - RAT.

Функции для работы с расширенным ОЗУ.

---

Определение объема памяти.

EMM\_FN0 EQU 0C0h

Выход: BC - объем свободного ОЗУ в блоках по 16k  
HL - объем всего ОЗУ в блоках по 16k

---

Инициализация распределения памяти. Стирание всех RAM-дисков, резервирование системных областей ОЗУ

EMM\_FN1 EQU 0C1h

---

Получить блок памяти N bytes

EMM\_FN2 EQU 0C2h

Вход: B - число необходимых блоков  
Выход: A - идентификатор блока/код ошибки  
CF - признак ошибки ( нет памяти )

---

Освободить блок памяти A

EMM\_FN3 EQU 0C3h

Вход: A - идентификатор блока  
Выход: CF - ошибка нет такого блока

---

Получить страницу N блока A

EMM\_FN4 EQU 0C4h

Вход: A - идентификатор блока, B - номер страницы  
Выход: A - страница, CF - ошибка - нет блока

---

Получить список страниц блока ОЗУ

EMM\_FN5 EQU 0C5h

Вход: A - блок, HL - буфер (256 байт)  
Выход: (HL) - номера страниц. Конец - байт 0FFh  
B - число страниц в этом блоке  
CF - ошибка (нет блока)

---

Получение адресов портов и данных для восстановления  
EMM\_FN6 EQU 0C6H  
Вход: A - номер окна проецирования - 0,1,2,3  
Выход: B - текущие данные порта,  
C - адрес порта окна

---

Получить следующую страницу блока памяти  
EMM\_FN7 EQU 0C7H  
Вход: A - страница ОЗУ  
Выход: A - следующая страница ОЗУ/ L=0FFh - конец

---

---

Работа с HDD

---

Инициализировать HDD  
HDD\_FN0 equ 040h  
Вход:  
Выход: CF - ошибка (считать, что нет винта)

---

Резерв - (рекалибровка)  
HDD\_FN1 equ 041h  
Вход:  
Выход: CF - ошибка

---

Тест интерфейса IDE  
HDD\_FN2 equ 042h  
Вход:  
Выход: рег B - информация о наличии IDE устройств  
bit0 = 1 - есть IDE-master  
bit1 = 1 - есть IDE-slave

---

Резерв  
HDD\_FN3 equ 043h  
Вход:  
Выход: CF = 1 - ошибка

---

Читать BPB винчестера (первый HDD, первая партиция)  
HDD\_FN4 equ 044h  
Вход: HL - адрес в странице 0C000h, A - страница  
Выход: CF = 1 - ошибка

---

Читать сектора с винчестера  
HDD\_FN5 equ 045h  
Вход: HL - адрес в странице 0C000h, A - страница  
IX:DE - абсолютный номер сектора  
B - число читаемых секторов  
\*\* При чтении по ровному адресу и переходе адреса на 0000h производится автоматическое переключение страницы по RAT.; По окончании операции в регистрах винчестера находится номер, последнего считанного сектора  
Выход: CF = 1 - ошибка

---

Писать сектора на винчестер  
HDD\_FN6 equ 046h  
Вход: HL - адрес в странице 0C000h, A - страница  
IX:DE - абсолютный номер сектора  
B - число записываемых секторов  
\*\* При записи с ровного адреса и переходе адреса на 0000h производится автоматическое переключение страницы по RAT.  
Выход: CF = 1 - ошибка

---

Резерв  
HDD\_FN7 equ 047h  
Вход:  
Выход: CF = 1 - ошибка

---

Открытие окна

WIN\_OPEN equ 0B0h

Вход: HL - место на экране по знакаместах (копия в IX+2,3)

IX - 32-байтовый описатель окна

(IX+0) - горизонтальный размер в знакахестах

(IX+1) - вертикальный размер в знакахестах

(IX+2) - положение окна по горизонтали на экране

(IX+3) - положение окна по вертикали на экране

(IX+4) - режим знакомеcта

(IX+5) - дополнительный режим знакомеcта

(IX+6) - положение по X в поле графики (по знакаместах)

(IX+7) - положение по Y в поле графики (по знакаместах)

(IX+8..31) - зарезервировано (переменные окна)

\*\* При открытии окна описатель копируется в системную страницу ОЗУ

Выход: DE - размер окна в символах

A - номер окна

---

(Закрывание окна)

WIN\_CLOSE equ 0B1h

Вход: A - номер окна

Выход: CF - ошибка

\*\* Функция резервирована

---

Копирование данных текстового окна в память

WIN\_COPY\_WIN equ 0B2h

Вход: A - номер окна

HL - размер окна вертикаль/горизонталь

DE - положение окна

IX - адрес буфера для запоминания

Выход: CF - ошибка

---

Восстановление данных текстового окна из памяти

WIN\_REST\_WIN equ 0B3h; Вход: A - номер окна

HL - размер окна вертикаль/горизонталь

DE - положение окна

IX - адрес буфера для запоминания

Выход: CF - ошибка

---

Открытие стандартных окон.

LP\_PRINT\_ALL equ 080h

Вход: B - тип окна, E - открываемая страница режима

0 - спектрумовское окно 32x24

1 - текстовое окно 64x24

3 - текстовое окно 80x32

4 - спектрумовское окно, HL - положение окна

5 - текстовое окно 64x24, HL - положение окна

7 - текстовое окно 80x32, HL - положение окна

8 - графическое окно 0, HL - положение окна

9 - графическое окно 1, HL - положение окна

---

Функции печати для работы с текущим окном.

---

Вывод символов на экран с текущего знакомеcта

LP\_PRINT\_ALL equ 081h

Вход: A - символ,

B - число выводимых символов,

E - атрибут символа

---

Вывод символов на экран с текущего знакомеcта без атрибута

LP\_PRINT\_SYM equ 082h

Вход: A - символ,

B - число выводимых символов,



---

Вывод атрибутов на экран с текущего знакоместа

LP\_PRINT\_ATR equ 083h

Вход: В - число выводимых символов,  
Е - атрибут символа

---

Установка текущего знакоместа в окне

LP\_SET\_PLACE equ 084h

Вход: Е - знакоместо по горизонтали  
D - знакоместо по вертикали

\*\* Превышение границ приводит не к ошибке, а к  
перезапуску сначала, за вычетом полного размера окна

---

Вывод строки символов на экран с текущего знакоместа

LP\_PRINT\_LN equ 085h

Вход: HL - адрес строки  
В - число выводимых символов,  
Е - атрибут символов строки

---

Вывод строки символов на экран с текущего знакоместа без атрибутов

LP\_PRINT\_LN2 equ 086h

Вход: HL - адрес строки  
В - число выводимых символов,

---

Вывод строки символов на экран с текущего знакоместа до разделителя после разделителя выводятся пробелы что бы  
получить В символов

LP\_PRINT\_LN3 equ 087h

Вход: HL - адрес строки  
В - число выводимых символов,  
D - разделитель  
Е - атрибут символов

---

Вывод строки символов на экран с текущего знакоместа без атрибутов после разделителя выводятся пробелы что бы  
получить В символов

LP\_PRINT\_LN4 equ 088h

Вход: HL - адрес строки  
В - число выводимых символов,  
D - разделитель

---

Очистка окна экрана

LP\_CLS\_WIN equ 089h

Вход: DE - положение окна  
В - атрибут очистки  
H - высота, L - ширина

\*\* Производится выводом пробелов с заданным атрибутом

---

Скроллинг части окна вверх/вниз

LP\_SCROLL\_UD equ 08Ah

Вход: В - тип скроллинга 1 - вверх/ 2 - вниз  
D - начальная строка скроллинга  
Е - число скролируемых строк

---

Вывод строки символов на экран с текущего знакоместа до разделителя после разделителя вывод останавливается

LP\_PRINT\_LN5 equ 08Bh

Вход: HL - адрес строки  
В - максимальное число выводимых символов,  
D - разделитель  
Е - атрибут символов

---

Вывод строки символов на экран с текущего знакоместа без атрибутов; после разделителя вывод останавливается

LP\_PRINT\_LN6 equ 08Ch

Вход: HL - адрес строки  
В - максимальное число выводимых символов,  
D - разделитель

---

Заключение.

Работа над Sprinter-ом продолжается. Совершенствуется железо и BIOS. Пишется софт, поддерживающий расширенные режимы работы компьютера.

По всем коммерческим вопросам связанным с приобретением компьютера можно обращаться в фирму "Петерс".

Адрес: 191014 Санкт-Петербург, ул. Восстания, д. 35, оф. 31.

Тел: (812) 327-35-31

E-mail: peters@atlant.ru

По техническим вопросам обращаться :

Fido: Ivan Mak (2:5030/529.24)

E-mail: ivan\_mak@yahoo.com

Схема распределения памяти Sprinter-a.

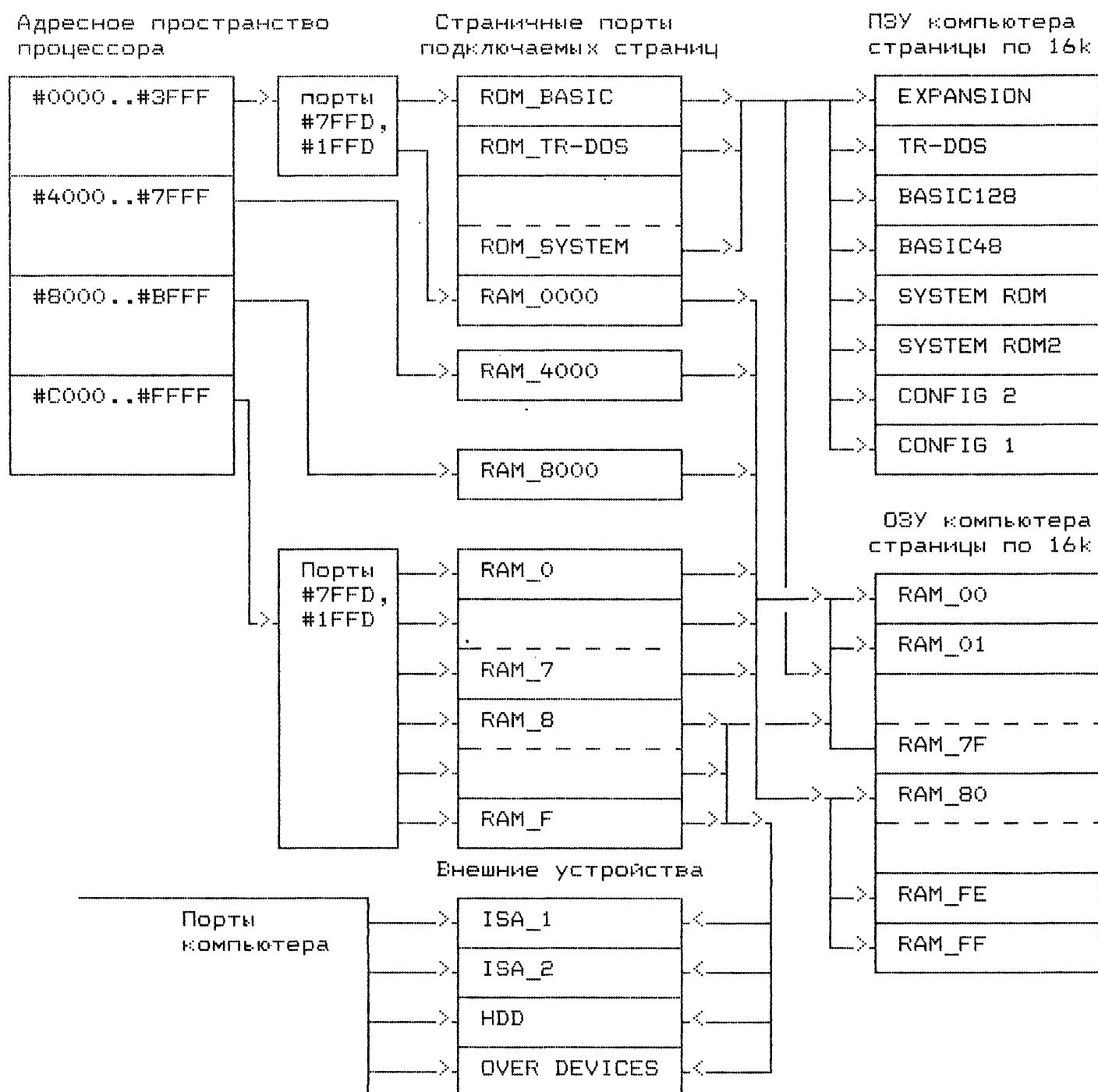


Рисунок 2.