

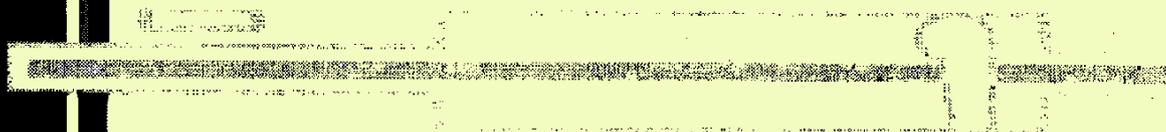


ОРИОНСОФТ
1998

КОМПЬЮТЕР СВОИМИ РУКАМИ

**Радиоловительский
компьютер**

Орион-ПРО



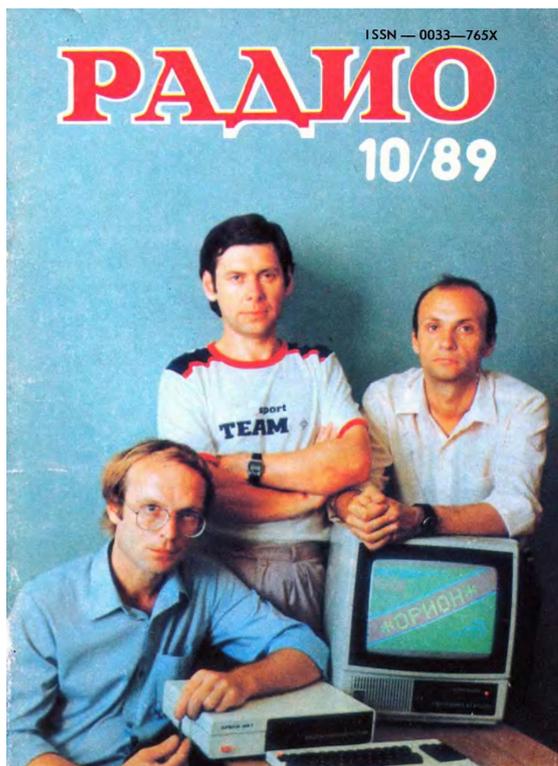
Владислав Петрович Сугоняко

Радиоловительский компьютер "Орион-ПРО"

Первый из семейства «Орионов» популярный радиоловительский компьютер "Орион-128" впервые демонстрировался на 34-й Всесоюзной радиовыставке ДОСААФ 22 мая 1989 года, где и был удостоен первой премии. Как продолжение в его развитии был разработан "Орион-ПРО". Эти компьютеры можно спаять и настроить своими руками. Но самое главное – вы можете припаять на плату еще пару своих микросхем (если возникли идеи!) и это не приведет к катастрофе (сочувствую, если вы пробовали "изобретать" на IBM PC). В этом и есть смысл радиоловительства – возможность творить и воплощать задуманное своими руками
Книга является скорее справочным руководством, чем учебным пособием по ПК "Орион-ПРО".

В книгу внесены изменения в перечень элементов и монтажную схему по версии 3.20, разработанной участниками форума <https://zx-pk.ru/forums/56-orion.html>

От авторов



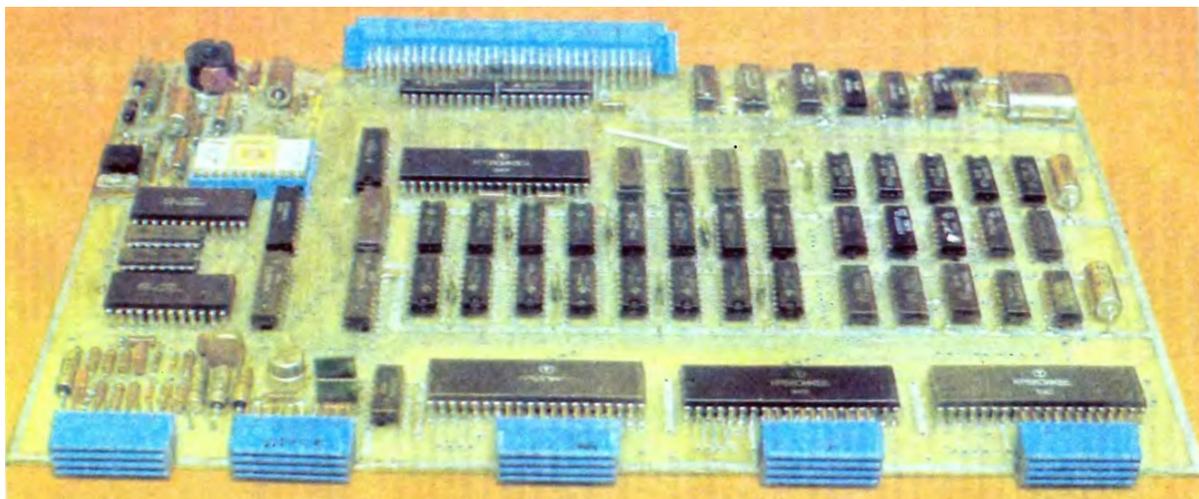
Впервые популярный радиолобительский компьютер "Орион-128" был продемонстрирован на 34-й Всесоюзной радиовыставке ДОСААФ 22 мая 1989 года, где и был удостоен первой премии. Можно смело считать этот день – днем его рождения и точкой – отсчета. Стремительно проходят годы. Это были годы расцвета и спадов. В мир "Орион" пришли талантливые программисты и разработчики, но не обошли его стороной дилетанты, пустословы и откровенные базарные рвачи.

Наше детище выросло целое поколение компьютерщиков. Часть из них осталась верна своему доброму другу "Ориону", а другие перешли на более совершенные машины. И это нормально. Важно что несложный и недорогой, а главное доступный для самостоятельной сборки компьютер помог всем им открыть для себя новый мир – мир компьютеров.

Вместе с тем, многие, кто первоначально бросил "Орион" и ушел в мир "PC" – затем вернулись к своей первой "любви". Все дело в том, что IBM PC это, прежде всего "конторская машина", совершенно не учитывающая дух любительства. Там вы – просто

пользователь, для которого создано все и на все случаи жизни (вот только, к сожалению, не на русском языке). И только небольшой процент "айбизмщиков" становится квалифицированными системщиками и программистами (к сожалению, их творчество зачастую сводится к "расковыриванию шедевров из-за бугра", в лучшем случае 2-3 летней давности). Остальные "наигравшись", вернулись в родные пенаты.

С течением времени становилось все яснее, что "железо" "Ориона-128" несколько устарело. "Западные" компьютеры подсказывали новые идеи и возможности, применимые даже для восьмиразрядных компьютеров. Кроме того предугадать это "старение", еще при разработке "Ориона", было крайне сложно. В 1988 году, а именно, тогда появилась первая версия ПК "Орион-128", процессор Z-80 только появился на "горизонте", в еще не ставшим массовым ПК "Spectrum". Использовать процессор Z-80 очень заманчиво, но KP580BM80A был тогда более массовым и более доступным (для советского радиолобителя!). И стратегический выбор оказался правильным. Несмотря на наличие уже ставших известными в тот период "Радио-86PK", "БК-0010", "Корвет" и др. массовых компьютеров "Орион-128" очень быстро завоевал популярность и определил свою "нишу" – радиолобительскую. Надо прямо сказать, что "разгону" популярности компьютера способствовала марафонная авторская публикация серии статей в почти единственном, в то время, радиолобительском журнале "Радио". Поэтому основная масса ПК "Орион-128" изготовлена пользователями самостоятельно в домашних условиях, "на коленке". Следует отметить, что "Орион-128" производился серийно на Ливенском заводе средств машинной графики (Орловская обл.), где авторы участвовали в подготовке его к производству.



Хочется особенно поблагодарить талантливого конструктора этого завода, энтузиаста "Ориона", Архипова Павла Викторовича. По оценкам разных экспертов, в пик популярности "популяция" ПК "Орион-128" (в СССР!) составляла около 40 тыс. компьютеров.

Вместе с тем, стало очевидным, что в "Орион-128" необходимо вдохнуть новые идеи. При этом авторы оказались перед выбором: произвести модернизацию "Орион-128", или создать новый компьютер, совместимый со старым, но открывающий принципиально новые возможности.

Безусловно, модернизация требовала значительно меньших затрат, но не позволяла внести принципиальные новшества. Ряд талантливых радиолюбителей еще раньше пытались модернизировать "Орион-128" и установили процессор Z-80, ввели другие мелкие усовершенствования. Это известная "Z80 Card2" группы радиолюбителей "SP-club" из С.Петербурга, "Z80-Card от ОрионСервис" из Москвы. Конечно, модернизация компьютера с помощью указанных плат "Card" расширяла возможности компьютера, позволила облегчить адаптацию игровых программ от ПК "Spectrum", но не вносила стратегически новых решений. Следует подчеркнуть, что авторы изначально создавали "Орион-128" для творческих начал, а не как игровой компьютер. Бессмысленно было вести конкуренцию со специализированными для игр ПК "Spectrum" или приставкой "Dendy". У каждого своя ниша. Кроме того, произвести модернизацию "Ориона-128" с целью установки процессора Z-80 могли далеко не все пользователи. Такая работа требовала достаточной квалификации и очень серьезных изменений в плате "Орион-128", особенно при установке "Z80 Card-2". Поэтому многие пользователи просто не решались на это, правильно полагая, что покромсав печатную плату хорошо работающего компьютера, неизвестно что получишь – будет ли он работать или и восстановить не удастся! Анализируя приходящую почту, авторы пришли к выводу, что установку Z-80 на ПК "Орион-128" произвели, в основном, только очень квалифицированные пользователи – электронщики по профессии, что составляло небольшой процент от общего количества имеющихся компьютеров "Орион-128". Понятно, что это не делало погоду.

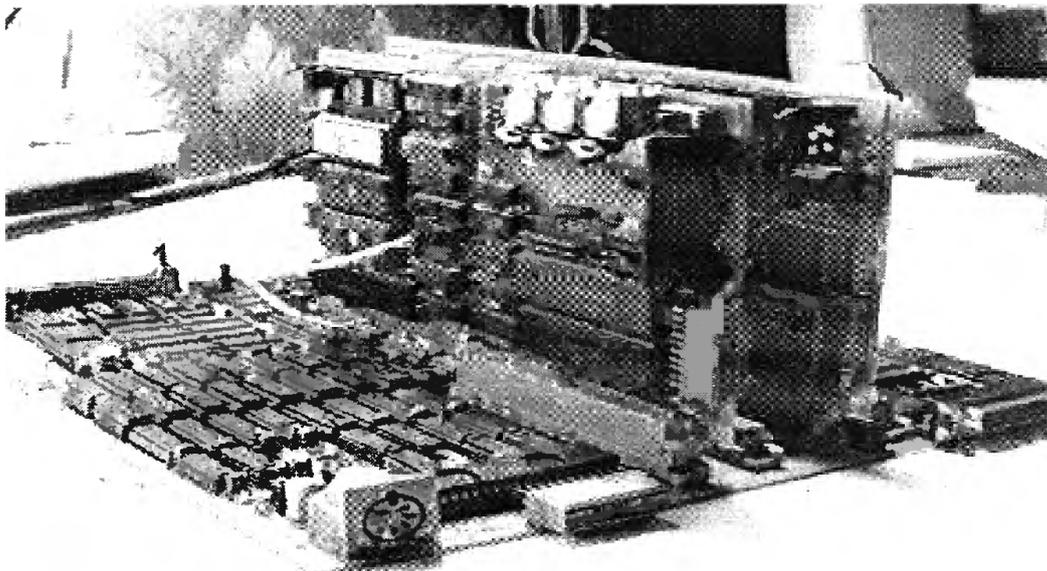
Авторы предпочли второй вариант – создание нового компьютера. Он не ставил пользователя перед вышперечисленными проблемами и выбором "ломать не ломать, а что если не получится". У вас работает "Орион-128"? Прекрасно! На него еще очень долго будут создаваться новые программы энтузиастами. Вместе с тем, в меру своих финансовых возможностей, вы можете собрать новый компьютер, а уже затем решить, что делать с "Оришей".

Итак, новый компьютер.

Мы назвали его "Orion-Pro". Второе название – "Orion-Power" ("Орион-Павер" в переводе – "Мощный Орион"). Из двух этих названий вы можете выбрать то, которое вам больше нравится. Он создан практически новым авторским коллективом, и это принесло свежие идеи и решения.

В свое время, при создании "Орион-128", авторы придерживались девиза "максимально просто и доступно для повторения". Это очень важно на начальном этапе освоения компьютерной грамотности. Вместе с тем, по мере накопления опыта, в таком "железе" становилось тесно и программисту, и пользователю. Поэтому при создании нового компьютера авторы исходили из другого: "прежде всего, максимальные возможности" (для восьмиразрядки), а затем простота (не в ущерб возможностям) и доступность. В то же время устояли против соблазна использовать ROM-прошивки для сокращения "россыпухи" (не в пример "ATM-turbo" и "Profi"), применили расхожую и современную элементную базу.

В структуре "Orion-Pro" (далее – "Pro") как бы два компьютера – старый "Орион-128" и новая архитектура. Совместить их во всем не удалось, поэтому пришлось ввести два режима, переключаемых программно.



Главное отличие "Pro" – это процессор Z80, работающий на тактовой частоте 10 МГц (Z80H). Второе – в старом "Орионе" порты были "положены" на адресацию памяти (F400...F7FF), а в "Pro" – обращение к портам через "IN" и "OUT".

Но самое мощное приобретение – это оконная система памяти – своего рода EMS (по аналогии IBM PC). Она позволила иметь огромную (по меркам для восьмиразрядных ПК) память – до 4 Мбайт. Управление таким большим массивом памяти производится с помощью диспетчера окон памяти. Это понятие уже знакомо пользователям "Z80-Card-2", правда там это сделано было очень упрощенно. "Pro" имеет память 512 Кбайт на основной плате. Она может быть увеличена до 4 Мбайт с помощью внешней интерфейсной платы, которая вставляется в один из системных разъемов.

Окна памяти открываются при выполнении программы в любой странице (банке), т.к. по управлению имеют более высокий приоритет, чем основная память. Это позволяет за рабочую (в режиме "Pro") выбрать страницу 2. Страницы 0 и 1 полностью отданы под экраны (а их четыре). При этом отпадают проблемы с цветом.

"Pro" не имеет привычное ПЗУ 2 Кбайта в области 0F800–OFFFН, где находился "Монитор" в "Орион-128" – там теперь память, допускающая, как чтение, так и запись. Вы можете загрузить Монитор собственного производства, если по каким-то причинам вас не устраивает стандартный.

Вместе с тем, на плате установлены две панельки под ПЗУ большого объема. Одна панелька для стартового ПЗУ (8К) – ROM1–BIOS. Во вторую панельку устанавливается ROM2–BIOS (32/64К) – расширение "ROM1". Если необходимо еще расширить BIOS, то это можно сделать через системный разъем с помощью дополнительной интерфейсной платы. ROM–BIOS содержит стартовый блок, все драйверы устройств (клавиатуры, дисплея, принтера, мыши и т.д.), внутренний служебный командный Монитор (типа "M128"). В ПЗУ расширения (ROM2–BIOS) размещается драйвер дисплея в режиме "Pro", эмулятор "Монитор-2" (который загружается в область 0F800–OFFFН), ORDOS и графический интерфейс (типа "Нортон"), а также PRO–DOS (CP/M–80) и соответствующий "Нортон". Следует отметить, что высокая тактовая частота процессора и шины накладывает ограничения на время выборки применяемых микросхем ПЗУ. Оно должно составлять не более 200–250 нс (можно и больше, однако придется вводить такты ожидания процессора, а значит несколько терять производительность компьютера).

Сохранена возможность иметь четыре экрана. Мало того, их размер может быть как 12К (аналогично "Орион-128"), так и 16К, т.е. 512x256 точек. Область экрана можно разместить в любом из окон памяти, при этом она становится доступной из любой страницы без переключения банков. Тем не менее, внутренний драйвер дисплея может обрабатывать экран в режиме теневого ОЗУ. Это позволяет исключить экранную область из рабочей страницы ОЗУ компьютера при выводе на экран символов и элементов графики.

Как мы отмечали выше, тактовая частота процессора составляет 10 МГц, и может переключаться программно на 2.5 МГц. Следует заметить, что без применения скоростных специальных микропроцессоров Z80Н работать на высоких тактовых частотах проблематично. Отдельные экземпляры процессора Z80В (6МГц) работают на 10 МГц, но перегреваются и требуют применения радиатора Z80 и Z80А – вообще не подходят. Тактовую частоту процессора можно понизить до 5 МГц (при этом Z80В работает в нормальном режиме); для этого на плате установлена специальная перемычка.

Расширен 4–х цветный (двухбитовый) графический режим: в дополнение к нему появились 8–ми и 16–ти цветный (соответственно трех – и четырехбитовый) В "Pro" применяется режим, когда синхронизатор дисплея за один цикл считывает два байта из разных областей памяти одной страницы. Так как в "Pro" имеется две физические линейки памяти, то читается одновременно четыре байта, а значит (учитывая структуру построения графического изображения в "Орионе") можно иметь 4 бита на точку. А это 16 цветов – полноценный EGA! Конечно, размер ОЗУ экрана увеличился в два раза, но используется все те же два банка. Принцип программной обработки экрана не изменился.

Какие еще новшества в "Pro"? Введен регистр для создания псевдоцветного 16–цветного режима. При этом, если используется только один цвет для экрана и один для фона, то нет необходимости занимать в ОЗУ (страница–1) область цвета, как это делает графическая оболочка "NC" или "VC" в "Орионе-128".

Новый ПК имеет восемь аппаратных векторов прерываний, при этом процессор программируется в Режим 2. Вектор 0 (самый высокий приоритет) отдан системному таймеру, который включается через специальный порт. В качестве генератора таймера

используется кадровый синхроимпульс, поэтому прерывания по таймеру происходят каждые 20 мс. Вектор 1 зарезервирован для клавиатуры. Он выведен на универсальный разъем пользователя и на системную шину. Остальные вектора доступны с системной шины и предназначены для интерфейсных плат расширения.

Конструктивно "Pro" создан как многоплатный компьютер, с открытой архитектурой. Состоит из основной ("материнской") платы, и плат расширения интерфейса (как в IBM PC). На основной плате размещается процессор Z80, ОЗУ 512 Кбайт – 16x565PU7 (расширение еще на 3.5 Мбайт – на дополнительной плате), контроллер дисководов. Контроллер работает только с двойной плотностью и допускает подключение двух дисководов на 80 дорожек, как 5.25", так и 3.5" (есть переключатель полярности сигнала "Готовность").

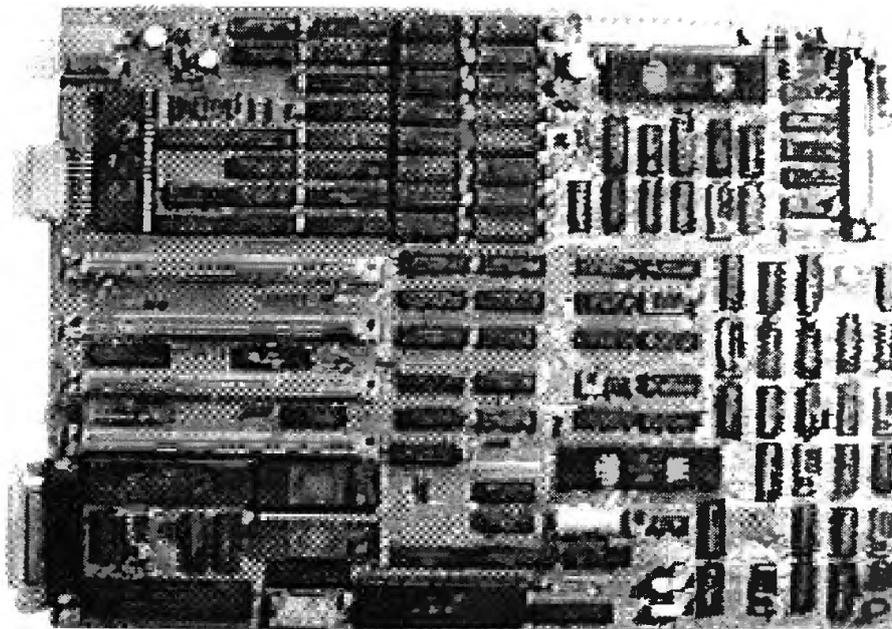
Следует подчеркнуть, что все порты и регистры в режиме "Орион-128" доступны через адреса, положенные на память, и параллельно – через IN/OUT (но по другим адресам). В режиме "Pro" доступ к портам – только через IN/OUT.

На основной плате также установлены четыре системных разъема для плат расширения интерфейса. Один из них имеет дополнительные сигналы для расширения ОЗУ и ПЗУ.

Для подключения принтера на плате установлен свой разъем (такой же, как на IBM PC), программно эмулирующий интерфейс "Centronics". Для подключения цветного и черно-белого дисплеев используется одно семиштырьковое гнездо. Сигналы RGBI выведены и на системную шину, что позволяет "изобретать" различные платы управления цветом и палитрами. Как пример – интерфейсная плата "мультикарта", где и размещается авторская расширенная система управления цветом и палитрами.

В качестве элементной базы использована серия 1533 (увы, серия 155 недостаточна по ассортименту, да и снята с производства).

Всего на основной плате установлено 121 корпус. Плата имеет размер 220x267 мм и может быть смонтирована в корпус MINI TOWER IBM PC при незначительных его доработках.



РАДИОЛЮБИТЕЛЬСКИЙ КОМПЬЮТЕР «Орион-ПРО»



Владислав Петрович СУГОНЯКО

Руководитель группы,

идейный вдохновитель и разработчик архитектуры компьютера, структуры BIOS, платы «мультикарта».



Владимир Олегович ПУШКОВ

Системный программист,

воплотил в «металл» BIOS, произвел адаптацию CP/M-80, написал основные программы для «Орион-ПРО».



Борис Яковлевич ЕВСЕЕВ

Схемотехник,

разработал электронную схему ПК «Орион-ПРО», плату расширения — "квазидиск".

Системная плата

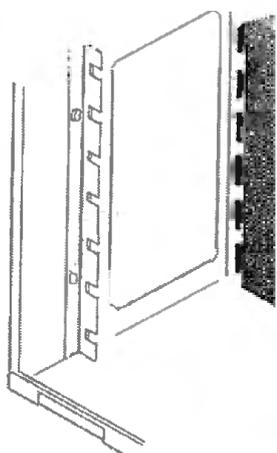
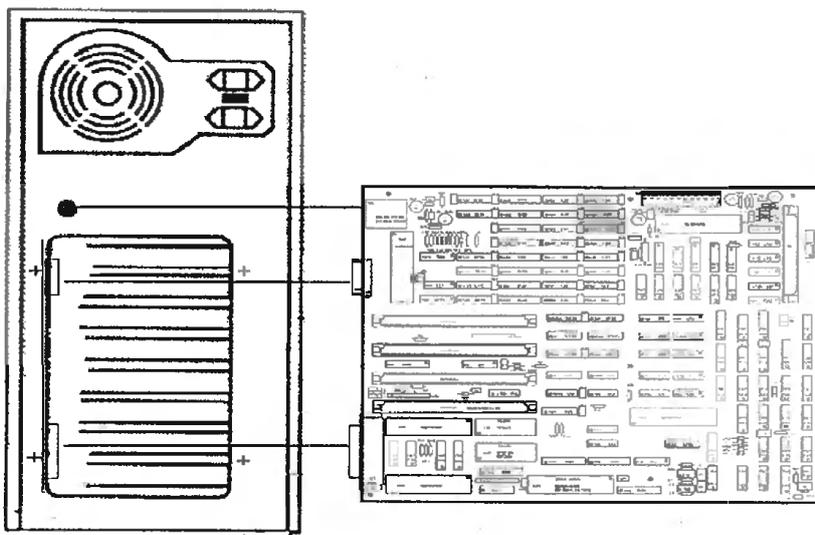
Системная плата Орион-ПРО предназначена для работы с процессорами Z80 фирмы Zilog на тактовой частоте 2.5, 5.0, 10.0 МГц. Ниже указаны основные особенности системной платы:

- Процессор Z80H, тактовая частота 10 МГц, частота шины – 5 МГц
- 512 Кбайт оперативной памяти (поддержка до 4 Мбайт)
- Отсутствие прошивок – микросхем с пережигаемыми перемычками
- Микросхемы серий 1533,580,531
- Встроенный контроллер принтера, разъем "Centronics"
- Встроенный контроллер ГМД, поддержка дисков 3.5"

Размеры

Системная плата имеет размер 267x220 мм и разработана с возможностью установки в стандартный конструктив АТ от IBM PC. Несмотря на то, что внешние размеры платы несколько отличаются от рекомендуемых спецификацией АТ, расположение разъемов ввода/вывода, разъемов расширения и крепежных отверстий максимально подогнаны под имеющиеся отверстия в корпусе.

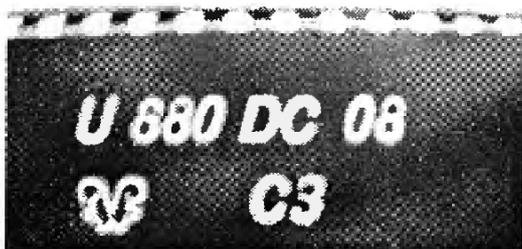
При монтаже платы в корпус последний необходимо доработать. Для этого высверливают четыре заклепки (на верхнем рисунке показаны крестиками), прикрепляющих решетку к корпусу. Если вместо заклепок применена точечная сварка, то сверлят отверстия в местах сварки.



Плату Орион-ПРО устанавливают на стойки откидной панели, подбирая высоту стоек так, чтобы разъем видеовыхода (X2) совпал с круглым отверстием (разъем клавиатуры IBM PC) в корпусе. Для того чтобы платы расширения не качались в разъемах, можно их фиксировать – также, как это делается с платами IBM PC. Фиксацию можно выполнить к примеру, даже проще: прикрепите двумя винтами уголок из пластмассы (можно вырезать от старого пластмассового корпуса транзисторного приемника, или другой подходящей пластмассовой коробки) к отверстиям (с правой стороны задней части корпуса), образовавшимся при высверливании заклепок. В горизонтальной полке уголка, по месту, ножовочным полотном прорежьте пропилы по толщине платы (1,5–2 мм). При установке платы, ее верхний

угол будет надежно фиксироваться в прорези уголка, что исключит раскачивание платы в разьеме.

Центральный процессор



Системная плата разработана для работы с процессором Z80. Процессоры этой серии имеют различные рабочие частоты (2, 2.5, 5, 8 МГц) и не все могут работать в Орион-ПРО. Подходит процессор, имеющий рабочую частоту 8 МГц и более. Такой процессор имеет обозначение Z80H (можно использовать процессоры бывшей ГДР – см. рис.). При затруднении с приобретением высокоскоростного процессора можно применить

процессор Z80B. Для этого необходимо произвести переключение переключки 1 на плате. Конечно, это не позволит реализовать полностью производительность компьютера.

Процессор Z80 обеспечивает обратную совместимость с процессорами I8080 (580), что позволяет применять программное обеспечение компьютеров «Орион128» и «Радио-86РК» (не все).

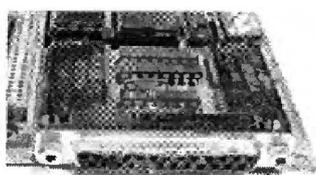
Память

На системной плате устанавливаются шестнадцать микросхем КР565РУ7И (или аналогичны по параметрам, например КР565РУ7А, импортные 41256), что составляет объем ОЗУ до 512 Кбайт. Минимальный объем памяти может составлять 256 Кбайта (одну линейку), но такое включение допустимо только на время отладки платы. Время доступа ОЗУ должно быть не более 120–150 нс. Контроль четности не используется.

Вся память объединена в два блока, или два банка по восемь микросхем в каждом. Банки обозначаются как Bank 0 и Bank 1. Каждый банк обеспечивает 8 разрядов данных. Все микросхемы в каждом банке должны иметь одинаковый тип памяти, объем и время выборки.

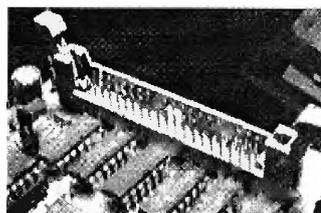
Набор микросхем ПК «Орион-ПРО» выполнен на микросхемах серии КР1533, КР580, КР531.

Порт принтера



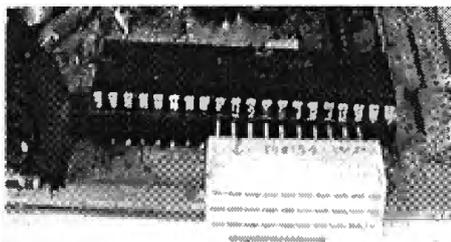
На плате реализован программно-аппаратный порт принтера с интерфейсом «Centronics». При инициализации системы BIOS автоматически конфигурирует порты ввода/вывода. Программно порт доступен через векторы BIOS.

Контроллер гибких дисков



Контроллер гибких дисков «Орион-ПРО» программно совместим с контроллером «Орион-128», а также с вариантами контроллеров «ОРИОН-СЕРВИС» и «SP-Club». Пользователь может подключить два дисководов гибких дисков для работы с дискетами 5.25" или 3.5". Тип используемых накопителей устанавливается при помощи переключки J10 на плате.

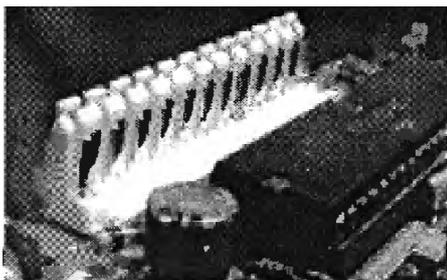
Интерфейс клавиатуры и манипулятора "мышь"



Разъемы клавиатуры расположен на системной плате со стороны задней панели. Может использоваться клавиатура от ПК «Орион-128» или «Радио-86РК без каких либо переделок. На разъем выведена линия 5V и вектор прерывания IRQ1.

Интерфейс мыши выполнен на плате «мультикарта» и рассчитан на применение мыши с параллельным интерфейсом от ПК «Корвет», «БК-0010».

Разъём блока питания



Компьютер ориентирован на использование с блоком питания от IBM PC, поэтому на плату Орион-ПРО устанавливается соответствующий разъем. Если есть проблемы с фирменным разъемом, то можно применить подходящие соединители от отечественных ножевых разъемов.

Для питания компьютера можно использовать другой подходящий источник питания (при создании самодельного корпуса), при этом разъем питания

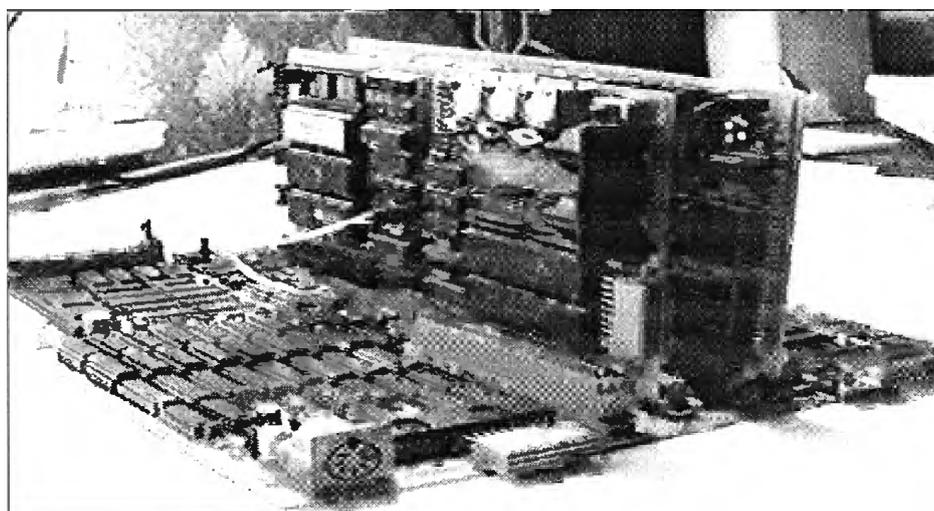
устанавливается по усмотрению пользователя.

Громкоговоритель

Внешний громкоговоритель используется для подачи звуковых сигналов об ошибках, возникающих в процессе начального самотестирования системы, а также при звуковом оформлении прикладных программ. Громкоговоритель подключается к разъему J11 и должен иметь сопротивление $>8\text{ом}$. Допустимо использовать различные телефонные динамические головки с подходящим внутренним сопротивлением и громкостью.

Кнопка сброса

Кнопка сброса системы должна иметь два нормально разомкнутых контакта при замыкании которых происходит сброс компьютера и перезагрузка операционной системы. Кнопка подключается к разъему J5.



Принципиальная схема

Основные технические характеристики

Процессор	Z80H (8-10МГц)
Разрядность шины данных	8 бит
Тактовая частота, МГц	2.5/10
Емкость ОЗУ, Кбайт	512-4048К
Емкость ПЗУ, Кбайт	40-2048К
Доступ к ОЗУ	странич/сегментный
Размер экрана, точек	384x256/512x256
Количество цветовых режимов	6
Внешняя память	НГМД 3.5", 5.25", НЖМД, ROM/RAM-диск
Число векторов аппаратных прерываний	8
Тип клавиатуры	MC7007 / PK-86
Интерфейс принтера	Centronics
Число системных разъемов для плат расширения интерфейса	4
Элементная база (основная)	серия KP1533

Схему компьютера "Orion-Pro" можно условно разделить на несколько основных модулей, каждый из которых хотя и не является законченным функциональным модулем, но выполняет определенные функции. Опорный генератор

(приложение, лист 1) - выполнен на микросхемах D101 и D26. Формирует сигналы частотой 20 МГц (на выходе 4 D101) и противофазные сигналы "64", "65" частотой 10 МГц (на выходах 9,8 D26).

Синхронизатор дисплея -

(приложение, лист 1) - микросхемы D84, D2, D3, D6, D7, D10/3,6,8*, D102/44, D105/4,6, D103/12, D88, D104/8, D118/8,3,11, D17/8, D107/8, D18/1, D28/5. Микросхемы D84, D2, D3, D6, D7 формируют адресные сигналы для непрерывного вывода информации из экранной области ОЗУ на дисплей, а также регенерацию всех страниц динамической памяти. На выходе D103/12 должны присутствовать импульсы строчной развертки (цепь "63") частотой 15 КГц. Сигнал "42" имеет период 200 нс. Частота сигнала кадровой синхронизации, цепи - "60", "VS" (D17/10), должна составлять 50 Гц (период 20мс). Сигнал "50" имеет период 128 мкс. На выходе D28/5 вырабатывается бланкирующий импульс "110", обеспечивающий формирование рабочего поля экрана.

Формирователь RAS/CAS - (приложение, лист 1) CAS и RAS, необходимые для работы ОЗУ, формируются микросхемами D83/6, D83/8, D112/3, D112/6, D47/6, D47/8. Запрет "лишних" CAS и RAS (для облегчения теплового режима PY7) осуществляют микросхемы D38/3, D38/6, D48/6, D48/8, D112/8.

Переключатель тактовой частоты - (приложение, лист 1) - состоит из микросхемы D29/5,6, D22/3, D22/6, D104/3, D104/6. Он обеспечивает синхронное переключение тактовой частоты процессора 2.5/10(5) МГц (зависит от положения переключки J10).

Синхронизатор ОЗУ - (приложение, лист 1). Микросхемы D24/6, D28/8, D115/8 формируют сигналы признака обращения к ОЗУ (цепи 70, 87). D104/11, D14/8 формируют сигнал (такты задержки) ожидания процессора (WAIT) при обращении к ОЗУ. Реальная тактовая частота процессора при этом снижается с 10 до 8.7 МГц из-за наличия тактов ожидания.

Узел тактов ожидания - (приложение, лист 1) - формирует сигналы задержки (микросхемы D14/3, D87, D22/8,11) на один такт при обращении к портам и медленным ПЗУ. Микросхема D22/8 используется для формирования задержки при обращении к портам, D22/11 - к ПЗУ. Если используется ПЗУ (D67, D68) со временем выборки не более 200 нс, то такты ожидания ПЗУ можно отключить отсоединением диода VD4.

Дешифратор портов ОРИОН 128 - регистров и схема управления областью памяти 0F000-0FFFFH (приложение, лист 1) состоит из микросхем D89, D85, D86/3,6,11, D105/12, D102/1, D25/3,6,11, D82, D99. Перестановкой переключки J9/J12/J11 можно выбрать один из адресов (соответственно 0F400H, 0F500H, 0F600H) универсального порта D69 (к которому подключается клавиатура) для режима "Orion-128". Параллельно с портами "Орион-128" ("положенными" на память) работают

порты "Pro" (при обращении к одним и тем же портам), но только через команды процессора IN/OUT.

Регистры управления дисплеем – (приложение, листы 1,2). D113/2,6 – регистр переключения экранов дисплея. Формирование сигналов переключения экранов обеспечивает D11/11, D11/8, D25/8. "Ширина" экрана определяется логическим уровнем сигнала на выходе D113/5. На D88 собран коммутатор сигналов синхронизации дисплея для "узкого"(384т) и "широкого" (512т) экранов. Микросхемы D110 и D115/5 образуют регистр режима контроллера дисплея (порт 0F800/0F8H).

Контроллер дисплея – (приложение, листы 3,6) – считывает данные из ОЗУ в регистры цвета (IP22). Далее информация переписывается в регистры сдвига (IP10), и затем коммутируются на видеовыход (КП11). Распределение микросхем для различных цветовых режимов следующее:

РЕЖИМЫ	IP22	IP10	КП11	
Монохромный режим	D90	- D50 -	D43	
Псевдоцветной режим	D90	- D50 \	D42	данные
		D49 /		цвет
4-цветный режим (2-х битный)	D90	- D50 \	D43	
	D97	- D49 /		
8-цветный режим (3-х битный)	D90	- D50 \		зеленый
	D91	- D58	D39	красный
	D96	- D51		синий
16-цветный режим (4-х битный)	D90	- D50 \		зеленый
	D97	- D49 \	D39	яркость
	D91	- D58 /		красный
	D96	- D51 /		синий
16-цветный режим с групп. Кодированием цвета	D90	- D50 \	D42	данные
	D97	- D35 /		цвет

Узел центрального процессора – (приложение, лист 2) – шинные формирователи: D79 (данные), D95, D94 (адрес), D75 (сигналы управления). D77 – процессор Z80H фирмы Zilog.

Дешифраторы портов "Pro" – (приложение, лист 2) – микросхемы D93, D100, D23/8, D17/11, D1/11, D14/11, D11/6. Позволяют обращаться к портам через команды процессора IN/OUT. Дешифратор портов 00H-1FH выполнен на микросхемах D76, D121/13, D17/3.

Схема управления прерываниями – (приложение, лист 2) – микросхемы D56, D68 – схема приоритета прерываний, D41 – схема формирования сигналов прерывания от таймера с частотой (кадров) 50 Гц.

Диспетчер окон – (приложение, листы 2,3). Оперативная память имеет страничную (по 64К) и сегментную (по16К) организацию. В адресном пространстве процессора, возможно, одновременное включение 3-х окон ОЗУ, обеспечивающих доступ к любому сегменту памяти. ПЗУ ROM1 (D78) и ROM2 (D67) доступны через окна ПЗУ, по адресам 0-1FFFH и 2000H-3FFFH (соответственно), причем в окне ROM2 можно включить как один из сегментов (всего восемь) микросхемы D67, так и сегменты дополнительных микросхем на плате расширения. Перемычки J6, J7, J8 определяют тип микросхемы ПЗУ "ROM2" (2764, 27256, 27512), а перемычка J9 разрешает ее выборку. При установке дополнительной платы расширения, позволяющей наращивать

(сверх 64К) ROM2 микросхемами ПЗУ и ОЗУ (например. 62256 – ОЗУ с возможностью сохранения информации при выключенном питании), переключатель J9 на основной плате убирается. Номер сегмента "ROM2" (D67), включаемого в окне 2000H–3FFFH, определяется сигналами "1BX" (D80 – порт 09H).

На D119, D66, D16, D107/2,4,6, D23/6, D4, D5, D12 выполнен коммутатор окон. Управление окнами: "1C1", "1C2", "1C3", "1C4", "EROM (1C5)" поступают с D80. Логическая "1" на выходе D111/14 в момент сброса принудительно включает окно ПЗУ "ROM1"(D78), обеспечивая тем самым доступ к программе "холодного" старта.

Регистры сегментов ОЗУ – (приложение, лист 3) – выполнен на микросхеме D55. Распределение портов:

"2Ax" (порт 04H) – адрес сегмента в окне "0" (0000–3FFFH);

"2Bx" (порт 05H) – адрес сегмента в окне "1" (4000–7FFFH);

"2Cx" (порт 06H) – адрес сегмента в окне "2" (8000–BFFFH).

Блок ОЗУ – (приложение, лист 3). Основная оперативная память (512 Кбайт) выполнена на микросхемах KP565PY7И. Расширение ОЗУ до 2 Мбайт выполняется на внешней интерфейсной плате. Доступ к ОЗУ – асинхронный, с использованием циклов ожидания (на тактовой частоте 2.5 МГц – без ожиданий). Управление доступом к памяти выполнено на микросхемах D104/11, D14/8, D24/6, D28/6, D115/8. На D109, D8, D9, D13, D108 собран коммутатор адресов процессора и синхронизатора дисплея для их обращения к ОЗУ. D92, D98 – защелки данных. D82 – дешифратор банков ОЗУ.

Диспетчер управления конфигурацией памяти – (приложение, лист 3). Сигналы ZZ1–ZZ4 (D80) определяют адрес страницы ОЗУ в режиме "Pro" (порт 08H). Регистр страниц для режима "Орион–128" выполнен на микросхеме D114 (порт 0F9/0F900H). Разряды диспетчера управления конфигурацией памяти (порт 0AH) соответствуют сигналам "1Cx" (D80). Коммутация регистров страниц (Орион–128/Pro) – осуществляется микросхемой D36.

Порты принтера и клавиатуры – (приложение, лист 4). Микросхема D81–порт SW(DIP)–переключателей (00H), принтера (01H, 02H), микросхема D69 – универсальный порт, используется как порт программной клавиатуры. В режиме PRO его адрес 18–1BH, а в режиме "Орион–128 (может устанавливаться J2, J3, J4) F400, F500, F600H.

Управление звуком – (приложение, листы 1,6). Звук формируется триггером D26/5 по принципу "flip–flop". Порт – FFH. Формирование через EI/DI, для совместимости с "Орион–128", не поддерживается. Порт FEH (D116) – универсальный ЦАП (как для звука, так и др.целей).

Контроллер НГМД – (приложение, лист 5). Обращение к KP1818BG93 (D33) через порты 10H, 41H, 12H, 13H (а также F700–F703H). Регистр управления контроллером (D117) – адрес 14H (а также F704, F714, F720H). На D101/8, D101/10 собран тактовый генератор с частотой выходного сигнала 8 МГц. Делитель D31 (D31/3 и D31/6) формирует сигналы с частотой 4 и 1 МГц, соответственно. D46 сепаратор сигнала чтения, D27, D1/6 – схема привязки сигнала чтения с дисководом к внутренней тактовой последовательности. D34 – формирователь сигнала записи. D20, D15/8, D15/11, D120/1 – адресный дешифратор. D32, D45, D19 – буферные элементы. При настройке и эксплуатации компьютера необходимо помнить, что пропадание +12V на D33 (вывод 40) приводит к выходу этой микросхемы из строя.

* *Примечание: цифра после косой черты в позиционном номере микросхемы указывает номер вывода выходного сигнала, чем идентифицируется логический элемент в корпусе микросхемы (D1.1, D1.2, D1.3 и т.д.)*

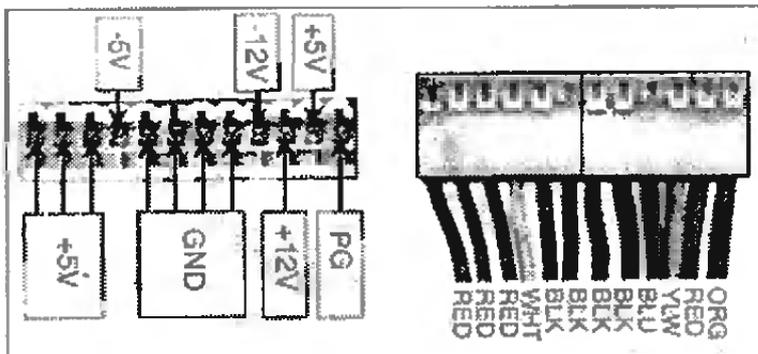
Рекомендации по сборке и наладке

Вы приобрели плату и уже не терпится тут же начать монтажные работы. Охладите пыл. Запаситесь терпением и мощной лупой, внимательно обследуйте плату (к сожалению, стоимость отечественных плат совсем не зависит от их качества). В качестве лупы хорошо подходят объективы от старых фотоаппаратов, имеющие высокую кратность увеличения. Внимательно обследуйте плату с двух сторон при сильном освещении – нет ли "соплей" от лужения или недотравленных перемычек между проводниками, обрывов. Очень тонкие нитевидные перемычки (бич отечественной технологии) хорошо видны на просвет платы. Хуже всего, если эти перемычки остаются под установленными микросхемами или разъемами – придется их выкусывать, чтобы "продрать залипуху". Поэтому не жалейте времени на дотошный осмотр и подготовку платы. Позаботьтесь о защите от электростатического напряжения.



Установите все микросхемы серии 1533 и 531 (1 шт.), резисторы, конденсаторы, транзисторы, диоды и др. мелочь. Установите панельки (желательно импортного производства) под микросхемы: D77, D78, D67, D69, D33. Установите остальные микросхемы. Установите все разъемы и перемычки (джамперы) – два (три) штырька впаянных в плату, на которые одевается замыкающая часть. Не устанавливаются PУ7, "PФ", Z80, ВГ93. J1/2–3 установ. **в положение 2–3**. Проверьте контакты панелек D77, D78 на замыкание между собой. Это долгая операция – проверять следует каждую ножку панельки со всеми остальными.

Далее отладку вести в следующем порядке: 1. Подключить дисплей и подать питание (убедитесь, что нет замыканий по каналам питания).



Проверить температуру микросхем – при наличии замыканий выходов микросхем на шины питания происходит их быстрый нагрев. При правильной работе синхронизатора дисплея на экране появится растр. Нет синхронизации – следует проверить работу счетчиков и смесителя видеосигнала – VD14, VT1 (R30, R31 подобрать уровень видеосигнала для устойчивой работы монитора). На D39/15 должен присутствовать бланкирующий сигнал, а

на D120/6 – ноль. Если другое значение сигнала, то проверить регистры управления режимом дисплея D110 и D115/5,6. На D110/2,7,10,15 D115/5 – "0". На D110/3,6,11,14 и D115/6 – "1".

2. При наличии растра проверить прохождение сигнала с выхода микросхем памяти (PУ7). Для этого поочередно замыкать 2-ю ножку посадочных мест PУ7 первого банка (D52, D71, D54, D57, D70, D60, D61, D59) O3У на общую шину. На экране должны появляться вертикальные линии шириной в одну точку. При их отсутствии проверить работу D90, D50, D43. Нулевой потенциал от 2-й ножки PУ7 записывается в D94 и сдвигается в D50, а через D43 подается на дисплей. Проверьте цепь "Сброс" (J5, D30, D106, D18, D107).

3. Установить D77, D69, в панельку D78 – ПЗУ с тестовой программой. Установить переключатели SW1–8 (D38) в состояние ON (замкнуто). Включить питание и нажать "Сброс". При правильной работе теста должно быть наличие импульсного сигнала на D81/6. На старшей части адресной шины (A8–A15) должен быть "0".

При переключении какого-либо переключателя SW в состояние OFF в соответствующем адресном разряде (SW1=A8 ... SW8=A15) будет появляться "1". В случае отсутствия взаимосвязи переключателей с соответствующим разрядом адреса, проверить:

- наличие сигнала обращения (импульсы) к стартовому ПЗУ D78/20. Проверить диспетчер окон и D111/4 (должна быть "1" и сбрасывается первым обращением к порту "F800"). Служит для принудительного включения при "холодном" старте.

- проверить шины адреса и данных на искажение формы сигнала (полочка, уровни) из-за взаимных замыканий.

Дамп программы Test0

```
0000: 3E 00 ED 47 DB 00 FE 07 28 2A FE 0F 20 F6 21 00
0010: C0 ED 57 16 03 06 00 0E 0F 77 23 0D 20 FB 2F 05
0020: 20 F5 15 20 F0 2F ED 47 21 00 00 23 7D B4 C2 2B
0030: 00 F2 04 00 3E 80 D3 07 D3 00 C3 04 00 FF FF FF
```

4. При правильной работе теста необходимо проверить сигналы "RAS" и "CAS" на ножках 4,15 "PY7" (время нарастания и спада фронтов - не более 20нс). Если все нормально - впаять (лучше установить панельки) первый банк памяти. Необходимо следить за сигналами "RAS" и "CAS" - их нарушение даже в течение короткого времени может вывести PY7 из строя.

5. Установить на SW код 0FH (1-4=OFF, 5-6=ON). Включить питание, "Сброс", и еще раз оперативно проверить "RAS" и "CAS" (на ножках 4,15 PY7). На экране должно быть поле, напоминающее шахматное, с постоянно инвертирующимися ячейками. Клетки размером 8x15 бит вытянуты по вертикали в колонки и имеют сдвиг на один бит. Нарушения картинки указывает на неисправности: обрыв или замыкание линий 201-208, 2090-2091 от мультиплексора к PY7, а также сигналов на выходе D109, D8, D9, D13, D108. Если обрыв или замыкание шины данных, то на экране будут тонкие вертикальные линии. Нажав и удерживая кнопку "Сброс", можно проверить стабильность работы памяти (рисунок не должен разрушаться), а также причину возникновения "мусора" на экране. Если он сохраняется, но перестает "мерцать", то это свидетельствует о нечеткой записи в ОЗУ - мало блокировочных емкостей, пульсации на шинах питания и т.д. В случае исчезновения "мусора" проверить синхронизатор дисплея. Подобное возникает и при использовании вместо KP1533 микросхем других серий (555). Если картинка имеет четкий растр, изображение без разрушений и помех - установить второй банк ОЗУ D89, D74, D72, D53, D73, D85, D82, D84. Запустить повторно тест.

6. Установить на переключателях SW код 07H (1-3=OFF, 4-8=ON). Осциллографом убедиться, что на выводах портов D80, D55 нулевое состояние.

7. Установить микросхему стартового ПЗУ (2764) в панельку D78 (ROM1) и микросхему расширенного ПЗУ (27256 или 27512) в панельку D67 (ROM2). Если микросхемы ПЗУ имеют время выборки менее 200нс, то диод VD4 можно не устанавливать. Установить переключатели SW1=ON, SW2=OFF, SW3=ON (для MC7007), SW4=OFF, SW5=OFF, SW6=ON, SW7=ON SW8=OFF. Установить переключки (джамперы): J1/1-2 для Z80H и 2-3 - Z80B, J2/1-2 (F400H), J5 - кнопка "Сброс", J6, J7/1-2, J8/2-3, J9, J10/2-3, J11 - "Звук" (динамик более 8ом), J12 - для мониторов "Электроника 6105". Включите питание, и нажмите кнопку "Сброс". На экране должно появиться системное Меню. Как им управлять, вы найдете в последующих главах этой книги.

РЕЖИМЫ РАБОТЫ КОМПЬЮТЕРА "ORION-PRO"

Назначение режимов работы компьютера выполняется процедурой "холодного" (начального) старта после его включения (или нажатия кнопки "Сброс" с учетом — состояния DIP-переключателей (SW1-SW8) на основной плате, которые определяют конфигурацию системы.



Каждый из восьми переключателей имеет два устойчивых положения и позволяет выбрать одно из значений соответствующего параметра:

1. Наличие дисковода: On—есть, Off—нет;
2. Наличие жесткого диска: On — есть, Off—нет;
3. Тип клавиатуры: On — MC7007 (основная клавиатура), Off — PK-86;
4. Рабочая страница ОЗУ для внутренней CP/M-80 из ROM2: On — 2, Off — 1;
5. Наличие контроллера символьного дисплея: On — есть, Off—нет;
6. Запуск внутреннего Меню после сброса: On — есть, Off—нет;
7. Тип загрузки операционной системы: On — внутренняя (ПЗУ), Off — внешняя;
8. Режим работы: On — "Pro" (CP/M-80), Off — "Orion-128" (ORDOS)

Примечание: положению "On" соответствует замкнутое состояние (смотри схему) переключателя, "Off" — разомкнутое.

Если переключатель "6" установлен в положение "On", то при "холодном" старте на экране появляется Меню:



Выбор требуемого режима осуществляется с помощью клавиш управления курсором и завершается нажатием клавиши ВК.

ВНИМАНИЕ: Если переключатель "6" установлен в положение "Off", то войти в данное Меню можно, нажав одновременно две любые клавиши сразу после "сброса" (включения) компьютера.

Режим "Monitor"

При назначении данного режима происходит вход в программу Монитор, находящуюся в ПЗУ "ROM1-BIOS". Монитор предоставляет пользователю средства для отладочных и проверочных целей, выяснения конфликтных ситуаций в системе.

Для сокращения занимаемого объема в Мониторе применено командное управление. Команда задается латинской буквой, после которой может следовать до трех числовых параметров, разделенных запятыми. Параметры представляются в шестнадцатиричном формате, при этом незначащие нули можно опускать. Если параметр отсутствует, по — умолчанию предполагается его нулевое значение.

Для выполнения команды после ее набора необходимо нажать клавишу ВК. Исправление неверно набранной команды до нажатия ВК осуществляется клавишей <— или Del (Забой). Для отказа от выполнения команды можно использовать клавишу AP2.

Признаком готовности Монитора к вводу команды является значок => в начале командной строки и мигающий курсор.

Список допустимых команд Монитора высвечивается после его старта в верхней части экрана между двумя горизонтальными линиями.

Команда Modify

Предназначена для модификации (внесения изменений) содержимого ячеек оперативной памяти, начиная с заданного адреса.

Формат: M XXXX

где: XXXX – адрес ячейки.

После ввода данной команды Монитор выводит текущий адрес и содержимое ячейки по этому адресу. Для модификации ячейки необходимо ввести шестнадцатиричный байт, а затем нажать ВК.

Команда Dump

Предназначена для вывода дампа памяти в шестнадцатиричном и символьном виде на экран.

Формат: D XXXX,YY

где: XXXX – начальный адрес, YY – номер страницы ОЗУ.

Если номер страницы ОЗУ не указан, предполагается нулевая страница.

После ввода команды на экране появляется дамп первого блока 256 байт, причем в левой части отображаются шестнадцатиричные коды, в правой – соответствующие символы.

В нижней строке экрана выводится Меню допустимых клавиш для управления. При нажатии клавиш курсора "Вверх" и "Вниз" происходит переход к предыдущему или следующему блоку, а при нажатии клавиши "В" или "Угол" – к начальному адресу, указанному в команде. Для выхода из команды "Dump" необходимо нажать клавишу AP2.

Следует помнить, что по адресам 0000–1FFFH отображаются коды ПЗУ "ROM1–BIOS". Чтобы просмотреть ОЗУ, необходимо эту область открыть в одном из окон (кроме окна 0, перекрывающегося с окном ПЗУ), например:

W1,0 [BK]

D4000 [BK]

Команда Fill

Предназначена для заполнения памяти заданной константой.

Формат: F XXXX,YYYY,ZZ

где: XXXX – начальный адрес, YYYY – конечный адрес, ZZ – константа.

Команда Transfer

Предназначена для перемещения области памяти. Формат: T XXXX,YYYY,ZZZZ

где: XXXX – начальный адрес перемещаемой области: откуда, YYYY – конечный адрес, ZZZZ – начальный адрес перемещения: куда.

Команда Compare

Предназначена для сравнения двух областей памяти. Формат: C XXXX,YYYY,ZZZZ

где: XXXX – начальный адрес области 1, YYYY – начальный адрес области 2, ZZZZ – конечный адрес области 2.

Команда Input

Предназначена для чтения значения порта.

Формат: I XX где XX – адрес порта.

Значение указанного порта выводится в шестнадцатиричном и двоичном виде, например:

IA [BK] (читать порт диспетчера)

50=0100_0000 – (получен результат в виде "HEX=BIN")

Команда Output

Предназначена для записи значения в порт.

Формат: O XX,YY где: XX – адрес порта, YY – значение.

Пример:

0A,58 [BK] – (включить ПЗУ ROM2)

Команда Video

Предназначена для включения/выключения широкого экрана.

Формат: V

Данная команда работает в триггерном режиме. При включении широкого экрана (512 точек) в правой части экрана будет просматриваться служебная область 0F000H–0FFFFH.

Команда Window

Предназначена для включения окна ОЗУ.

Формат: W X,YY где X – номер окна (0–2), YY – номер сегмента (0–1F).

Окна ОЗУ: Номера сегментов по страницам ОЗУ

Окно	Адреса	ПП
Окно 0 – 0000–3FFFH	0 1 2 3 4 5 6 7	Адреса ПП
Окно 1 – 4000–7FFFH	-----	-----
Окно 2 – 8000–BFFFH	03 07 0B 0F 13 17 1B 1F	C000–FFFF
	02 06 0A 0E 12 16 1A 1E	8000–BFFF
	01 05 09 0D 11 15 19 1D	4000–7FFF
	00 04 08 0C 10 14 18 1C	0000–3FFF

Команда XWindow

Предназначена для вывода содержимого портов управления сегментами окон ОЗУ и ПЗУ, состояния диспетчера, а также справочной информации о расположении сегментов по страницам.

Формат: X

По этой команде на экран выводится следующая информация:

```
03 07 0B 0E 13 17 1B 1F
02 06 0A 0E 12 16 1A 1F <== номера сегментов
01 05 09 0D 11 15 19 1D
00 04 08 0C 10 14 18 1C
```

```
03 00=0000_0000 (состояние регистра сегментов ROM2)
02 00=0000_0000 (состояние регистра сегментов Окна 2)
01 00=0000_0000 (состояние регистра сегментов Окна 1)
00 00=0000_0000 (состояние регистра сегментов Окна 0)
DP 50=0101_0000 (состояние диспетчера)
```

Здесь приведено состояние регистров после "Сброса"

Команда "2"

Предназначена для переключения компьютера на тактовую частоту 2.5 МГц.

Формат: 2

Команда "8"

Предназначена для переключения компьютера на тактовую частоту 8.7 МГц.

Формат: 8

Следует иметь в виду, что процессор может работать с тактовой частотой 10 МГц. При обращении к ОЗУ, ПЗУ и портам реальная частота снижается до 8.7 МГц из-за наличия тактов ожидания.

Если используются микросхемы ПЗУ (ROM1 и ROM2) с временем выборки не более 200–250 нс, то такты ожидания ПЗУ можно выключить (для этого отключается диод VD4 на плате компьютера). В этом случае реальная частота процессора при выполнении программ в ПЗУ будет составлять 10 МГц.

Переключатель J1 на плате позволяет устанавливать верхнюю тактовую частоту процессора 5 или 10 МГц (в зависимости от применяемого процессора).

Команда Go

Предназначена для передачи управления по заданному адресу.

Формат: G XXXX где XXXX – адрес перехода.

Режим "ORION-PRO"

Это – основной режим, который позволяет практически в полной мере реализовать как аппаратные, так и программные возможности компьютера "Orion-Pro". Если для работы компьютера в режиме "Orion-128" можно использовать ПЗУ ROM2 объемом всего 8 Кбайт, то в режиме "Orion-Pro" объем ROM2 должен быть равен 32 или 64 Кбайт для размещения драйвера дисплея "TV-PRO" и встроенной операционной системы CP/M-80. Следует сразу оговориться что режим "Pro" – это прежде всего CP/M-режим, имеющий расширенные возможности за счет резидентных драйверов и более упорядоченной архитектуры компьютера.

При работе в 1-й странице часть адресного пространства процессора будет занята экранной областью (атрибуты цвета) ОЗУ. Операционная система OSDOS для компьютера "Orion-128" загружается именно в первую страницу. С одной стороны, это уже стало стандартом и традицией, и многие прикладные программы для компьютеров семейства "Orion" это учитывают в своей работе, с другой – это ограничивает возможности работы с цветом и обработки нескольких экранов. Заметим, что при загрузке операционной системы из ПЗУ в первую страницу ОЗУ автоматически назначается режим "Orion-128".

Так как драйвер дисплея "TV-PRO" обеспечивает работу с несколькими экранами из любой страницы во всех цветовых режимах, проблема доступа к экранной памяти снимается, и в качестве рабочей может быть выбрана 2-я страница ОЗУ. При этом программы пользователя не будут конфликтовать с данными экранов, расположенными в страницах 0 и 1. Операционная система CP/M-80, адаптированная на компьютере "Orion-Pro" для работы во 2-й странице ОЗУ, была названа PRODOS.

Таблица векторов (в области 0F800H) стандартных подпрограмм в режиме "Orion-Pro" отличается от соответствующей таблицы в режиме "Orion-128", поэтому

совместимость между программами, разработанными специально для этих режимов, НЕ обеспечивается.

Стандартные подпрограммы, доступные для программ пользователя через таблицу векторов, составляют основу так называемой Базовой Системы Ввода/Вывода (BIOS), выполняющей прямое взаимодействие с памятью и портами компьютера. В компьютере "Орион-128" вместо "BIOS" часто использовалось понятие "Стандартные подпрограммы Монитора". Таблица векторов BIOS для режима "Orion-Pro" будет описана ниже. Заметим, что в режиме "Orion-Pro" непосредственный доступ пользователя к рабочим и системным ячейкам BIOD не предусмотрен.

Переключатель "7" определяет тип загрузки CP/M-Pro: внутренняя (положение – "On") – загрузка производится из ROM2-BIOS, и внешняя – с системной дискеты. ROM2-BIOS версии 1.00 (размер 32K – 27256) систему CP/M-Pro не содержит.

Если DIP-переключатель "8" находится в положении "On", то режим "Orion-Pro" назначается автоматически.

Режим "ORION-128"

При входе в этот режим открывается окно ROM2, и управление передается по адресу 2000H специальному загрузчику, размещенному в 0-м сегменте ПЗУ (если микросхема "ROM2" установлена). Предполагается стандартом, что в данном сегменте объемом 8 Кбайт (может быть, единственным) записан штатный "Монитор-2" и система "ORDOS" с программой "PC" (Power Commander "Нортон"). Однако, пользователь может разместить там другую систему по своему усмотрению. Следует иметь в виду, что при размещении альтернативной системы (вместо ORDOS) по адресу 2000H должна находиться и своя управляющая программа – загрузчик, который "знает что делать" с Вашей системой.

После загрузки "Монитор-2/PRO" система проверяет состояние DIP-переключателя "7". От его состояния будет зависеть загрузка ORDOS: On внутренняя ORDOS из ROM2 (эта система сама "знает", где она находится и как себя грузить), Off – внешняя (не обязательно ORDOS) – из мультикарты.

Внутренняя ORDOS при обращении к ROM-диску проверяет его первый байт: если его значение равно 00H или 0FFH, то считается, что ПЗУ в панельке отсутствует, если это значение 0C3H (код команды JMP), то предполагается, что у пользователя установлено "старое" ПЗУ из ROM-диска "Орион-128.2", начальные 2K которого содержат ORDOS, а сами программы находятся с адреса 800H. Если же значение первого байта другое (как стандарт определен код 0E7H), то предполагается, что информация записана в НОВОМ формате. Новый формат устанавливает, что программы в таком ПЗУ начинаются с адреса 10H (зачем оставлять 2K вначале, если сама ORDOS находится в другом месте). Содержимое ячеек 1-0FH зарезервировано и должно иметь значение 0FFH. Внутренняя ORDOS автоматически распознает форматы записи информации в ROM-диске, поэтому в панельки на мультикарте можно устанавливать одновременно ПЗУ с разными форматами.

Таблица векторов стандартных подпрограмм "Монитор-2/PRO", доступная пользователю в режиме "Orion-128", в основном совпадает с соответствующей таблицей компьютера "Орион-128", однако имеет и новшества:

- F800 (START) – холодный старт BIOS (вход в Монитор)
- F803 (KBRD) – ввод с клавиатуры
- F806 – не используется
- F809 (TVC) – вывод символа (C) на экран

- F80C – не используется
- F80F (TVA) – вывод символа (A) на экран
- F812 (STTS) – статус клавиатуры
- F815 (HEX) – вывод байта двумя HEX-символами
- F818 (MSG) – вывод текстового сообщения
- F811 (INKEY) – опрос кода нажатой клавиши
- F81E (RCUR) – чтение положения курсора
- F821 (PRINT) – вывод байта на принтер
- F824 (MOVBL) – перенос блока памяти из одной страницы в другую. При этом:
 HL – нач. адрес блока
 DE – нач. адрес "куда"
 BC – размер блока
 A= биты 4–7 — откуда читать (№ страницы),
 биты 0–3 – куда записать (№ страницы).
 После выполнения операции возврат происходит в страницу,
 установленную вектором XPAGE (по умолчанию 0). Адрес возврата
 находится в стеке.
- F827 (FILL) – запись константы (байта) в выбранную страницу ОЗУ:
 HL – начальный адрес
 DE – размер области заполнения
 C – записываемый байт
 A страница, куда записывается байт. После выполнения операции
 возврат происходит в страницу, установленную вектором XPAGE
 (по умолчанию 0). Адрес возврата находится в стеке.
- F82A (CSUM) – вычисление контрольной суммы (по "0–128").
- F82D (UNPCK) – распаковка внутреннего знакогенератора.
- F830 (XPAGE) – установка/чтение рабочей страницы
 Вход: A=0 – читать A≠0 – установить C=номер страницы (0–7)
 Выход: при чтении – A=номер страницы при записи – A=предыдущее состояние
- F833 (RESET) – восстановление исходного состояния рабочих ячеек Монитора.
 Вход: A=0 (с другим A не выполняется) Выход: A= байт-описатель:
- D0 \ тип ПК: 00 – Орион 128.2
 D1 \ 01 – 0–128.2+Card2 (от Чистякова)
 D2 / 02 – 0–128.2+Card (Орион-Сервис)
 D3 / 03 – 0–128.2+Z80Card (Орионсофт)
 04 – Орион-ПРО V3.10 (Орионсофт)
- D4 \ реальная скорость ПК:
 D5 – 00 – 2.5 МГц
 D6 / 01 – 3.5 МГц
 02 – 5.0 МГц
 03 – 6.5 МГц
 04 – 8.5 МГц
 05 – 10.0 МГц
 06 – более 10 МГц

D7 – тип дисплея:

- 0 – электронно-лучевой
- 1 – жидкокристаллический

- F836 (RRAM) – чтение байта из дополнительной страницы
- F839 (WRAM) – запись байта в дополнительную страницу
- F83C (WCUR) – установка положения курсора
- F83F (BEEP) – подача звукового сигнала.

Режим "TEST"

При входе в режим "TEST" на экране появляется подменю:

Configuration

RAM – T e s t

ROM – T e s t

TV – T e s t

Main Menu

При обращении к пункту "Configuration" на экран выводится состояние DIP-переключателей, определяющих конфигурацию системы.

1 Floppy Drive	Yes	(No)
2 Hard drive	No	(Yes)
3 Keyboard	M7007	(RK86)
4 Page CP/M-80	1	(2)
5 Ext.display	No	(Yes)
6 Start-Menu	No	(Yes)
7 Loading	ROM-2	(Extrn)
8 System	ORDOS	(CPM80)

При назначении режима "RAM Test" на экране появляется изображение двух линеек ОЗУ согласно их расположению на печатной плате с их номерами по принципиальной схеме. Внизу отображается шкала хода тестирования. Тестирование включает 4 цикла: заполнение всей памяти нулем; проверка содержимого памяти на 0; заполнение всей памяти байтом 0FFH; проверка содержимого памяти на 0FFH. Тестом не проверяется экранный сегмент ОЗУ (3) и сегмент служебного ОЗУ (31). Собственно программа тестирования расположена в ROM1 и не использует рабочих ячеек в ОЗУ. В случае неисправности микросхемы ОЗУ ее изображение будет закрашено. Выход из режима "RAM Test" – по нажатию AP2 (два раза).

В режиме "ROM Test" микросхемы ROM1 и ROM2 проверяются методом подсчета контрольной суммы и вывода ее на экран. Контрольные суммы ROM1, ROM2 рекомендуется записать на бумаге и при очередной проверке сверять эти значения. Если контрольные суммы со временем изменились или меняются в процессе теста, то это может быть вызвано неисправностью микросхем ПЗУ. Объем ПЗУ ROM2 определяется в процессе теста автоматически (32К или 64К).

"TV Test" – это тест для проверки правильности вывода градаций яркости на монитор. На экране монохромного монитора появится 16 вертикальных полос с

нарастающей яркостью. В левой части экрана монитора будет самая темная полоса, а справа – самая светлая. С помощью этого теста можно производить точную подгонку резисторов ЦАП монохромного видеовыхода.

На экране цветного монитора "TV Test" будет проявляться сначала в виде 8ми монохромных полос (при подключении дисплея к видеовыходу "Мультикарта"), а после нажатия любой клавиши – в виде 8-ми цветных полос в следующей последовательности слева направо: черный, синий, красный, малиновый, зеленый, голубой, желтый, белый. Если последовательность цветов не совпадает, то необходимо уточнить правильность подключения цветного монитора.

Для выхода из подменю "TEST" в главное меню используется клавиша AP2 (если ее нажать при отображении монохромных полос, то видеовыход останется в монохромном режиме).

ОРГАНИЗАЦИЯ ОПЕРАТИВНОЙ ПАМЯТИ

Оперативная память компьютера "Orion-Pro" имеет страничную и сегментную организацию. Управление режимами памяти осуществляется с помощью порта диспетчера с адресом 0AH.

Одна страница ОЗУ имеет размер 64 Кбайта и соответствует максимально возможному объему памяти, непосредственно адресуемому процессором Z80. Каждую страницу условно можно разделить на 4 сегмента размером по 16 Кбайт.

Количество страниц ОЗУ на основной плате равно 8 (их номера 0..7), что соответствует 32 сегментам и объему основной памяти 512 Кбайт. Для выбора номера рабочей (текущей) страницы используется порт с адресом 08H (для режима "Орион-128" – 0F9/0F900H). В режиме "Orion-Pro" все страницы равносильны, и нет необходимости работать именно в нулевой.

Распределение сегментов по страницам основного ОЗУ:

	Стр.0	Стр.1	Стр.2	Стр.3	Стр.4	Стр.5	Стр.6	Стр.7
FFFFH	3	7	11	15	19	23	27	31
C000H	2	6	10	14	18	22	26	30
8000H	1	5	9	13	17	21	25	29
4000H	0	4	8	12	16	20	24	28
0000H								

Доступ к сегментам ОЗУ осуществляется через три независимых окна, которые можно "открыть" в адресном пространстве процессора в пределах рабочей страницы ОЗУ:

Окно ОЗУ "RAM2" – 8000–BFFFH

Окно ОЗУ "RAM1" – 4000–7FFFH

Окно ОЗУ "RAM0" – 0000–3FFFH

Включение (открытие) каждого из окон осуществляется установкой соответствующего разряда D0, D1, D2 порта диспетчера 0AH в единицу, а отключение (закрытие) – установкой разряда в 0.

Назначение разрядов порта 0AH следующее:

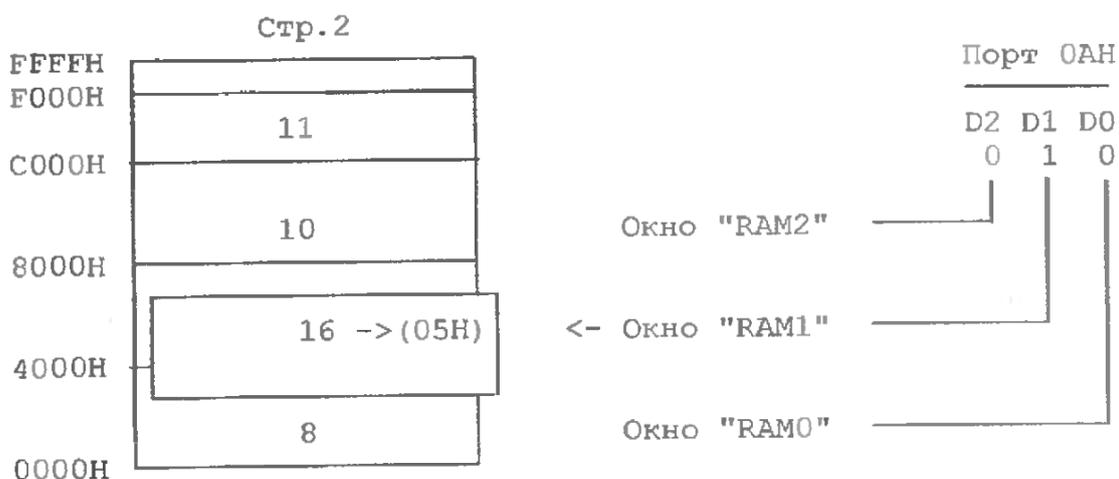
- D0 – включить окно ОЗУ "RAM0"
- D1 – включить окно ОЗУ "RAM1"
- D2 – включить окно ОЗУ "RAM2"
- D3 – включить окно ПЗУ "ROM2-BIOS"
- D4 – включить окно ПЗУ "ROM1-BIOS"
- D5 – включить тактовую частоту процессора 2.5 МГц
- D6 – отключить переключение ОЗУ 0F000H..0FFFFH (в режиме "Orion-128" игнорируется)
- D7 – включить режим "Orion-128" (область 0F000H..0FFFFH недоступна для записи).

Использование разрядов D3..D7 будет описано ниже.

Для выбора сегментов в каждом из окон "RAM0" "RAM1", "RAM2" в компьютере предусмотрены три порта с адресами соответственно 04H, 05H, 06H, в которые могут быть записаны номера сегментов ОЗУ. Разрядность портов позволяет управлять переключением 256 сегментов, что составляет 4 Мбайта памяти. Это – предел для расширения памяти на "Orion-Pro".

Заметим, что порты диспетчера 04H, 05H, 07H, 08H, 0AH доступны как для записи, так и для чтения.

Сегмент памяти, включенный в окне, имеет более высокий приоритет, чем основная память в рабочей странице. Так, например, если в рабочей странице 2 открыть окно "RAM1" (установкой разряда D1 порта 0AH в единицу) и включить в нем сегмент 16 (записью в порт 05H значения 16), то адресное пространство процессора будет распределено следующим образом:



Таким образом, при доступе к памяти через окна ОЗУ над основной памятью страницы (тоже сегменты, но приписанные к данной странице) образуется как бы второй "слой".

Заметим, что область памяти 0F000H..0FFFFH только в режиме "Orion-Pro" (разряд D7 порта 0AH установлен в 0) доступна как для чтения, так и для записи. Кроме того, программно можно установить режим, при котором указанная область или переключается вместе с переключением страниц (D6=0), или не переключается (D6=1).

Если область памяти 0F000H..0FFFFH находится в непереключаемом режиме, то физически (аппаратно) там проецируется соответствующая часть 31 сегмента. Это позволяет весь сегмент 3 отдать под экран и иметь экран-0 также 16 Кбайт т.е.

512x256 точек. Учитывая вышесказанное, программы пользователя не должны использовать 31 (1FH) сегмент.

В режиме "Orion-128" (разряд D7 порта 0AH установлен в 1) указанная область памяти является непереключаемой независимо от значения разряда D6, к тому же ячейки с адресами 0F400H..0FA00H доступны как порты (причем порты 0F800H..0FA00H доступны только на запись, так как при чтении по адресам 0F000H..0FFFFH включается ОЗУ).

Заметим, что разряд D5 порта диспетчера 0AH управляет выбором тактовой частоты компьютера: для D5=0 она составляет 10 МГц (5 МГц при перестановке переключки J1/2-3), для D5=1 – 2.5 МГц. Режим с тактовой частотой 2.5 МГц используется в основном для старых программ (игрушек) от "Орион-128".

Доступ к расширенной памяти, находящейся на внешней интерфейсной плате, в пределах первых 512 Кбайт также может быть как постраничным (страницы 8..15), так и посегментным (сегменты 32..63). Остальная расширенная память (свыше 1 Мбайта) не может переключаться страницами и доступна как сегменты через RAM-окна (номера сегментов 64..256).

ОРГАНИЗАЦИЯ ЭКРАННОЙ ПАМЯТИ

Экранная память располагается в 0 и 1 страницах ОЗУ, причем количество экранов и распределение сегментов в них зависит от текущего цветового режима, задаваемого разрядами порта 0F8H:

D4	D3	D2	D1	D0	
0	x	0	0	0	Монохромный, палитра 1
0	x	0	0	1	Монохромный, палитра 2
0	x	0	1	z	Запрет видеосигнала
0	x	1	0	0	2-битный (4-цветный), палитра 1
0	x	1	0	1	2-битный (4-цветный), палитра 2
0	x	1	1	x	16-цветный с групповым кодированием
0	1	1	1	x	Псевдоцветной (цвет в порт 0FCH)
1	x	0	x	x	3-битный (8-цветный RGB)
1	x	1	x	x	4-битный (16-цветный RGBI)

В монохромном режиме палитре 1 соответствует комбинация цветов (черный, зеленый), палитре 2 – (белый, зеленый). В 4-цветном (2-битном) режиме палитре 1 соответствуют цвета – (черный, синий, зеленый, красный), палитре 2 – (белый, синий, зеленый, красный). Код палитры для псевдоцветного режима записывается в порт с адресом 0FCH.

Выбор на отображение одного из 4-х экранов выполняется путем записи номера экрана в порт 0FAH:

- D0 \ номер экрана
- D1 /
- D6 – выключение регенерации ОЗУ
- D7 – включение широкого экрана

Разряды D2–D5 являются резервными.

Если разряд D7 установлен в единицу, то ширина экрана составляет 512 точек (64 байта), что при высоте 256 байт соответствует объему памяти 16 Кбайт. В противном случае экранная плоскость ОЗУ имеет ширину 384 точки (48 байт) и занимает объем 12 Кбайт. В 3-х битном и 4-х битном (ЕGА-режим) цветовых режимах допускается использование только двух экранов, поэтому разряд D0 порта OFAH игнорируется.

Рассмотрим распределение сегментов экранного ОЗУ в различных цветовых режимах.

Монохромный и цветной режимы

В монохромном и псевдоцветном режимах возможно использование до 4-х экранов, занимающих только сегменты 0-й страницы ОЗУ:

	Стр.0	Экран 12 К	Экран 16 К
Экран 0 →	3	C000H..EFFFH	C000H..FFFFH
Экран 1 →	2	8000H..AFFFH	8000H..BFFFH
Экран 2 →	1	4000H..6FFFH	4000H..7FFFH
Экран 3 →	0	0000H..2FFFH	0000H..3FFFH

В монохромном режиме единичному значению некоторого бита экранного сегмента ОЗУ соответствует засветка изображаемой точки, нулевому – гашение.

В псевдоцветном режиме цвет отображаемых точек зависит от кода палитры, записанного в порт OFCH. Старшие 4 бита значения этого порта определяют один из 16 цветов фона (для погашенных точек), младшие 4 бита – один из 16 цветов переднего плана (для засвеченных точек).

Заметим, что при широком экране-0 область OF000H..0FFFFH экрана (не путать с системной областью OF000..0FFFFH) доступна только через окно. Прямой доступ к экрану возможен только по адресам 0C000-0EFFFH. Это относится ко всем цветовым режимам.

4-цветный режим

В 4-цветном (2-битном) режиме цвет каждой отображаемой точки зависит от соответствующих битов двух экранных плоскостей (сегментов), находящихся в страницах 0 и 1 ОЗУ:

	Стр.0	Стр.1	
Экран 0 →	3	7	
Экран 1 →	2	6	
Экран 2 →	1	5	
Экран 3 →	0	4	

0 0	→	черный (белый)
0 1	→	красный
1 0	→	зеленый
1 1	→	синий

8-цветный и 16-цветный режимы

Это новый графический режим. Функционально он тождествен EGA режиму на IBM PC AT (был широко распространен на 286 моделях). В 8-цветном (3-битном) и 16-цветном (4-битном) режимах для формирования отображаемой точки в каждом из двух экранов используются соответственно 3 и 4 плоскости экранного ОЗУ:

Сегментам 3 и 1 соответствует зеленый цвет (G), 2 и 0 – красный (R), 6 и 4 – синий (B), 7 и 5 (в 3-битном режиме не используются) – управление яркостью (I). Путем записи комбинации битов в соответствующие сегменты экрана можно получить точку заданного цвета.

	Стр. 0	Стр. 1
Экран 0 →	3 (G) 2 (R)	7 (I) 6 (B)
Экран 1 →	1 (G) 0 (R)	5 (I) 4 (B)

Режим с групповым кодированием цвета

В 16-цветном режиме с групповым кодированием каждый из 4-х экранов формируется из содержимого двух сегментов памяти: из плоскости изображения (0 страница ОЗУ) и плоскости цветовых атрибутов (1 страница ОЗУ), причем восьми соседним точкам плоскости изображения, расположенным в пределах одного байта, соответствует один байт из плоскости цветовых атрибутов.

Старшие 4 бита в байте цветового атрибута определяют цвет фона (для погашенных точек), младшие 4 бита – цвет переднего плана (для засвеченных точек) в пределах одного экранного байта.

	Стр. 0 (изобр)	Стр. 1 (цвет)
Экран 0 →	3	7
Экран 1 →	2	6
Экран 2 →	1	5
Экран 3 →	0	4

Для всех цветовых режимов действует ограничение на использование широкого экрана с номером 0.

Следует помнить, что экраны аппаратно привязаны к конкретным сегментам ОЗУ, а не к окнам, т.е. отображение информации экрана не зависит от рабочей страницы ОЗУ и включения / выключения окон.

ОРГАНИЗАЦИЯ ПОСТОЯННОЙ ПАМЯТИ

Постоянная память, расположенная на основной плате, состоит из двух частей:

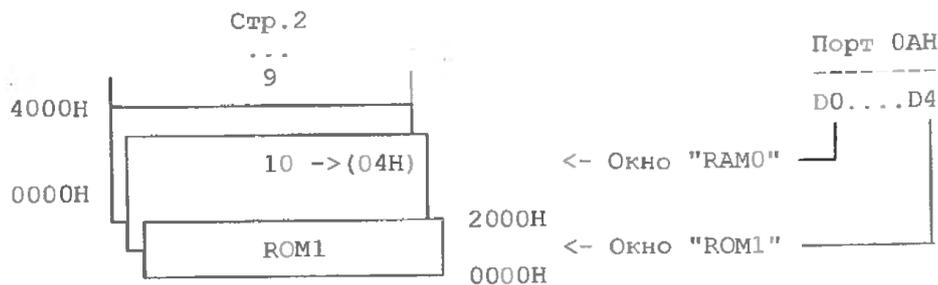
"ROM1-BIOS" – стартовое ПЗУ объемом 8 Кбайт;

"ROM2-BIOS" – ПЗУ пользователя объемом 8-64 Кбайт.

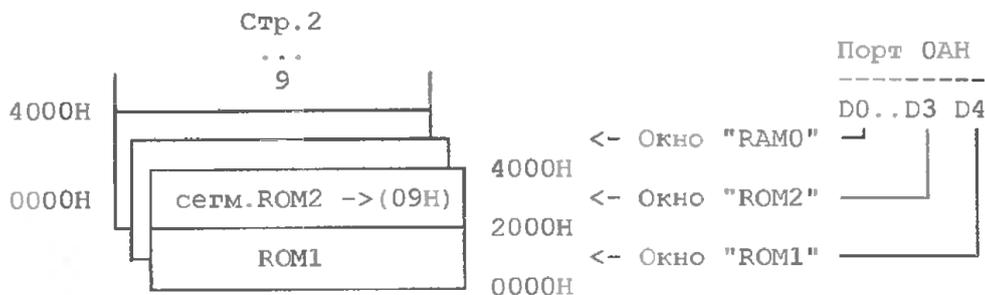
Для доступа к постоянной памяти в адресном пространстве процессора предусмотрено включение соответственно двух ROM-окон ПЗУ. Окно для "ROM1-BIOS" включается по адресам 0000H..1FFFH при аппаратном сбросе компьютера, тем самым обеспечивая доступ к стартовым и другим подпрограммам. Для управления включением и выключением окна "ROM1-BIOS" предназначен бит D4 порта диспетчера 0AH.

Включением окна "ROM2-BIOS" по адресам 2000H..3FFFH управляет бит D3 порта 0AH (независимо от окна "ROM1-BIOS"), причем доступ к ПЗУ в этом окне осуществляется по сегментам размером 8Кбайт (отсюда и минимальный размер ПЗУ). Номер сегмента "ROM2-BIOS" записывается в специальный порт с адресом 09H, допускающий как запись, так и чтение информации.

Окна ПЗУ имеют самый высокий приоритет: если окно "ROM1-BIOS" и/или "ROM2-BIOS" включено, то доступ к нему обеспечивается из любой текущей страницы, в том числе при "открытом" окне ОЗУ "RAM0". Например, если во 2-й странице ОЗУ в любом порядке открыть окно "RAM0" с сегментом 10 (путем записи значения 10 в порт 04H и установкой разряда D0 порта 0AH в единицу) и окно "ROM1" (установкой разряда D4 порта 0AH в единицу), то распределение памяти будет следующим:



По адресам 0000H..1FFFH будет доступно ПЗУ "ROM1", а по адресам 2000H..3FFFH – "верхняя" часть 10-го сегмента ОЗУ. Если в приведенном выше примере открыть еще и окно ПЗУ "ROM2" (установкой бита D3 порта 0AH в единицу и записью в порт 09H номера требуемого сегмента "ROM2"), то 10-й сегмент ОЗУ в окне "ROM0" будет полностью затенен и недоступен:



Можно сказать, что при включении окон ПЗУ образуется как бы третий "слой" памяти в пределах адресного пространства процессора.

ВНИМАНИЕ: Программист должен сам следить за тем, чтобы коды программы, область стека и используемые данные не оказались в переключаемой части страницы или в одном из переключаемых окон ОЗУ (ПЗУ) для исключения их "потери".

ПОРТЫ ВВОДА-ВЫВОДА

В режиме "Orion-128" (бит D7 порта 0AH установлен в 1) разрешен доступ к портам 0F400H..0FA00H, адресуемым через область памяти, и к портам 10H..14H, 18H..1BH, 0F8H..0FFH, а также портам периферии – с помощью команд процессора IN, OUT.

В режиме "Orion-Pro" (бит D7 порта 0AH установлен в 0) доступ к портам как к ячейкам ОЗУ запрещен.

Порты ПК "Орион-ПРО:

00H – состояние DIP-переключателей (SW), определяющих конфигурацию системы;

D7 – 0 – Режим «Орион-ПРО»
 1 – Режим «Орион-128»

D6 – загрузка
 128 0 – внешний ROM-диск
 1 – внутренний ROM-диск

ПРО 0 — CP/M из ROM-BIOD
1 — CP/M с дискеты

D5 = 0 Режим Меню

D4 = 0 Наичие символьного дисплея

D3 = 0 Страница CP/M из ROM-BIOS (0→2, 1→1)

D2 — клавиатура, 1 → RK86, 0 → MC7007

D1 = 0 наличие Hard-диска

D0 = 0 наличие дисковода

01H — данные принтера "Centronics" (инвертируются);

02H — сигналы управления принтером (инвертируются);

D7 ← BUSY

D6 ← ERROR

D5 ← PAGEEND

D4

D3

D2

D1

D0 → STROBE

03H — регистр настройки портов 00H..02H;

04H — регистр сегментов для окна "RAM0";

05H — регистр сегментов для окна "RAM1";

06H — регистр сегментов для окна "RAM2";

07H — регистр настройки портов 04H..06H;

08H — регистр страниц ОЗУ;

D3 —

D2 —

D1 —

D0 —

09H — регистр сегментов "ROM2-BIOS";

0AH — диспетчер управления конфигурацией памяти;

D7 \

D6 / 00 \ порты «128» в ОЗУ

01 / F000-FFFF — R/O

10 порты IN/OUT F000-FFFF — RD/WR, переключ.

11 порты IN/OUT F000-FFFF — RD/WR, непереключ.

D5 = 1 — Фтакт = 2,5 МГц

D4 = 1 — вкл. окно ROM-BIOS2 (старт.)

D3 = 1 — включить окно ROM-BIOS1

D2 = 1 — включить окно 2

D1 = 1 — включить окно 1

D0 = 1 — включить окно 0

0BH — регистр настройки портов 0BH..0AH;

0CH..0EH — системный резерв;

10H — регистр команд (0F710H, 0F700H);

11H – регистр дорожек (0F711H, 0F701H);

12H – регистр секторов (0F712H, 0F702H);

13H – регистр данных (0F713H, 0F703H);

14H – регистр управления (0F704H, 0F714H, 0F720H);

18H..1BH – универсальный порт, используемый как порт клавиатуры, в режиме "Orion-128" может быть переключен (параллельно с обращением 18-1BH) на адреса одного из портов 0F4XH, 0F5XH, 0F6XH по выбору пользователя;

Порты мультикарты

1CH – вспомогательный порт (X3)

1EH – порт мыши с параллельным выходом (X4)

D7– резерв

D6– резерв

D5– правая кнопка

D4– левая кнопка

D3– вверх

D2– вниз

D1– вправо

D0– влево

1FH – порт джойстика (X6)

D7– резерв

D6– резерв

D5– резерв

D4– огонь

D3– вверх

D2– вниз

D1– вправо

D0– влево

20-23H – универсальный порт (разъем X2)

24-27H – резерв

28-2BH – порт ROM-диска (BC – адрес, A – данные)

2C-2FH – переключение страниц ROM-диска (D0-D3)

30-9FH – резерв для плат расширения (будет объявляться с выходом конкретной интерфейсной платы).

Порты электронного диска

0A0H – номер страницы (0-15)

0A1H – старший байт адреса страницы

0A2H – младший байт адреса страницы

0A3H – порт данных диска 1 Мбайт (диск D:)

0A4H – порт данных диска 128 Кбайт (диск C:)

0B0-0DFH – область портов, выделенная пользователю.

Порты управления палитрами

0E0H – R 07-04 – адрес ячейки

0E1H – G 03-00 – содержимое ячейки

0E2H – B

0E8H – регистр управления цветовым режимом дисплея; в режиме "Orion-128" доступен также, как ячейка 0F800H (см.выше);

0F9H – регистр страниц; для режима "Orion-128" доступен также, как ячейка 0F900H;

0FAH – регистр номера и размера экрана; в режиме "Orion-128" доступен также, как ячейка 0FA00H;

D7 = 1 – экран 16K

D6 = 1 – выкл.регенер. ОЗУ (1,3,5,7 – стр)

D1 \ – номер экрана

D0 /

0FBH – регистр включения прерываний от таймера (06); 06 = 1 – разрешение прерывания по таймеру.

0FCH – регистр цвета для псевдоцветного режима; D7-D4 – код фона. D3-D0 – код цвета В порт F8 записывается код включения режима – 0EH или 0FH.

0FDH – резерв для двухбайтовых адресов портов расширений "ZX";

0FEH – порт четырехразрядного ЦАП; 03-00 – код (0-E) значения ЦАП.

0FFH – порт звука (по факту обращения к порту)

Порты 0F8H..0FFH работают только на запись.

BIOS компьютера "ORION-PRO"

Под Базовой Системой Ввода – Вывода (BIOS) компьютера "Orion-Pro" будем понимать набор подпрограмм, обеспечивающих взаимодействие программ пользователя с аппаратными ресурсами компьютера. Излагаемый далее материал касается BIOS версии 2.00. Доступ к стандартным подпрограммам BIOS осуществляется через таблицу векторов, расположенную в не переключаемой области оперативной памяти по адресам 0F800H..0F865H. Каждый вектор занимает в таблице 3 байта: первый байт содержит команду процессора "JP" (код 0C3H), второй и третий – соответственно младший и старший байты адреса перехода. Местоположение каждого вектора в таблице и интерфейс соответствующей подпрограммы строго определены и не изменяются при доработках BIOS.

Так как таблица векторов расположена в ОЗУ, адреса перехода на их обработку могут быть "перехвачены" программистом для подключения внешних подпрограмм.

BIOS можно условно разделить на три составные части:

"ROM1-BIOS" – размещается в ПЗУ "ROM1" объемом 8 Кбайт и содержит стартовый блок, набор драйверов (клавиатуры, мыши и пр.) и прочие специальные программы;

"ROM2-BIOS" – размещается в ПЗУ "ROM2" объемом не менее 32 Кбайт и содержит драйвер дисплея "TV-PRO" (первые 8 Кбайт используются для

хранения внутренней системы ORDOS для режима "Orion-128", а собственно "TV-PRO" занимает следующие 24 Кбайта);

"RAM-BIOS" – размещается в не переключаемой области ОЗУ и содержит таблицу векторов, резидентные подпрограммы обращения к ОЗУ, ПЗУ, портам, а также рабочие ячейки. Эта часть формируется из "ROM1" и "ROM2" при "холодном" старте компьютера.

В качестве рабочей области BIOS используется сегмент ОЗУ 1FH, что должно учитываться при разработке программ для исключения конфликтных ситуаций. По относительным адресам 3000H..3FFFH, относительно начала этого сегмента находится не переключаемая область ОЗУ (0F000H..0FFFFH), распределение которой приведено в Таблице 1, а по адресам 1000H..2FFFH – дополнительная рабочая область драйвера "TV-PRO". Адреса 0000H..0FFFFH в сегменте 1FH зарезервированы для дальнейших расширений.

Заметим, что прямое обращение к рабочим ячейкам BIOS, минуя стандартные подпрограммы, запрещено, поэтому конкретные адреса этих ячеек и их назначение не объявляются.

РАСПРЕДЕЛЕНИЕ НЕПЕРЕКЛЮЧАЕМОЙ ПАМЯТИ

0FFFFH	Резидентная часть BIOS
0F866H	Таблица векторов BIOS
0F800H	Служебный стек BIOS
0F7C0H	Рабочий стек "Orion-Pro"
0F710H	Таблица адресов подпрограмм ISR
0F700H	Рабочие ячейки "TV-PRO"
0F5C0H	Оверлейная область "TV-PRO"
0F3C0H	Свободная область ОЗУ
0F000H	

ТАБЛИЦА ВЕКТОРОВ BIOS

Доступ к стандартным подпрограммам BIOS в режиме "Orion-Pro" возможен через таблицу векторов:

0F800H	(RESET)	перезагрузка системы в режиме "Pro"
0F803H	(KBRD)	ввод символа с клавиатуры
0F806H	(KBDCTL)	управление клавиатурой
0F809H	(TVC)	вывод символа на экран
0F80CH	(SCRCTL)	управление экраном
0F80FH	(SYMCTL)	управление выводом символа и курсора
0F812H	(KBSTAT)	опрос состояния клавиатуры
0F815H	(HEX)	вывод байта в шестнадцатиричной форме
0F818H	(MSG)	вывод сообщения на экран
0F81BH	(INKEY)	опрос кода нажатой клавиши
0F81EH	(RCUR)	получение позиции курсора
0F821H	(PRINT)	вывод на печать
0F824H	(NUMKEY)	опрос номеров нажатых клавиш
0F827H	(INFAST)	быстрый опрос клавиш курсора, спецклавиш
0F82AH	(GRAPH)	вывод графики
0F82DH	(MEMORY)	система динамического распределения ОЗУ
0F830H	(FILL)	заполнение страницы ОЗУ константой
0F833H	(MOVBL)	перенос блока кодов
0F836H	(RRAM)	чтение байта из страницы ОЗУ
0F839H	(WRAM)	запись байта в страницу ОЗУ
0F83CH	(WCUR)	установка позиции курсора
0F83FH	(SOUND)	синтез звука
0F842H	(RDWIN)	чтение портов диспетчера в ОЗУ
0F845H	(WRWIN)	восстановление портов диспетчера из ОЗУ
0F848H	(JPWIN)	установка варианта диспетчера и переход
0F84BH	(OPCLW)	открыть/закрыть окно ОЗУ, ПЗУ
0F84EH	(MOUSE)	опрос порта мыши
0F851H	(MSCTL)	управление мышью
0F854H	(RDSEC)	чтение сектора с диска
0F857H	(WRSEC)	запись сектора на диск
0F85AH	(SYSCTL)	управление системой
0F85DH	(VERS)	получение номера версии ROM1/ROM2
0F860H	(KBSND)	формирование звука при нажатии клавиши

0F863H (KBSIG) сигнализация переключения флагов клавиатуры

Рассмотрим назначение и особенности использования векторов стандартных подпрограмм.

Вектор RESET (0F800H) не имеет параметров и используется для перезапуска компьютера в режиме "Опoп-Рго". Это единственный вектор, не являющийся подпрограммой, т.е. для обращения к нему необходимо использовать команду процессора JP 0F800H.

Вектора KBRD (0F803H), KBDCTL (0F806H), KBSTAT (0F812H), INKEY (0F81BH), NUMKEY (0F824H), INFASТ (0F827H), KBSND (0F8600H) и KBSIG (0F863H) предназначены для обслуживания клавиатуры и описаны отдельно.

Подпрограмма TVC (0F809H) выводит на экран символ, код которого задан в регистре С. Ниже приведен перечень управляющих кодов и ESC-последовательностей, поддерживаемых подпрограммой TVC

УПРАВЛЯЮЩИЕ КОДЫ

07H – BELL (звуковой сигнал);

08H – LEFT (курсор влево);

09H – TAB (табуляция на 8);

0AH – LF (перевод строки, ПС);

0CH – CLS+HOME (очистка окна и курсор в левый верхний угол);

0DH – CR (возврат каретки, ВК);

10H – CTRL (префикс перед символами с кодами 00–1FH);

18H – RIGHT (курсор вправо);

19H – UP (курсор вверх);

1AH – DOWN (курсор вниз);

1BH – ESCC (начало ESC последовательности, AP2);

1FH – CLS+HOME (очистка окна и курсор в левый верхний угол);

ESC-ПОСЛЕДОВАТЕЛЬНОСТИ

ESC '2' – вкл. автоперевод при выводе в посл.позицию строки;

ESC '3' – выкл. автоперевод при выводе в посл.позицию строки;

ESC '4' – вкл. скроллинр при выводе ЪЕ в последней строке;

ESC '5' – выкл.скроллинг при выводе ТЕ в последней строке;

ESC '6' – включение инверсии символов;

ESC '7' – выключение инверсии символов;

ESC '8' – включение подчеркивания символов (курсором);

ESC '9' – выключение подчеркивания символов;

ESC ':' – включение непрерывного отображения курсора в TVC;

ESC ';' – выключение отображения курсора;

ESC '<' – включение мигания курсора при вводе в KBRD;

ESC '>' – выключение мигания курсора при вводе в RBRD;
ESC 'A' – UP (курсор вверх);
ESC 'B' – DOWN (курсор вниз);
ESC 'C' – RIGHT (курсор вправо);
ESC 'D' – LEFT (курсор влево);
ESC 'E' – CLS (очистка окна);
ESC 'F',C1,C2 – уст.цветов фона C1 и изображения C2;
ESC 'H' – HOME (курсор в левый верхний угол окна);
ESC 'J' – стирание текста от курсора до конца окна;
ESC 'K' – стирание текста от курсора до конца строки;
ESC 'L' – вставка строки;
ESC 'M' – удаление строки;
ESC 'R',Byte – установка толщины курсора;
ESC 'S',Num – уст.номера доступного и видимого экранов;
ESC 'T',Pal – установка палитры видимого экрана;
ESC 'U',Mode – уст.режима доступного и видимого экранов;
ESC 'V',Flag – вкл./выкл. изображения видимого экрана;
ESC 'Y',Lin+20H,Col+20H – уст.курсора в позицию Col,Lin.

В состав драйвера дисплея "TV-PRO" входят также следующие управляющие вектора:

SCRCTL (0F80CH) – управление экраном: режимы, цвета, окна;

SYMCTL (0F80FH) – управление выводом символа и курсора;

GRAPH (0F82AH) – вывод графики;

MEMORY (0F82DH) – система динамического распределения сегментной оперативной памяти.

Каждый из указанных векторов выполняет целый набор функционально объединенных подпрограмм (функций), причем для обращения к требуемой функции в регистр A необходимо поместить ее номер. Функции "TV-PRO" будут описаны далее.

Подпрограмма **HEX** (0F815H) используется для вывода байта в шестнадцатиричной форме на экран, указанного в регистре A.

Подпрограмма **MSG** (0F818H) служит для вывода на экран сообщения, адрес начала которого задан в регистровой паре HL (признаком конца сообщения является байт 00H). Заметим, что для вывода символов в подпрограммах HEX и MSG используется подпрограмма с адресом TVC.

Подпрограмма **RCUR** (0F81EH) возвращает символьную позицию курсора: в регистре H – по Y, в регистре L – по X, причем левому верхнему углу экрана (Home) соответствуют координаты (0,0).

Подпрограмма **WCUR** (0F83CH) выполняет обратную функцию: устанавливает курсор в позицию экрана, заданную значениями регистров H и L.

Подпрограмма **PRINT** (0F821H) предназначена для обслуживания печатающего устройства и выполняет две функции. Если входное значение регистра A=0, то подпрограмма PRINT выводит символ с кодом, указанным в регистре C, на печать. В регистре A на выходе возвращается байт состояния печатающего устройства со следующими значениями битов:

D0 – отсутствие бумаги в принтере,
D1 – признак ошибки ERROR,
D2 – признак отсутствия готовности BUSY,
D3..D7 – нули.

Таким образом, признаком готовности принтера является выходное значение A=0. При входном значении регистра A=1 подпрограмма PRINT выдает байт состояния принтера (как и при печати символа).

Подпрограмма **FILL**. (0F830H) заполняет заданную область указанной страницы ОЗУ константой. В регистре A указывается номер страницы ОЗУ, в регистре C записываемый байт, в регистрах HL – начальный адрес ОЗУ, DE – длина заполняемой области.

Подпрограмма **MOVBL** (0F833H) копирует блок данных из одной страницы памяти в другую. В регистре A указываются номера обеих страниц (старшие 4 разряда – "откуда", младшие – "куда"), в регистрах HL – начальный адрес "откуда", DE начальный адрес "куда", BC – число копируемых байт.

Подпрограмма **RRAM** (0F836H) считывает в регистр C содержимое ячейки по адресу HL из страницы ОЗУ с номером, указанным в регистре A.

Подпрограмма **WRAM** (0F839H) записывает значение регистра C по адресу HL в страницу ОЗУ с номером, указанным в регистре A.

Подпрограмма **SOUND** (0F83FH) предназначена для формирования звукового сигнала. В регистровой паре BC задается длительность в миллисекундах, в DE частота в герцах. Так как генерация звука осуществляется программным способом, данная подпрограмма выдает сигнал с усредненной частотой. Если значение DE=0, то формируется пауза заданной длительности.

Вектора **RDWIN** (0F842H), **WRWIN** (0F845H), **JPWIN** (0F848H) предназначены для оперативного управления памятью компьютера. Подпрограмма RDWIN считывает (сохраняет) в ОЗУ состояние портов диспетчера, для чего используется специальная таблица (буфер) с адресом HL длиной 6 байт. Значения портов сохраняются в указанном буфере в следующем порядке:

порт 04H (номер сегмента ОЗУ в окне 0)
порт 05H (номер сегмента ОЗУ в окне 1)
порт 06H (номер сегмента ОЗУ в окне 2)
порт 08H (номер страницы ОЗУ)
порт 09H (номер сегмента ROM2)
порт 0AH (состояние диспетчера).

Подпрограмма **WRWIN** выполняет "зеркальную" функцию – восстанавливает состояние перечисленных портов из буфера ОЗУ, адрес которого задан в регистровой паре HL.

Ограничение: буфер для сохранения/восстановления портов должен находиться в непереключаемой области памяти, в противном случае он может быть "потерян" при переключении сегментов (страниц) ОЗУ.

Подпрограмма **JPWIN** предназначена для установки значений портов из таблиц (аналогично WRWIN) с последующей передачей управления. Данная подпрограмма позволяет работать с несколькими предварительно настроенными таблицами, каждая из которых состоит из 8 байт: первые 6 байт содержат значения портов, последние 2 байта – адрес перехода. Адрес начала всего блока таблиц задается в регистровой паре HL при первоначальном вызове JPWIN со значением аккумулятора A=0FFH. Для выполнения функции настройки диспетчера памяти с передачей управления в регистре A должен быть указан номер таблицы (номер канала), по которой JPWIN формирует состояние портов и выполняет команду перехода (JP).

Подпрограмма **OPCLW** (0F84BH) используется для открытия (при A=1) и закрытия (при A=0) окон в памяти. В регистре C задается номер окна:

- 0 – окно 0 ОЗУ,
- 1 – окно 1 ОЗУ,
- 2 – окно 2 ОЗУ,
- 3 – окно ROM2,
- 4 – окно ROM1.

В регистре B для функции открытия окна указывается номер сегмента ОЗУ (ПЗУ).

Подпрограммы **RDSEC** (0F854H) и **WRSEC** (0F857H) используются для чтения/записи сектора гибкого диска, при этом на входе в регистровой паре HL указывается адрес начала буфера сектора, а результат операции возвращается в регистре A (A=0 – успешно). Регистровая пара DE на выходе содержит адрес конца буфера + 1, а HL не изменяется.

Подпрограммы **MOUSE** (0F84EH) и **MSCTL** (0F851H) предназначены для обслуживания мыши. Они будут описаны в соответствующем разделе.

Подпрограмма **SYSCTL** (0F85AH) выполняет функции управления системой. Номер функции задается в регистре A:

- A=0 – инициализация рабочих ячеек "ROM1-BIOS";
- A=1 – инициализация портов диспетчера памяти (отключение всех окон);
- A=2 – освобождение оверлейной области ОЗУ, используемой драйвером "TV-PRO";
- A=3 – проверка занятости оверлейной области ОЗУ, используемой драйвером "TV-PRO" ("Z" – область свободна для размещения внешних данных, "NZ" – область занята данными "TV-PRO").

Подпрограмма **VERS** (0F85DH) используется для определения номеров версий "ROM1-BIOS" и "ROM2-BIOS":

- H – двоично-десятичный № версии "ROM1-BIOS",
- E – двоично-десятичный № версии "ROM2-BIOS" ("TV-PRO").

Для ROM-BIOS V2.00 в регистрах HL возвращается значение 2020H.

ОСОБЕННОСТИ ДРАЙВЕРА КЛАВИАТУРЫ

Драйвер клавиатуры рассчитан на обслуживание клавиатур типа РК-86 и MC7007.

Подпрограмма **KBRD** (0F803H) используется для ввода символа с клавиатуры (с ожиданием нажатия клавиши и выводом курсора). Для клавиатуры MC7007 поддерживается старое и новое распределение спецклавиш. Обеспечивается работа в 4-х кодировках: альтернативной, КОИ-8, КОИ-7 / Н2 (большие латинские и большие русские буквы), КОИ-7 / Н1, Н0 (большие и маленькие латинские, большие и маленькие русские). В данной подпрограмме предусмотрена возможность работы с внутренним буфером клавиатуры, рассчитанным на хранение 15 кодов клавиш. Буфер

организован по принципу очереди (FIFO – "первым пришел – первым ушел"). Подпрограмма KBRD при вызове проверяет наличие данных в буфере и выталкивает первый находящийся в нем байт на выход. Если же буфер пуст осуществляется переход к вводу с клавиатуры очередного символа. Наличие буфера KBRD дает возможность использования функциональных клавиш. Кроме того, пользователь в своих программах может заполнять буфер клавиатуры требуемыми кодами для автоматического выполнения некоторых команд.

Заметим, что драйвер клавиатуры использует еще один внутренний буфер NUMKEY с кодами сканирования нажатых клавиш (смотри описание подпрограммы NUMKEY).

Принцип обработки функциональных клавиш состоит в следующем. При нажатии такой клавиши (отдельно или совместно с одной из клавиш УПР, SHIFT) соответствующая ей строка кодов заносится в буфер, откуда будет посимвольно выдаваться при последующих вызовах KBRD. Функциональной может быть назначена любая клавиша, кроме спецклавиш УПР (УС), SHIFT (СС), ФИКС (РУС/ЛАТ), ГРАФ, АЛФ. Коды сканирования функциональных клавиш задаются в специальных таблицах вместе с соответствующими генерируемыми строками. Таких таблиц может быть три:

- для отдельно нажатых клавиш,
- для клавиш, нажатых совместно с УПР (УС),
- для клавиш, нажатых совместно с SHIFT (СС),

причем клавиши во всех таблицах могут быть выбраны разные.

Наличие данных в любой из этих таблиц не является обязательным: все таблицы вообще могут быть пустыми – в этом случае KBRD работает без функциональных клавиш.

Каждая из указанных таблиц имеет следующую структуру:

n	Список кодов сканирования клавиш
L1	Строка для 1-й клавиши длиной L1 байт
L2	Строка для 2-й клавиши длиной L2 байт
...	...
Ln	Строка для n-й клавиши длиной Ln байт

Первый байт таблицы задает число n клавиш, которые будут функциональными, т.е. определяет длину списка клавиш. Список кодов сканирования клавиш включает n байт, при этом значению n=0 соответствует пустая таблица (она будет состоять всего из 1 байта).

Далее в таблице (если она не пустая) располагаются n следующих друг за другом строк, предваряемых байтом длины L1..Ln.

При нажатии функциональной клавиши соответствующая строка будет заноситься в буфер клавиатуры. Адреса таблиц функциональных клавиш могут быть установлены/получены с помощью вектора KBDCTL.

Подпрограмма KBDCTL. (0F806H) используется для управления клавиатурой. В зависимости от номера функции, задаваемой в регистре А, она позволяет устанавливать и получать значения следующих параметров:

- код режима (KBMODE) и байт флагов (KBFLAG) клавиатуры, которые действуют для векторов KBRD и INKEY;
- константы задержки автоповтора и мигания курсора для вектора KBRD;
- коды клавиш управления курсором (в том числе при нажатии клавиш УПР и SHIFT) для векторов KBRD и INKEY;
- функциональные клавиши и соответствующие им последовательности кодов для вектора KBRD;
- коды клавиш включения/выключения псевдографического режима для клавиатуры РК-86.

Режим клавиатуры (KBMODE) описывается состоянием следующих битов:

- D7 – признак старого распределения специальных клавиш для клавиатуры МС7007 (при единичном значении этого бита управление МС7007 совпадает с РК-86);
- D1, D0 – определяют тип кодировки клавиатуры:
 - 00 – альтернативная;
 - 01 – КОИ-8;
 - 10 – КОИ-7/Н2;
 - 11 – КОИ-7/Н1, 0.

В байте флагов клавиатуры (KBFLAG) имеется 5 временных флагов, устанавливаемых каждый раз после вызова подпрограмм KBRD, INKEY и NUMKEY. и 3 триггерных флагов, устанавливаемых нажатием комбинаций спецклавиш в подпрограмме KBRD:

а) временные флаги – сигнализируют о нажатии соответствующих клавиш:

- D0 – нажата клавиша УПР (УС);
- D1 – нажата клавиша SHIFT (СС);
- D2 – нажата клавиша ФИКС (РУС/ЛАТ);
- D3 – нажата клавиша АЛФ (для МС7007);
- D4 – нажата клавиша ГРАФ (для МС7007);

б) триггерные флаги – определяют текущий режим ввода:

- D5 – режим CAPS LOCK (1 – большие буквы);
- D6 – режим РУС/ЛАТ (1 – русские);
- D7 – режим ГРАФ/АЛФ (1 – псевдографика).

Бит CAPS LOCK действует во всех кодировках, кроме КОИ-7/Н2 (там нет больших и малых букв), бит РУС/ЛАТ – во всех кодировках, кроме КОИ-7 /Н1, Н0, бит ГРАФ/АЛФ – только в альтернативной кодировке и КОИ-8.

Путем принудительной установки триггерных флагов пользователь может переводить клавиатуру в требуемое состояние.

При вызове подпрограммы **KBDCTL** в регистре А указывается номер функции:

A=0 – Установка режима клавиатуры

На входе:

- E – код режима (KBMODE);
- D – байт флагов (KBFLAG);

A=1 – Получение режима клавиатуры

На выходе:

- E – код режима (KBMODE);
- D – байт флагов (KBFLAG);
- HL – адрес буфера NUMKEY с номерами клавиш после последнего вызова

KBRD, INKEY, NUMKEY;

A – число нажатых клавиш в буфере NUMKEY;

C – номер последней клавиши в буфере NUMKEY (если буфер пуст, то C=0FFH, и на выходе установлен флаг "Z");

A=2 – Установка констант задержки

На входе:

HL – константа автоповтора;

DE – константа мигания курсора

A=3 – Получение констант задержки

На выходе:

HL – константа автоповтора;

DE – константа мигания курсора;

A=4 – Установка кодов клавиш курсора

На входе:

C – влево

B – вправо

E – вверх

D – вниз

A=5 – Получение кодов клавиш курсора

На выходе:

C – влево

B – вправо

E – вверх

D – вниз

A=6 – Установка кодов клавиш курсора при нажатой УПР

На входе:

C – влево + УПР

B – вправо + УПР

E – вверх + УПР

D – вниз + УПР

A=7 – Получение кодов клавиш курсора при нажатой УПР

На выходе:

C – влево + УПР

B – вправо + УПР

E – вверх + УПР

D – вниз + УПР

A=8 – Установка кодов клавиш курсора при нажатой SHIFT

На входе:

C – влево + SHIFT

B – вправо + SHIFT

E – вверх + SHIFT

D – вниз + SHIFT

A=9 – Получение кодов клавиш курсора при нажатой SHIFT

На выходе:

C – влево + SHIFT

B – вправо + SHIFT

E – вверх + SHIFT

D – вниз + SHIFT

A=10 – Установка адреса таблицы функциональных клавиш

На входе:

С – номер сегмента ОЗУ с таблицей,

HL – адрес таблицы в сегменте

A=11 – Получение адреса таблицы функциональных клавиш

На выходе:

С – номер сегмента ОЗУ с таблицей

HL – адрес таблицы в сегменте

A=12 – Установка адреса таблицы функц. клавиш при нажатой УПР

На входе:

С – номер сегмента ОЗУ с таблицей, HL – адрес таблицы в сегменте

A=13 – Получение адреса таблицы функц. клавиш при нажатой УПР

На выходе:

С – номер сегмента ОЗУ с таблицей, HL – адрес таблицы в сегменте

A=14 – Установка адреса таблицы функц. клавиш при нажатой SHIFT

На входе:

С – номер сегмента ОЗУ с таблицей, HL – адрес таблицы в сегменте

A=15 – Получение адреса таблицы функц. клавиш при нажатой SHIFT

На выходе:

С – номер сегмента ОЗУ с таблицей, HL – адрес таблицы в сегменте

Следует заметить, что устанавливаемые коды клавиш курсора действуют как для подпрограммы KBRD, так и для INKEY, а коды функциональных клавиш только для KBRD, причем приоритет последних выше, т.е. если клавиша управления курсором выбрана в качестве функциональной, то при ее нажатии на выходе KBRD будет сформирована строка кодов.

A=16 – Запись символа в буфер KBRD

На входе:

С – код символа; На выходе: "CY" – признак переполнения буфера

A=17 – Чтение символа из буфера KBRD

На выходе:

"CY" – буфер пуст, иначе: С – код символа.

A=18 – Обнуление буфера KBRD

A=19 – Проверка буфера клавиатуры

На выходе:

"CY" – буфер пуст, иначе содержит символ(ы).

A=20 – Установка кодов клавиш включения алфавитного и псевдографического режимов на РК86:

На входе:

L – для включения алфавитного режима;

H – для включения псевдографического режима;

A=21 – Получение кодов клавиш включения алфавитного и псевдографического режимов на РК-86:

На выходе:

L – для включения алфавитного режима (^O);

H – для включения псевдографического режима (^N);

Наличие функций 20, 21 позволяет назначать другие клавиши для включения / выключения псевдографики на клавиатуре РК-86 (и MS7007 в "старом режиме") и

освободить комбинации $\wedge O$, $\wedge N$ для других цепей. Например, если установить $H=L=07H$, то включение и выключение псевдографики будет осуществляться в триггерном режиме нажатием $\wedge G$.

Подпрограмма **KBSTAT** (0F812H) возвращает $A=0$ (и флаг "Z"), если ни одна клавиша не нажата, и $A=0FFH$ (флаг "NZ") – если нажата хотя бы одна клавиша, при этом клавиши УПР (УС), SHIFT (СС), ФИКС (РУС/ЛАТ) не опрашиваются (для клавиатуры МС7007 не опрашиваются также клавиши АЛФ и ГРАФ).

Подпрограмма **INKEY** (0F81BH) предназначена для опроса кода нажатой клавиши. Код нажатой клавиши возвращается в регистре А, при этом дополнительно устанавливается флаг "СУ" и $A=0FFH$, если ни одна клавиша не нажата, нажата одна из спецклавиш (SHIFT, УПР) или недопустимая комбинация спецклавиш. В остальных случаях на выходе INKEY формируется признак "NC", а в аккумуляторе – код клавиши с учетом нажатия клавиш УПР (УС), SHIFT (СС), АЛФ и ГРАФ (для МС7007). Кроме того, при определении кода клавиши учитывается текущее состояние байта тритерных флагов клавиатуры (KBFLAG). Эти флаги могут быть изменены нажатием соответствующих комбинаций спецклавиш как при вызове KBRD, так и при вызове INKEY.

Подпрограмма **NUMKEY** (0F824H) опрашивает клавиатуру с определением номеров (кодов сканирования) нескольких нажатых клавиш и может использоваться для разработки собственных драйверов клавиатуры. Номера нажатых клавиш в порядке возрастания помещаются в специальный внутренний буфер NUMKEY, адрес которого возвращается в регистровой паре HL.

*00 - 9 цифр.	*16 - . цифр.	2C - M M	42 - Г G
*01 - 8 цифр.	*17 - , цифр.	2D - DEL	43 - O O
02	18 - K1	*2E - ИСП (КОМ)	44 - Ъ X
03	19 - 1 !	2F - ПОМ (ПС)	45 - . >
04 - SHIFT	1A - Ц С	30 - 4 \$	46 - : *
*05 - 4 цифр.	1B - Ы Y	31 - E E	47 - - =
*06 - 5 цифр.	1C - Ч ^	32 - П P	48 - 7 '
*07 - 6 цифр.	*1D - 1 цифр.	33 - И I	49 - Ш [
08 - AP2	*1E - 2 цифр.	34 - SPACE	4A - Л L
09 - ТАВ	*1F - 3 цифр.	35 - &	4B - Б B
0A - УПР	20 - K2	36 - ENTER	4C - '
*0B - ГРАФ	21 - 2 "	37 - / ?	4D - Э \
*0C - АЛФ	22 - У U	38 - K4	4E - X H
*0D - + цифр.	23 - В W	39 - 5 %	4F - O
*0E - - цифр.	24 - С S	3A - H N	50 - 8 (
*0F - ВВОД цифр.	*25 - 7 цифр.	3B - P R	51 - Щ]
10 - + ;	26 - УСТ	3C - T T	52 - Д D
11 - Й J	27 - СБРОС	3D -)	53 - Ю @
12 - Ф F	28 - K3	3E - (54 - , <
13 - Я Q	29 - 3 #	*3F -	55 - Ж V
14 - ФИКС	2A - К K	40 - K5	56 - З Z
*15 - 0 цифр.	2B - А A	41 - 6 &	57 - 9)

Значком "*" отмечены клавиши, отсутствующие на клавиатуре РК-86

В регистре А подпрограмма NUMKEY выдает количество нажатых клавиш, а в регистре С – номер нажатой клавиши, записанный в буфере последним (если ни одна клавиша не нажата, то $A=0$, $C=0FFH$, и на выходе устанавливается флаг "Z"). Кроме того, NUMKEY устанавливает пять младших битов в байте (KBFLAG) при нажатии

соответствующих клавиш. На основе подпрограммы NUMKEY разработаны подпрограммы KBRD и INKEY, поэтому после их вызова можно получить текущее состояние буфера NUMKEY с номерами нажатых клавиш (смотри функцию A=1 вектора KBDCTL).

Нумерация клавиш осуществляется в соответствии с их расположением в матрице клавиатуры MC7007 (для ПК-86 номера клавиш преобразуются в кодировку MC7007).

Подпрограмма **INFAST** (0F827H) выполняет быстрое сканирование матрицы клавиатуры с целью определения факта нажатия клавиш управления курсором и спецклавиш УПР (УС), SHIFT (СС), ФИКС (РУС/ЛАТ).

В регистре А возвращаются признаки нажатия указанных клавиш:

D7 – курсор вниз,	D3 – резерв,
D6 – курсор вправо,	D2 – ФИКС (РУС/ЛАТ),
D5 – курсор вверх,	D1 – SHIFT (СС),
D4 – курсор влево,	D0 – УПР (УС).

Если ни одна из указанных клавиш не нажата, устанавливается флаг "Z".

Подпрограммы **KBSND** (0F860H) и **KBSIG** (0F863H) являются служебными векторами драйвера клавиатуры:

0F860H (KBSND) – формирование звука при нажатии клавиши (вызывается в векторе 0F803H); На вход этой подпрограммы при вызове передаются следующие параметры:

- A – код нажатой клавиши;
- D – байт флагов клавиатуры (KBFLAG);
- E – байт режима клавиатуры (KBMODE);

Подменой этого вектора можно формировать звук, отличный от стандартного, при этом необходимо, чтобы новая подпрограмма обработки вектора не попадала в область адресов окна ROM1.

0F863H (KBSIG) – сигнализация переключения флагов клавиатуры (вызывается в векторах 0F803H, 0F81BH). На вход этой подпрограммы передаются:

- A – байт флагов клавиатуры (KBFLAG) до переключения флагов;
- D – байт флагов клавиатуры (KBFLAG) после переключения (при этом необязательно A<>D);
- E – байт режима клавиатуры (KBMODE).

Данный вектор вызывается при нажатии комбинаций спецклавиш, предназначенных для переключения флагов клавиатуры, и его подменой можно включать соответствующий звуковой сигнал или отображать флаги клавиатуры, например, в правом верхнем углу экрана (в исходном состоянии стоит "заглушка" RET). Требования по размещению подпрограммы обработки этого вектора – такие же, как и для вектора 0F860H (вне окна ROM1).

Заметим, что в подпрограммах обработки обоих векторов 0F860H, 0F863H запрещается использовать вектора, реализованные в ROM1: 0F803H, 0EB06H, 0F812H, 0F81BH, 0F827H, 0F851H.

ДРАЙВЕР ДИСПЛЕЯ

В таблице стандартных подпрограмм BIOS драйверу дисплея отведены следующие базовые вектора:

0F80CH – управление экраном (режимы, цвет, окна);
0F80FH – управление выводом символа и курсора;
0F82AH – вывод графики;
0F82DH – система динамического распределения ОЗУ.

Каждый из этих векторов содержит набор подпрограмм (функций), предназначенных для управления и для вывода информации. Выбор требуемой функции осуществляется по ее номеру, указываемому в регистре А перед вызовом соответствующего вектора, при этом соблюдаются следующие соглашения:

- все регистры сохраняются, кроме АХ;
- в качестве параметров используются только ВС, DE, HL;
- на выходе некоторых функций устанавливаются флаги "СУ" и "Z", свидетельствующие о выполнении (невыполнении) заданных условий.

Остальные вектора BIOS, связанные с выводом на экран (0F809H, 0F815H, 0F818H, 0F81EH, 0F83CH) являются производными, т.е. основываются на функциях управляющих векторов. Так, например, вектор 0F809H для вывода символа в очередную позицию курсора использует функцию А=0 вектора 0F80FH, а вектора 0F815H, 0F818H вызывают вектор 0F809H.

Далее будем рассматривать только вектора 0F80CH, 0F80FH, 0F82AH и 0F82DH.

Набор функций, предоставляемых рассматриваемыми векторами, позволяет как изменять значения параметров, так и получать их текущие значения. Такая возможность предусмотрена специально для того, чтобы пользователь перед изменением режимов мог сохранить текущие значения, а потом восстановить их. Это правило должно быть положено в основу использования функций драйвера: в начале работы программы изменяемые параметры сохраняются, а в конце работы – восстанавливаются (это не относится к специальным программам предназначенным для резидентной настройки соответствующих режимов).

Итак, в состав "TV-PRO" входят четыре группы базовых функций:

- 1). Функции вектора 0F80CH для управления экраном, обеспечивающие управление режимами экранов, переключение их номеров, установку цветов, а также определение размеров и положения текущего окна вывода;
- 2). Функции вектора 0F80FH для управления выводом символа и курсора, позволяющие формировать практически любые символы и курсор заданной формы в пределах текущего окна экрана;
- 3). Функции вектора 0F82AH для вывода графики: точки, линии, прямоугольника, окружности, заполнения ограниченной области.
- 4). Функции вектора 0F82DH для построения системы динамического распределения ОЗУ в PRODOS.

Драйвер "TV-PRO" занимает адреса ОЗУ 0F3C0H..0F6FFH под оверлейную область и рабочие ячейки, а адреса 0F7C0H..0F7FFH – под внутренний стек. Кроме того, для внутренних нужд драйвера используется область ОЗУ страницы 7 с адресами 0D000H..0EFFFH (или, что то же самое, в сегменте 31 по адресам 1000H..2FFFH). Программы пользователя не должны размещать в указанных областях свои данные.

Управление экраном (ВЕКТОР 0F80CH)

Драйвер дисплея почти в полной мере реализует возможности компьютера "Orion-Pro" по управлению экранами: пользователь может переключать номера экранов, их

цветовые режимы, выводить информацию на экран сразу или в теневом режиме. Ограничением является невозможность работы с широким экраном под номером 0.

Рассмотрим понятия доступного и видимого экранов. Экран представляет собой область видеопамати компьютера, выбранную для вывода и отображения данных на экране дисплея. Местоположение экрана в пределах страниц ОЗУ (адрес начала экрана) определяется его номером, а занимаемый объем (длина экранной области и число экранных "плоскостей") – заданным цветовым режимом. Так, например, 4-х цветный экран с номером 1 располагается, начиная с адреса 8000H, и занимает 2 экранные "плоскости" – в 0-й и 1-й страницах ОЗУ. Для ширины экрана 384 точки длина экранной области (в одной "плоскости") составляет 12 Кбайт, а для ширины 512 точек – 16 Кбайт.

Формирование изображения на экране заключается в заполнении данными выбранной экранной области памяти в соответствии с установленным цветовым режимом и заданными кодами цветов. Эту задачу как раз и выполняет драйвер дисплея. Экранную область памяти, к которой в данный момент может обращаться драйвер дисплея для чтения и записи информации, будем называть активным или доступным экраном. Для доступного экрана логически назначается цветовой режим (число обрабатываемых драйвером экранных "плоскостей" и их длина), номер экрана (начальный адрес) и коды цветов, в соответствии с которыми будет заполняться экранная память. Эти параметры могут быть выбраны с помощью рассматриваемых ниже функций управления доступным экраном.

Под видимым экраном будем понимать область видеопамати, отображаемую в данный момент на дисплее. Отображение информации видимого экрана осуществляется аппаратно контроллером дисплея, который путем постоянного сканирования видеопамати преобразует экранные байты в аналоговый видеосигнал и передает его на дисплей. Видимый экран характеризуется аппаратно установленными (путем записи соответствующих кодов в порты компьютера) цветовым режимом отображения, номером экрана и палитрой отображения цветов (далее будем рассматривать аппаратные палитры, поддерживаемые основной платой компьютера, так как схема управления палитрами на "Мультикарте" драйвером не поддерживается). Назначение параметров видимого экрана осуществляется с помощью соответствующих функций управления видимым экраном (по аналогии с доступным), при этом программист освобождается от необходимости напрямую обращаться к портам управления. Кроме того, изображение видимого экрана можно вообще отключить (запретить подачу видеосигнала на дисплей) или включить. Эта возможность может быть использована при переключении цветовых режимов видимого экрана для исключения эффекта "мелькания".

Таким образом, драйвер дисплея обеспечивает управление параметрами как доступного, так и видимого экранов, однако может обращаться только к памяти доступного экрана. Как правило, номера и цветовые режимы доступного и видимого экранов совпадают, т.е. один и тот же экран используется и драйвером, и контроллером дисплея, и формируемое драйвером изображение сразу высвечивается на экране. Если для доступного и видимого экранов установить различные цветовые режимы, можно получить определенные эффекты.

Для каждого типа экрана может быть назначен один из следующих режимов:

- 0 – монохромный 384x256 точек
- 1 – монохромный 512x256 точек
- 2 – псевдоцветной 384x256 точек
- 3 – псевдоцветной 512x256 точек
- 4 – 4-цветный 384x256 точек
- 5 – 4-цветный 512x256 точек

- 6 – 8-цветный 384x256 точек
- 7 – 8-цветный 512x256 точек
- 8 – 16-цветный 384x256 точек
- 9 – 16-цветный 512x256 точек
- 10 – 16-цветный 48x256 байт (с групповым ко-
- 11 – 16-цветный 64x256 байт дированием цвета)

В цветовых режимах 6–9 номера экранов могут принимать значения 0,1, в остальных режимах – 0,1,2,3. В момент назначения режима для доступного экрана никаких видимых изменений не происходит – просто данные, впоследствии выводимые на этот экран, будут представляться в виде, соответствующем заданному режиму.

Для видимого экрана режим назначается сразу, если его изображение было включено (если изображение было выключено, то назначение режима видимого экрана откладывается до его включения). При переключении номеров видимого и доступного экранов их соответствующие цветовые режимы не изменяются.

Рассмотрим функции вектора 0F80CH, связанные с назначением режимов экранов (здесь и далее слева в скобках указано значение регистра А – номер функции):

(0) Установить режим доступного экрана

Исходные данные: С – номер режима (0..11);
Выходные данные: нет.

Назначение режимов с нечетными номерами (ширина экрана 512 точек) для доступного экрана 0 заблокировано, так как доступ к экранным ячейкам, начиная с адреса 0F000H, драйвером не обеспечивается.

(1) Получить режим доступного экрана

Исходные данные: нет;
Выходные данные: С – номер режима (0..11);

(2) Установить режим видимого экрана

Исходные данные: С – номер режима (0..11);
Выходные данные: нет.

(3) Получить режим видимого экрана

Исходные данные: нет;
Выходные данные: С – номер режима (0..11);

(4) Установить номер доступного экрана

Исходные данные: С – номер экрана (0..3 или 0,1);
Выходные данные: нет.

(5) Получить номер доступного экрана

Исходные данные: нет;
Выходные данные: С – номер экрана (0..3 или 0,1).

(6) Установить номер видимого экрана

Исходные данные: С – номер экрана (0..3 или 0,1);
Выходные данные: нет.

(7) Получить номер видимого экрана

Исходные данные: нет;
Выходные данные: С – номер экрана (0..3 или 0,1).

После переключения доступного и видимого экранов из режимов 0–5 или 10–11 в режимы 6–9 и обратно программист должен назначить требуемый номер экрана так как в противном случае он может быть некорректно определен.

(8) Выключить изображение видимого экрана

Исходные данные: нет;

Выходные данные: нет

(9) Включить изображение видимого экрана

Исходные данные: нет;

Выходные данные: нет.

При формировании содержимого доступного экрана используются понятия логических цвета фона и цвета изображения (переднего плана). Формирование содержимого доступного экрана при выводе информации будет зависеть от установленного для него режима и заданных кодов цветов.

Для видимого экрана набор отображаемых цветов зависит от его режима и установленной аппаратной палитры (схема управления палитрами, расположенная на "Мультикарте", драйвером не поддерживается).

Цвета доступного экрана

В монохромном и псевдоцветном режимах (0,1,2,3) для доступного экрана используется только 2 цвета: 0и1 (как для фона, так и для переднего плана).

В 4-цветных режимах (4,5) коды цветов принимают следующие значения:

0 – черный или белый	2 – зеленый
1 – синий	3 – красный

В 16-цветных режимах (8..11) коды цветов могут быть следующими:

0 – черный	8 – темно-серый
1 – синий	9 – светло-синий
2 – зеленый	10 – светло-зеленый
3 – голубой	11 – светло-голубой
4 – красный	12 – светло-красный
5 – малиновый	13 – светло-малиновый
6 – коричневый	14 – желтый
7 – светло-серый	15 – белый

В 8-цветных режимах (6,7) для доступного экрана используются коды цветов 0..7.

Цвета (палитры) видимого экрана

В монохромных режимах (0,1) номер аппаратной палитры принимает значения 0,1 (зеленые символы соответственно на черном или белом фоне).

В псевдоцветных режимах (2,3) номер палитры может принимать 256 значений от 00H до 0FFH, при этом старший ниббл определяет цвет отображения погашенных точек видео ОЗУ, а младший – засвеченных точек.

В 4-цветных режимах (4,5) действуют две палитры:

0 – черный, красный, зеленый, синий;
1 – белый, красный, зеленый, синий.

Для 8-цветных и 16-цветных режимов аппаратная палитра только одна и она не переключается. Функции для работы с цветами:

(10) Установить текущий цвет фона доступного экрана

Исходные данные: С – код цвета (0..15);

Выходные данные: нет.

(11) Получить текущий цвет фона доступного экрана

Исходные данные: нет;

Выходные данные: С – код цвета (0..15).

(12) Установить текущий цвет изображения доступного экрана

Исходные данные: С – код цвета (0..15);

Выходные данные: нет.

(13) Получить текущий цвет изображения доступного экрана

Исходные данные: нет;

Выходные данные: С – код цвета (0..15).

При переключении режимов доступного экрана программист должен самостоятельно выполнять инициализацию цветов изображения и фона доступного экрана, в противном случае цветовые атрибуты будут не определены.

(14) Установить код аппаратной палитры видимого экрана

Исходные данные: С – номер палитры (0..255);

Выходные данные: нет,

(15) Получить код аппаратной палитры видимого экрана

Исходные данные: нет;

Выходные данные: С – номер палитры (0..255);

Для каждого из режимов видимого экрана драйвер хранит свой код аппаратной палитры, поэтому при переключении из режима в режим автоматически включается соответствующая ранее установленная палитра.

Вывод данных на экран с помощью "TV-PRO" осуществляется в так называемое окно вывода. Окном будем считать прямоугольный фрагмент доступного экрана, в пределах которого происходит вывод графической информации. Возможно определение только одного текущего окна. Максимальные размеры окна ограничены размерами доступного экрана.

Началом координат в окне считается левый верхний угол (0, 0). Вывод данных за пределами окна (при $X < 0$ или $X \geq$ ширины окна или $Y < 0$ или $Y \geq$ высоты окна) блокируется. При установке окна задается его смещение относительно левого верхнего угла доступного экрана, ширина и высота окна. В момент установки окна никаких изменений в содержимом доступного экрана не происходит.

Функции для работы с окном:

(16) Установить текущее окно

Исходные данные: С – смещение окна по Y (0..255), DE – смещ.окна по X (0..383/511), HL – ширина окна (1..384/512), B – высота окна (1..256).

Выходные данные: "CY" – признак ошибки.

(17) Получить текущее окно

Исходные данные: нет;

Выходные данные: С – смещение окна по Y (0..255), DE – смещ.окна по X (0..383/511), HL – ширина окна (1..384/512), B – высота окна (1..256).

ВНИМАНИЕ: При переключении режимов и номеров доступного экрана окно всегда устанавливается на весь экран!

(18) Очистить текущее окно цветом фона

Исходные данные: нет,

Выходные данные: нет.

Функции установки и очистки окна обнуляют позицию курсора и координаты начала линии. В драйвере имеется возможность сохранения и последующего восстановления содержимого текущего окна доступного экрана. Длина буфера для сохранения окна зависит от размеров окна, а размерность буфера (количество требуемых сегментов ОЗУ) – от текущего режима доступного экрана. Данные из каждой экранной плоскости сохраняются в отдельном сегменте, при этом адреса размещения данных во всех сегментах одинаковы. Например, для 8-цветного режима размерность буфера равна 3, т.е. для сохранения окна требуется 3 сегмента.

(19) Установить адрес буфера для сохранения окна

Исходные данные: CBED – номера сегментов ОЗУ для сохранения содержимого окна (С – для режимов 0,1,2,3, СВ – для режимов 4,5,10,11, СВЕ – для режимов 6,7, CBED – для режимов 8,9); HL – начальный адрес размещения данных в сегментах (во всех одинаковый).

Выходные данные: нет.

При вызове функции установки адреса буфера допускается указывать как адрес в пределах сегмента ОЗУ (0..3FFFH), так и действительный адрес в пределах страницы ОЗУ (0..0FFFFH); в последнем случае необходимая коррекция старшего байта адреса будет выполнена автоматически. Указанное замечание относится также к другим функциям установки буферов: установки адреса ЗГ. адреса шаблона курсора, адреса буфера для сохранения информации под курсором.

(20) Получить адрес буфера для сохранения окна

Исходные данные: нет,

Выходные данные: CBED – номера сегментов ОЗУ для сохранения содержимого окна (С – для режимов 0,1,2,3, СВ – для режимов 4,5,10,11, СВЕ – для режимов 6,7, CBED – для режимов 8,9); HL – начальный адрес размещения данных в сегментах (во всех одинаковый).

(21) Определить длину буфера для сохранения текущего окна

Исходные данные: нет;

Выходные данные: DE – длина буфера в байтах (для всех сегментов одинакова), необходимая для сохранения текущего окна.

(22) Сохранить текущее окно

Исходные данные: С – признак сдвига буфера: С = 00H – ускоренный режим сохранения без сдвига буфера; С = 0FFH – режим сохранения с последующим сдвигом буфера влево. Выходные данные: "CY" – признак переполнения буфера.

При сохранении текущего окна в буфер записывается только его содержимое без дополнительных параметров. Если входное значение С=0FFH, то после сохранения информации окна содержимое буфера сдвигается влево для приведения его к нулевому смещению в пределах байта.

(23) Восстановить текущее окно

Исходные данные: С – признак сдвига буфера: С = 00H – ускоренный режим восстановления без сдвига буфера; С = 0FFH – режим восстановления со сдвигом буфера вправо

Выходные данные: "CY" – признак переполнения буфера. Значение параметра С при восстановлении окна должно быть таким же, каким оно было при сохранении окна.

Размеры восстанавливаемого окна должны совпадать с размерами, действовавшими при сохранении, при этом допускается установка нового положения окна для его перемещения по экрану. Смещение окна по Y относительно верхней границы экрана может быть выбрано произвольным, а допустимые значения смещения

окна по X относительно левой границы экрана зависят от величины входного параметра C.

Если $C=0$, то смещение восстанавливаемого окна по X в пределах байта должно оставаться прежним, т.е. окно может быть перемещено по X на величину, кратную 8. Этим режимом целесообразно пользоваться при восстановлении окна на "Старом" месте экрана без перемещения.

Если $C=0FFH$, то смещение восстанавливаемого окна по X может быть выбрано без ограничений. В этом случае предполагается, что окно было сохранено с приведением содержимого буфера к нулевому смещению ($C=0FFH$ при вызове функции сохранения окна). В процессе восстановления окна содержимое буфера сначала сдвигается вправо на величину, определяемую смещением окна в пределах левого столбца байтов. После восстановления буфер снова сдвигается влево до нулевого смещения. Таким образом, обеспечивается перемещение окна в любую точку экрана с некоторой потерей скорости.

Для реализации стековой системы автоматического сохранения/восстановления окон могут быть использованы функции вектора $0F82DH$, обеспечивающие распределение/освобождение динамической памяти (для хранения информации окон и их параметров) в пределах заданных сегментов ОЗУ.

(24) Скроллинг содержимого окна вверх

Исходные данные: H – высота скроллируемой части окна (от верхней границы окна); L – величина сдвига окна (высота строки), $0 < L \leq H$; C – смещение скроллируемой части окна относительно верхней границы окна, $C+H \leq$ высоты окна;

Выходные данные: "CY" – признак ошибки.

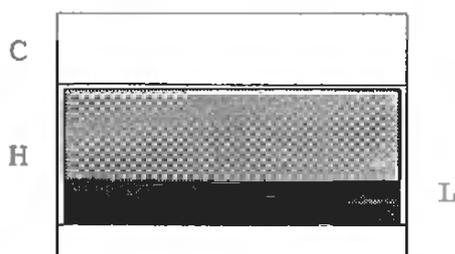
(25) Скроллинг содержимого окна вниз

Исходные данные: H – высота скроллируемой части окна (от верхней границы окна); L – величина сдвига окна (высота строки), $0 < L \leq H$; C – смещение скроллируемой части окна относительно верхней границы окна, $C+H \leq$ высоты окна; Выходные данные: "CY" – признак ошибки.

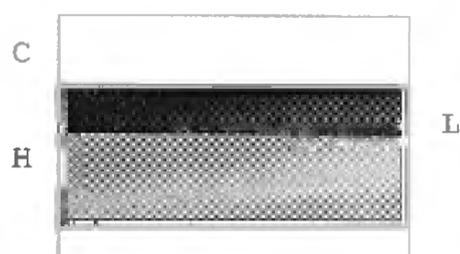
Скроллинг окна заключается в сдвиге его содержимого вверх (вниз) на заданное число пикселей с последующей очисткой "освободившейся" нижней (верхней) строки.

При выполнении операций скроллинга текущее окно разделяется в общем случае на 4 части:

Скроллинг содержимого окна
вверх



Скроллинг содержимого окна
вниз



Верхняя часть окна высотой [C] пикселей не скроллируется. В середине окна находится скроллируемая область высотой [H], которая в свою очередь делится на две части – копируемую (серая) и очищаемую цветом фона (темная) с высотой [L]. Оставшаяся нижняя часть окна не изменяется

(26) Очистка до конца строки

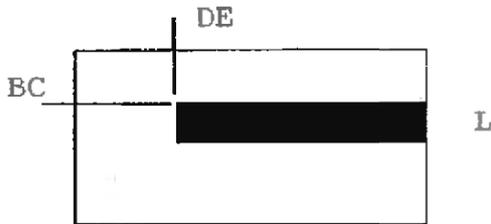
Исходные данные: BC – текущая графическая позиция Y; DE – текущая

графическая позиция X; L – высота строки; Выходные данные: "CY" – строка вне окна;

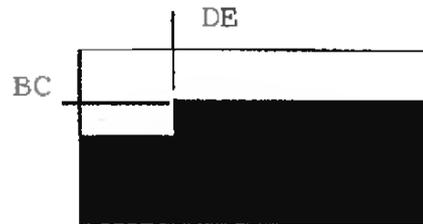
(27) Очистка до конца окна

Исходные данные: BC – текущая графическая позиция Y; DE – текущая графическая позиция X; L – высота строки; Выходные данные: "CY" – строка вне окна;

Очистка до конца строки



Очистка до конца окна



Описанные функции работы с окном, имеющиеся в данном векторе 0F80CH, не зависят от размеров установленного шрифта, т.е. являются универсальными. Пользователь может применять их по своему усмотрению, например, при разработке специальных процедур вывода символов.

Функция (4) в векторе управления выводом символа и курсора 0F80FH позволяет выполнить некоторые операции с окном, действие которых зависит от текущих размеров символов.

Управление выводом символа и курсора (ВЕКТОР 0F80FH)

Вывод символа и курсора осуществляется в графическом режиме, и описываемые ниже процедуры можно использовать для формирования других двухцветных изображений. Далее рассматриваются функции вектора 0F80FH, обеспечивающие вывод информации в текущее окно доступного экрана в заданном цветовом режиме. Для вывода графических символов задаются параметры знакогенератора (ЗГ) с шаблонами символов, и режим вывода символов (см. функции 7–12). При задании знакогенератора указывается его адрес, шаг следования шаблонов (в байтах), размеры символов и шаг приращения координат курсора в пикселях по X и по Y. Можно иметь несколько знакогенераторов со шрифтами разного размера, при этом каждый знакогенератор должен размещаться целиком в сегменте ОЗУ (в одном сегменте может быть несколько знакогенераторов). Шаблон символа задается последовательностью байтов: сначала следует левый столбец байтов (начиная с левого верхнего угла шаблона – вниз), за ним подряд – все остальные столбцы (если есть), причем количество байтов в каждом столбце равно высоте символа, а количество столбцов определяется шириной символа и смещением шаблона в левом столбце байтов.

Теоретически размеры знакоместа могут быть в пределах от 1 до 256 пикселей как по X, так и по Y. Однако, следует учитывать тот факт, что ЗГ должен целиком уместиться в выбранном сегменте ОЗУ, т.е. суммарная длина всех шаблонов не может быть больше 16 Кбайт. Отсюда следует, что в полном знакогенераторе, содержащем шаблоны 256 символов, длина одного шаблона не должна превышать $16384/256 = 64$ байт. Для большинства применений этого достаточно, но если Вам необходимо иметь символы большего размера, ЗГ можно разделить на несколько неполных частей, разместив каждую из них в отдельном сегменте ОЗУ, причем одна такая часть будет соответствовать определенному диапазону символов. Переключение между частями ЗГ в зависимости от кода выводимого символа должно выполняться в программе пользователя с помощью функции A=11 вектора 0F80FH с последующим приведением кода символа к нулю для привязки к началу соответствующей части ЗГ. Решение

проблемы упрощается, если Вам нужны всего лишь несколько крупноразмерных символов, а не полный ЗГ. В этом случае шаблоны всех этих символов, возможно, вместятся в одном сегменте ОЗУ. Рассмотрим шаблон символа "А" с размером знакоместа 12x8 пикселей.

В системе имеется набор стандартных шрифтов для 4-х типов кодировок, которые могут быть распакованы в указанный сегмент ОЗУ по требуемому адресу (см. функцию 13):

- ALT (альтернативная);
- КОИ-8;
- КОИ-7/Н2 (большие латинские и русские);
- КОИ-7/Н1 (только русские).

Для реализации знакогенератора КОИ-7/Н0 (только латинские) можно использовать первую половину кодовой таблицы ALT или КОИ-8.

Заметим, что драйвер не высекает старший бит символов при использовании наборов КОИ-7. Любой знакогенератор содержит шаблоны 256 символов (в КОИ7/Н2 или КОИ-7/Н1 обе половины знакогенератора совпадают).

Базовый размер стандартных символов составляет 6x10 пикселей, однако при распаковке стандартного шрифта можно установить символы увеличенной вдвое ширины (12 пикселей) или/и увеличенной вдвое высоты (20 пикселей).

Распаковка стандартного шрифта производится только в том случае, если в заданном сегменте по указанному адресу имеется достаточно места. При вызове функции распаковки можно указать признак автоматической установки (активизации) знакогенератора.

Кроме того, при распаковке ЗГ можно задать признак формирования шаблона стандартного курсора, что приведет к записи в конец знакогенератора нескольких байтов 0FFH (количество записываемых байтов равно длине шаблона одного символа). Пользователь может впоследствии изменять размеры стандартного курсора в пределах знакоместа, не меняя его шаблон (или вообще заменить шаблон, размеры и форму курсора, сделав его, например, в виде крестика или рамочки) Если дополнительно задан признак активизации ЗГ, то автоматически устанавливается стандартный курсор в виде черточки, причем его размер будет соответствовать размеру символа выбранного стандартного шрифта.

От использования стандартного курсора можно отказаться. В этом случае должен быть установлен новый шаблон курсора, который представляется в ОЗУ аналогично шаблону символа. Размеры и форма курсора могут быть выбраны независимо от параметров текущего шрифта (см. функции 16-19).

Функции вывода и гашения курсора (2), (3) используются подпрограммой ввода с клавиатуры и драйвером мыши, а также применяются для реализации режима подчеркивания символов и режима непрерывного вывода курсора в функции (0). Однако пользователь может использовать указанные функции и по своему усмотрению (например, для вывода движущихся образов с сохранением информации экрана) с обязательным возвратом прежних параметров курсора.

Для вывода символа и курсора может быть по отдельности задан один из режимов наложения шаблона на содержимое окна экрана (см. функции 7,8,14,15):

- режим полной замены (когда единичным битам шаблона на экране будут соответствовать точки, засвеченные цветом изображения, а нулевым битам шаблона точки, погашенные цветом фона);

- режим наложения по ИЛИ (фон не меняется, цветом изображения засвечиваются точки, соответствующие единичным битам шаблона);
- режим наложения по И (наоборот – меняется только фон, соответствующий нулевым битам шаблона);
- режим наложения по ИСКЛЮЧАЮЩЕМУ ИЛИ (независимо от установленных кодов цветов фона и изображения инвертируются те экранные биты, которым в шаблоне соответствуют единицы) – при повторном выводе в то же место экрана по тому же шаблону происходит восстановление прежней информации экрана.

Кроме того, может быть задан признак инверсии, когда в каждом из перечисленных режимов (кроме последнего) заданные цвета фона и изображения меняются местами.

Рассмотрим интерфейс функций вектора 0F80FH (слева в скобках указан номер функции, задаваемый в регистре A):

(0) Вывести графический символ в очередную поз. Курсора (используется в 0F809H)

Исходные данные: C – код символа;

Выходные данные: нет.

(1) Вывести графический символ в заданную позицию окна

Исходные данные: L – код символа;

DE – X (левого верхнего угла символа);

BC – Y (левого верхнего угла символа);

Выходные данные: "CY" – символ вне окна.

Функция (1) не обрабатывает управляющие коды и ESC-последовательности, а также не изменяет позицию курсора.

(2) Отобразить графический курсор в заданной позиции окна

Исходные данные: BC – позиция Y;

DE – позиция X;

Выходные данные: "CY" – вне окна.

Если установлен режим сохранения информации под курсором, то перед выводом курсора содержимое соответствующей части окна экрана автоматически записывается в заранее заданный буфер. При этом информация экрана впоследствии может быть восстановлена (курсor погашен).

(3) Погасить графический курсор в заданной позиции окна

Исходные данные: BC – позиция Y; DE – позиция X;

Выходные данные: "CY" – вне окна.

Курсор убирается следующим образом: если в данный момент установлен режим сохранения информации под курсором, то происходит ее восстановление из заданного буфера, в противном случае курсор просто выводится повторно, при этом, если курсор выводился с наложением по ИСКЛЮЧАЮЩЕМУ ИЛИ, изображение восстановится автоматически.

(4) Выполнить управляющую подфункцию

Исходные данные: C – номер подфункции:

- 0 – курсор в левый верхний угол
- 1 – курсор влево
- 2 – курсор вправо
- 3 – курсор вверх
- 4 – курсор вниз
- 5 – табуляция
- 6 – возврат каретки
- 7 – перевод строки
- 8 – BackSpace (забой влево)
- 9 – установка символьного окна:
 - L–ширина окна, H–высота окна,
 - E–смещение по X, D–смещение по Y
 - "CY" – признак ошибки.
- 10 – очистка окна
- 11 – очистка до конца окна
- 12 – очистка до конца строки
- 13 – скроллинг содержимого окна вверх
- 14 – скроллинг содержимого окна вниз
- 15 – вставить строку
- 16 – удалить строку

Выходные данные: нет.

Функция (4) введена в состав драйвера специально для создания "внешних" подпрограмм вывода символа на основе функции (0) или (1). Особенностью функции (4) является привязка управляющих подфункций к текущим размерам символов и к позиции курсора.

Подфункция (4/1) возвращает флаг "CY" при переходе курсора из позиции Home текущего окна в последнюю позицию, а подфункция (4/2) – при переходе курсора из последней позиции окна в позицию Home (скроллинг не выполняется).

Подфункции (4/3), (4/4) возвращают флаг "CY" при переходе курсора соответственно из верхней строки на нижнюю и из нижней строки на верхнюю.

На выполнение подфункции (4/2) влияет состояние бита C5, на выполнение подфункции (4/7) – бита C6, описывающих режим вывода символов (см. функцию 7).

Управляющая подфункция BackSpace (4/8) стирает символ слева от курсора и перемещает курсор на его место. Забой символа цветом фона происходит всегда, независимо от текущего режима наложения символов.

Подфункция (4/9) служит для установки символьного окна. Параметры окна указываются при вызове в регистрах DE, HL (все размеры задаются в количестве символов). Курсор перемещается в левый верхний угол установленного окна

(5) Установить символьную позицию курсора

Исходные данные: H – символьная позиция по Y;

L – символьная позиция по X;

Выходные данные: нет.

(6) Получить позицию курсора

Исходные данные: нет;

Выходные данные: H – символьная позиция по Y;

L – символьная позиция по X;

BC – графическая позиция по Y;

DE – графическая позиция по X.

Функции (5) и (6) используются соответственно в векторах 0F83CH и 0F81EH.

Необходимо заметить, что функция (6) возвращает и графические координаты курсора для того, чтобы можно было отобразить / погасить курсор в текущей позиции окна (например, из драйвера клавиатуры) с помощью функций (2), (3).

(7) Установить режим вывода символов

Исходные данные: C – код режима, определяется значением следующих битов регистра C:

C7 – резерв (0);

C6 – признак разрешения скроллинга при выводе ПС в последней строке окна (при C6=0 действие ПС соответствует перемещению курсора вниз);

C5 – признак разрешения автоперевода строки при выводе в последнюю позицию строки (при C5=0 лишние символы отсекаются);

C4 – признак разрешения обработки управляющих функций 00H..1FH (если C4=0, то вместо функций высвечиваются соответствующие символы)

C3 – резерв (0);

C2 – признак инверсии символа;

C1, C0 – задают режим наложения в графическом режиме:

00 – наложение с полной заменой (если в шаблоне 0, то точка гасится цветом фона, если 1 засвечивается цветом изображения);

01 – наложение по ИЛИ (если в шаблоне 0, то вывод блокируется, если 1 – производится засветка цветом изображения);

10 – наложение по И (если в шаблоне 0, то производится гашение точки цветом фона, если 1 – вывод блокируется);

11 – наложение по исключающему ИЛИ (независимо от цветов фона и изображения: если в шаблоне 1, то производится инверсия цвета, если 0 – цвет не меняется);

Выходные данные: нет.

Биты C6, C5 анализируются при выполнении управляющих кодов соответственно "ПС" и "курсор вправо" в функциях (0) и (4), бит C4 – при выводе символов с

помощью функции (0), биты C2..C0 – при формировании изображения символа в функциях (0) и (1).

На значения битов C6, C5 воздействуют управляющие последовательности ESC '2' – ESC '5', на бит C4 – 10H (^P), на бит C2 – ESC '6' и ESC '7'

(8) Получить режим вывода символов

Исходные данные: нет;

Выходные данные: C – код режима.

(9) Установить размеры графического символа

Исходные данные: D – ширина символа;

E – высота символа;

B – шаг вывода символов по X;

C – шаг вывода символов по Y;

L – смещение левого столбца битов шаблона в левом столбце байтов (0..7);

Выходные данные: нет.

(10) Получить размеры графического символа

Исходные данные: нет;

Выходные данные: D – ширина символа;

E – высота символа;

B – шаг вывода символов по X;

C – шаг вывода символов по Y;

L – смещение левого столбца битов шаблона в левом столбце байтов (0..7);

Шаг вывода символов по X и по Y определяется соответственно шириной и высотой символа, однако эти параметры можно увеличивать (или уменьшать) для получения какого-либо эффекта, например, для вывода шрифта вразрядку.

(11) Установить адрес текущего знакогенератора

Исходные данные: C – номер сегмента ОЗУ со знакогенератором;

NL – начальный адрес ЗГ в пределах сегмента ОЗУ (значение NL может задаваться в пределах страницы ОЗУ, так как два старших бита N игнорируются);

DE – расстояние в байтах между соседними шаблонами в ЗГ

Выходные данные: нет.

Функция установки адреса знакогенератора позволяет оперативно менять шрифты (если их предварительно загрузить в нужные сегменты ОЗУ).

(12) Получить адрес текущего знакогенератора

Исходные данные: нет;

Выходные данные: С – номер сегмента ОЗУ со знакогенератором;
HL – начальный адрес ЗГ в пределах сегмента ОЗУ.
DE – расстояние в байтах между соседними шаблонами в ЗГ.

(13) Установить (распаковать) стандартный шрифт из ПЗУ

Исходные данные: С – номер сегмента ОЗУ для распаковки;
HL – начальный адрес ЗГ в сегменте;
В – код распаковки:
В7 – признак активизации ЗГ (и курсора при В6=1) после распаковки;
В6 – признак зачки стандартного курсора (шаблон курсора дописывается в конец ЗГ);
В5–В4 – резерв (00);
В3 – признак высоких букв;
В2 – признак широких букв;
В1,В0 – номер кодировки:
00 – ALT;
01 – КОИ–8;
10 – КОИ–7/Н2;
11 – КОИ–7/Н1.

Выходные данные: DE – длина знакогенератора с учетом шаблона курсора (если В6=1) или без учета шаблона курсора (если В6=0);
"СУ" – выход за пределы сегмента (шрифт не распаковывается).

Если задано В7=1, то функция (13) распаковывает стандартный шрифт из ПЗУ в ОЗУ по указанному адресу с автоматической настройкой всех параметров, задаваемых функциями (9) и (11). Кроме того, если задано В6=1, то автоматически устанавливается стандартный курсор в виде черточки с изменением всех параметров, задаваемых функциями (16) и (18). Текущий режим вывода курсора не изменяется.

При В7=0 операция распаковки знакогенератора выполняется без его активизации (и без установки курсора для В6=1), т.е. параметры текущего шрифта и курсора не изменяются.

Заметим, что если знакогенератор не умещается в заданном сегменте, никаких изменений не происходит, а на выходе функции устанавливается флаг "СУ". Однако, в любом случае выходное значение DE, содержащее длину ЗГ в байтах, является действительным. Последнее свойство может быть использовано для определения длины ЗГ без распаковки (перед распаковкой), если на входе функции (13) задать заведомо недопустимое значение HL=0FFFFH.

(14) Установить режим вывода графического курсора

Исходные данные: С – код режима, задаваемый следующими разрядами:

- C7 – резерв (0);
- C6 – признак непрерывного отображения курсора при выводе символа;
- C5 – признак разрешения вывода курсора при вводе с клавиатуры;
- C4 – признак мигания курсора при вводе с клавиатуры;
- C3 – признак сохранения информации под курсором;
- C2 – признак инверсии;
- C1–C0 – задают режим наложения;
- 00 – наложение с полной заменой;
- 01 – наложение по ИЛИ;
- 10 – наложение по И;
- 11 – наложение по исключающему ИЛИ;

Выходные данные: нет.

Бит C6 введен специально для обеспечения совместимости с существующими программами, рассчитанными на использование аппаратного курсора в среде CP/M-80. Режим непрерывного вывода курсора не рекомендуется использовать при разработке новых программ. Заметим, что если при вызове функции (14) задано C6=1, то курсор сразу отображается, в противном случае – сразу гасится.

Биты C5, C4 являются информационными и анализируются подпрограммой ввода символа с клавиатуры KBRD (0F803H).

Комбинация битов C6 = C5 = 1 запрещена.

Биты C3..C0 используются функциями (2) и (3) при формировании изображения курсора.

На значения битов C6, C5, C4 влияют управляющие последовательности ESC ':', ESC ';', ESC '<', ESC '=':

Режим курсора	ESC	C6	C5	C4
Непрерывный	':'	1	0	0
Отсутствует	';'	0	0	0
Мигает в KBRD	'<'	0	1	1
Не мигает в KBRD	'='	0	1	0

(15) Получить режим вывода графического курсора

Исходные данные: нет;

Выходные данные: С – код режима.

(16) Установить размеры графического курсора

Исходные данные: D – ширина курсора;

E – высота (толщина) курсора;

B – смещение по X относительно позиции курсора;

C – смещение по Y относительно позиции курсора;

L – смещение шаблона в левом столбце байтов;

Выходные данные: нет.

(17) Получить размеры графического курсора

Исходные данные: нет;

Выходные данные: D – ширина курсора;

E – высота курсора;

B – смещение по X относительно позиции курсора;

C – смещение по Y относительно позиции курсора;

L – смещение шаблона в левом столбце байтов.

(18) Установить адрес шаблона графического курсора

Исходные данные: С – номер сегмента шаблона;

HL – адрес шаблона в сегменте;

Выходные данные: нет.

(19) Получить адрес шаблона графического курсора

Исходные данные: нет;

Выходные данные: С – номер сегмента шаблона;

HL – адрес шаблона в сегменте.

(20) Установить адрес буфера для сохр.данных под курсором

Исходные данные: CBED – номера сегментов ОЗУ;

HL – адрес в пределах сегментов;

Выходные данные: нет.

(21) Получить адрес буфера для сохр.данных под курсором

Исходные данные: нет;

Выходные данные: CBED – номера сегментов ОЗУ;

HL – адрес в пределах сегментов.

(22) Получить длину буфера для сохранения инф. под курс.

Исходные данные: нет;

Выходные данные: DE – длина буфера;

Вывод графики (ВЕКТОР 0F82AH)

В векторе 0F82AH имеется набор функций для вывода точки, пинии, прямоугольника, окружности, заполнения ограниченной области (заливки) и работы со спрайтами.

Координаты выводимых точек – двухбайтовые. Они могут быть отрицательными (в дополнительном коде с единичным старшим битом). Диапазон координат X,Y: – 32768..–1..0..32767 (8000H ..0FFFH..0..7FFFH). Координаты указываются относительно левого верхнего угла текущего окна. Отрицательные значения координат могут быть использованы, например при рисовании линий, выходящих за пределы окна, и в других случаях. Если точка не попадает в текущее окно, то она не выводится.

Функции, связанные с выводом точки (слева в скобках указан номер функции, помещаемый перед вызовом в регистр A):

(0) Засветить точку заданным цветом

Исходные данные: DE – координата X, BC – координата Y, L – код цвета;

Выходные данные: "CY" – признак нахождения вне окна.

(1) Засветить точку цветом изображения

Исходные данные: DE – координата X, BC – координата Y;

Выходные данные: "CY" – признак нахождения вне окна.

(2) Погасить точку цветом фона

Исходные данные: DE – координата X, BC – координата Y;

Выходные данные: "CY" – признак нахождения вне окна.

(3) Проинвертировать точку

Исходные данные: DE – координата X, BC – координата Y;

Выходные данные: "CY" – признак нахождения вне окна.

(4) Определить цвет точки

Исходные данные: DE – координата X, BC – координата Y;

Выходные данные: "CY" – признак нахождения вне окна, иначе: A – код цвета точки;

В режиме группового кодирования дополнительно возвращается признак "NZ", если точка засвечена в ОЗУ переднего плана, и "Z" – если погашена. Во всех остальных режимах всегда возвращается признак "NZ".

Для вывода линии должны быть заданы координаты ее начальной и конечной точек. В драйвере имеется возможность вывода непрерывной и пунктирной линии. Вывод пунктирной линии осуществляется по специальной 16 – разрядной маске. Формирование каждой точки линии происходит в соответствии со значением очередного разряда маски (циклически, начиная со старших разрядов): если бит маски нулевой, то точка гасится, если единичный – то засвечивается.

Пунктирная линия вырождается в непрерывную при значениях маски 0000H (гашение линии) или 0FFFFH (засветка).

Функции для вывода линии:

(5) Установить координаты начальной точки для линии

Исходные данные: DE – координата X начала линии;

BC – координата Y начала линии;

Выходные данные: нет.

(6) Получить координаты начальной точки для линии

Исходные данные: нет;

Выходные данные: DE – координата X начала линии;

BC – координата Y начала линии.

(7) Вывести линию засветкой точек заданным цветом

Исходные данные: DE – координата X конца линии;

BC – координата Y конца линии;

L – код цвета линии;

Выходные данные: нет.

(8) Вывести линию инверсией точек

Исходные данные: DE – координата X конца линии;

BC – координата Y конца линии;

Выходные данные: нет.

(9) Вывести пунктирную линию

Исходные данные: DE – координата X конца линии;

BC – координата Y конца линии;

Выходные данные: нет.

После вывода линии координаты конечной точки запоминаются в качестве начальной точки для вывода следующей линии. Таким образом, можно чертить ломаную линию, не используя каждый раз функцию установки координат начальной точки.

(10) Установить режим вывода пунктирной линии

Исходные данные: C – код режима:

C7–C3 – резерв (00000);

C2 – признак инверсии (цвета фона и изображения меняются местами);

C1, C0:

0 – наложение с полной заменой (если очередной разряд маски 0, то текущая точка линии гасится цветом фона, если 1 засвечивается цветом изображения);

1 – наложение по ИЛИ (цветом изображения засвечиваются только те точки, которым в маске соответствует 1);

2 – наложение по И (цветом фона гасятся только те точки, которым в маске соответствует 0);

3 – наложение по исключающему ИЛИ (независимо от цветов фона и изображения: если в маске 1, то производится инверсия цвета, если 0 – цвет точки не меняется)

HL – Маска линии;

Выходные данные: нет.

(11) Получить текущий режим вывода пунктирной линии

Исходные данные: нет;

Выходные данные: C – номер режима; HL – маска линии.

Функция (10) сбрасывает указатель в маске на старший бит. Совместно с функцией (11) данное свойство может быть использовано, например, для прорисовки углов фигуры (для этого надо последовательно вызвать функции (11) и (10)).

Для вывода прямоугольника в виде рамки необходимо указать координаты его двух противоположных углов. Один из углов задается координатами начала линии (функцией 5), другой – входными параметрами DE, BC.

(12) Вывести прямоугольник в виде рамки заданным цветом

Исходные данные: BC – координата Y одного из углов;

D – координата X одного из углов;

L – код цвета;

Выходные данные: "CY" – выход за пределы окна.

(13) Вывести прямоугольник в виде рамки с инверсией

Исходные данные: BC – координата Y одного из углов;

DE – координата X одного из углов;

Выходные данные: "CY" – выход за пределы окна.

Функция (13) может использоваться для вывода рамки с возможностью последующего стирания. Для вывода прямоугольника, закрашенного заданным цветом (столбца) используется следующая функция.

(14) Вывести столбец, закрашенный заданным цветом

Исходные данные: BC – координата Y одного из углов; DE – координата X одного из углов; L – код цвета;

Выходные данные: "CY" – выход за пределы окна.

При выводе окружности указываются координаты ее центра и радиус. Так как изображение экрана в компьютере "Orion" несколько сжато с боков, функция вывода

окружности автоматически корректирует значение радиуса окружности по вертикали, устанавливая его равным 3/4 от заданного.

(15) Вывести окружность заданным цветом

Исходные данные: BC – координата Y центра;
DE – координата X центра;
H – радиус (по горизонтали);
L – код цвета;

Выходные данные: Нет.

Рассмотрим функции, связанные с заполнением (заливкой) ограниченной области.

(16) Заполнить ограниченную область заданным цветом

Исходные данные: BC – координата Y начальной точки заполнения;
DE – координата X начальной точки заполнения
H – код цвета границы;
L – код цвета заполнения;

Выходные данные: "CY" – начальная точка вне окна.

Особенностью этой функции в режиме группового кодирования является необходимость задания одинаковых цветов границы и заполнения. При выполнении функции (16) в качестве временного буфера используется часть сегмента 31 (страница 7 ОЗУ) с адресами 0EC40H..0EFFFH.

(17) Установить режим заполнения

Исходные данные: C – код режима:
0 – направление вверх и вниз;
1 – направление вправо и влево;

Выходные данные: нет.

(18) Получить режим заполнения

Исходные данные: нет;

Выходные данные: C – код режима:
0 – направление вверх и вниз;
1 – направление вправо и влево.

Для обработки спрайтов используются функции, описанные ниже.

(19) Сохранить область экрана (спрайт) в буфере

Исходные данные: BC – Y левого верхнего угла спрайта в окне доступного экрана;
DE – X левого верхнего угла спрайта в окне доступного экрана;
H – высота спрайта;
L – ширина спрайта;

Выходные данные: "CY", "NZ" – переполнение буфера;

"CY""Z" – выход за пределы окна;

"NC" – нормальное завершение.

(20) Отобразить область экрана (спрайт) из буфера

Исходные данные: BC – Y левого верхнего угла спрайта в окне доступного экрана;

DE – X левого верхнего угла спрайта в окне доступного экрана;

H – высота спрайта;

L – ширина спрайта;

Выходные данные: "CY","NZ" – переполнение буфера;

"CY","Z" – выход за пределы окна;

"NC" – нормальное завершение.

Структура буфера для сохранения/восстановления спрайтов аналогична буферу для сохранения/восстановления окна экрана

(21) Установить адрес буфера для сохранения спрайта

Исходные данные: CBED – номера сегментов ОЗУ;

HL – адрес начала буфера (во всех сегментах одинаковый);

Выходные данные: нет.

(22) Получить адрес буфера для сохранения спрайта

Исходные данные: нет;

Выходные данные: CBED – номера сегментов ОЗУ;

HL – адрес начала буфера (во всех сегментах одинаковый).

(23) Определить длину буфера для сохранения спрайта

Исходные данные: H – высота спрайта;

L – ширина спрайта;

Выходные данные: DE – необходимая длина буфера для сохранения спрайта указанных размеров (для одной экранной плоскости).

СИСТЕМА ДИНАМИЧЕСКОГО РАСПРЕДЕЛЕНИЯ ОЗУ

Наличие относительно большого для 8-разрядных компьютеров объема памяти в ПК "Orion-Pro" (32 сегмента по 16 Кбайт, что соответствует суммарной емкости 512 Кбайт в базовом варианте) вызывает необходимость централизованного управления ей с тем, чтобы исключить конфликты программ пользователя между собой и операционной системой PRODOS. Для этого ресурсы памяти ПК должны выделяться и освобождаться в процессе выполнения программ (т.е. динамически) через единую систему распределения ОЗУ, являющуюся частью операционной системы.

Такая система динамического распределения памяти должна обеспечивать как минимум выполнение следующих функций:

- инициализацию своих внутренних параметров;
- проверку занятости указанной области памяти;
- принудительное выделение указанной области памяти с установкой признака ее занятости;
- запрос на выделение области памяти требуемого объема, при этом система сама находит первую подходящую область и сообщает ее адрес с автоматической установкой признака занятости области;
- освобождение ранее выделенной области памяти.

Так как память в ПК "Orion-Pro" имеет страничную (по 60 Кбайт) и сегментную (по 16 Кбайт) организацию, при разработке системы динамического распределения ОЗУ необходимо было решить вопрос о том, на каком уровне выполнять распределение памяти: на страничном или сегментном.

Выбор был сделан в пользу последнего способа, поскольку, во-первых, при обращении к ОЗУ посегментно (через одно из окон в адресном пространстве процессора) можно получить доступ практически ко всей памяти компьютера, а при постраничном обращении теряется 4 Кбайта на каждую страницу (из-за наличия непереключаемого ОЗУ 0F000..0FFFFH). Во-вторых, драйвер дисплея TV-PRO взаимодействует с ОЗУ также через сегменты, и целый ряд его функций для работы с буфером экрана (например, сохранение и восстановление окна, вывод спрайта и т.д.) требуют указания сегментных адресов памяти, т.е. номеров сегментов и адресов в пределах сегментов (0..3FFFFH).

Таким образом, рассматриваемую систему можно назвать системой сегментного динамического распределения памяти. Недостатком такого подхода можно считать то, что система не может выделить область памяти, находящуюся на границе двух сегментов, а также то, что максимальный объем непрерывной области в адресном пространстве процессора в общем случае не превышает 16 Кбайт

Система распределения памяти обеспечивает работу с многомерными областями памяти, имеющими размерность от 1 до 4. Под размерностью области памяти понимается количество "параллельных" плоскостей ОЗУ, размещенных в разных сегментах, но имеющих одинаковые адреса в пределах сегментов. Данное условие необходимо для реализации функций драйвера TV-PRO, работающих с буфером экрана. Так, при сохранении окна монохромного экрана используется буфер размерностью 1, а при сохранении окна 16-цветного экрана (EGA) - размерностью 4.

Таблица описания области памяти

При выполнении функций распределения ОЗУ должны быть указаны параметры выделяемой (освобождаемой) области памяти, включающие ее начальный адрес и длину. Эти параметры представляются в виде специальной таблицы, называемой ТАБЛИЦЕЙ ОПИСАНИЯ ОБЛАСТИ (ТОО) длиной от 6 до 9 байт:

- размерность области 1..4 (число плоскостей, размещенных в разных сегментах, но имеющих одинаковые начальный адрес и длину) – 1 байт;
- длина области (1..4000H) – 2 байта;
- начальный адрес области в пределах сегмента (0..3FFFH) без учета ее положения в странице ОЗУ, т.е. два старших бита адреса всегда нулевые (такой адрес будем называть сегментным адресом) – 2 байта;
- список номеров сегментов ОЗУ, в которых находятся плоскости области, при этом количество сегментов соответствует размерности области – 1..4 байта.

Программа пользователя должна сохранять параметры ТОО для каждой из задействованных ею областей с целью последующего освобождения памяти по окончании своей работы.

Таблица описания системы

Для выполнения задач, возлагаемых на систему распределения ОЗУ, она должна вести учет наличия свободной памяти. В самом начале работы системы, когда память еще не задействовалась, в распоряжении системы находится исходный массив сегментов, указанный в момент ее инициализации. При этом необязательно, чтобы в динамическую память были включены все сегменты, имеющиеся в компьютере: например, в нее можно не включать экранные сегменты и сегменты рабочей страницы ОЗУ. Для указания системе допустимых к использованию сегментов используется СПИСОК НОМЕРОВ СЕГМЕНТОВ (СНС). Номера сегментов в этом списке не должны повторяться. Под длиной СНС будем понимать количество сегментов, выделяемых под динамическую память (для хранения номера сегмента используется 1 байт).

По мере выделения и освобождения ОЗУ в процессе выполнения программ объем свободной динамической памяти будет изменяться, поэтому система ведет специальный служебный СПИСОК СВОБОДНЫХ ОБЛАСТЕЙ (ССО). Каждый элемент списка свободных областей называется ОПИСАТЕЛЕМ СВОБОДНОЙ ОБЛАСТИ и содержит 5 байт:

- длина свободной области (1..4000H);
- сегментный адрес начала свободной области (0..3FFFH);
- номер сегмента, в котором находится свободная область.

Под текущей длиной ССО будем понимать текущее количество его элементов (описателей), а под максимальной длиной – их максимально допустимое количество, указанное при инициализации (соответствующий максимальный объем ССО в байтах будет в 5 раз больше этой величины).

При инициализации системы в ССО будет включено столько описателей, сколько сегментов было задано в СНС (по одному описателю на каждый сегмент); начальные адреса всех свободных областей будут равны 0, а длины – 4000H (16 Кбайт).

Итак, состояние системы распределения сегментной динамической памяти характеризуется двумя списками: СНС и ССО, параметры которых постоянно хранятся в так называемой ТАБЛИЦЕ ОПИСАНИЯ СИСТЕМЫ (ТОС), содержащей 7 байт:

- текущая длина ССО – 1 байт;
- максимальная длина ССО – 1 байт;
- адрес размещения ССО – 2 байта;
- длина СНС – 1 байт;
- адрес размещения СНС – 2 байта.

Заметим, что адреса размещения списков СНС и ССО указываются в пределах рабочей страницы ОЗУ, при этом запрещается использовать адреса в области окна ROM2 – 2000H..3FFFH.

Инициализация системы

Перед инициализацией системы распределения памяти необходимо выполнить следующие действия:

- создать СНС, разместив его в рабочей странице ОЗУ;
- зарезервировать память под ССО, задавшись его максимальной длиной (количеством элементов) и учитывая, что один элемент ССО занимает 5 байт;
- заполнить все байты ТОС, кроме первого.

При инициализации первый байт ТОС заполняется системой автоматически исходя из длины СНС.

Вызов функции распределения памяти (ВЕКТОР 0F82DH)

Набор базовых функций низкого уровня системы распределения памяти находится в ROM2 V2.00 и доступен через вектор ROM-BIOS с адресом 0F82DH, при этом номер вызываемой функции определяется значением регистра (A).

Операционная система PRODOS использует указанные функции низкого уровня (за счет чего сокращается ее объем) и имеет свои операции более высокого уровня, доступные через вызов расширенной BDOS и позволяющие обращаться программам — пользователя к уже инициализированной системной динамической памяти. Здесь будут рассмотрены только функции низкого уровня, доступные через вектор 0F82DH (слева в скобках указан номер функции, помещаемый перед вызовом в регистр A):

(0) Инициализация динамической памяти (INIMEM)

На входе: DE – адрес ТОС с заполненными данными, кроме первого байта; должен быть создан СНС и зарезервирована память под ССО;

На выходе: A – код завершения:

- 0 – ССО инициализирован успешно;
- 2 – переполнение ССО;
- 3 – повторный сегмент в СНС;
- 4 – недопустимый номер сегмента–в СНС;
- 5 – недопустимая длина СНС.

(1) Проверка занятости области памяти (CHKMEM)

На входе: DE – адрес ТОС;

HL – адрес ТОО с параметрами области;

На выходе: A – код завершения

- 0 – указанная область полностью свободна;
- 1 – указанная область полностью занята или вообще отсутствует в системе;
- 2 – указанная область занята частично;
- 3 – недопустимое значение размерности области в T00 (вне диапазона 1..4).

Принудительное выделение области памяти (MRKMEM)

(установка признака использования области)

На входе: DE – адрес T0C;

HL – адрес T00 с параметрами выделяемой области;

На выходе: A – код завершения:

- 0 – указанная область выделена успешно и переведена в разряд занятых;
- 1 – указанная область уже полностью занята или вообще отсутствует в системе;
- 2 – указанная область уже занята частично;
- 3 – недопустимое значение размерности области в T00 (вне диапазона 1..4);
- 255 – фатальная ошибка памяти (нарушение целостности системы).

(3) Запрос области памяти требуемой длины (GETMEM)

На входе: DE – адрес T0C;

HL – адрес T00 с первыми тремя заполненными байтами:

(HL+0) – требуемая размерность выделяемой области 1..4;

(HE+1), (HE-2) – требуемая длина выделяемой области 1..4000H;

На выходе: A – код завершения:

- 0 – требуемая область выделена успешно;
- 1 – нет сегментной памяти для выделения требуемой области;
- 2 – переполнение CCO;
- 3 – недопустимое значение размерности области (вне диапазона 1..4);
- 255 – фатальная ошибка памяти (нарушение целостности системы);

Для A=0:

HL указывает на T00 со всеми параметрами выделенной области.

(4) Освобождение указанной области памяти (FREEMEM)

На входе: DE – адрес T0C;

HL – адрес T00 с параметрами области;

На выходе: A – код завершения:

- 0 – указанная область освобождена успешно;
- 1 – указанная область не была выделена (полностью или частично свободна);
- 2 – переполнение CCO;
- 3 – недопустимое значение размерности области в T00 (вне диапазона 1..4);
- 4 – недопустимый номер сегмента в T00 (отсутствует в СНС);
- 255 – фатальная ошибка памяти (нарушение целостности системы).

ДРАЙВЕР МЫШИ

Для поддержки мыши в BIOS имеются два вектора: MOUSE (0F84EH) и MSCTL (0F851H). После вызова указанных векторов все регистры сохраняются, за исключением AF.

Вектор MOUSE предназначен для непосредственного опроса порта мыши (1EH) и возвращает в регистре A его состояние:

- D7 – резерв (1)
- D6 – резерв (1)
- D5 – /правая кнопка
- D4 – /левая кнопка
- D3 – /вверх
- D2 – /вниз
- D1 – /вправо
- D0 – /влево

Активному состоянию какого-либо разряда соответствует уровень лог.0 (исходное состояние всех разрядов – единичное).

Вектор MSCTL выполняет несколько функций по управлению мышью. Номер функции задается в регистре A перед вызовом подпрограммы 0F851H:

A=0 – Перемещение указателя мыши в окне до нажатия кнопки

- На входе: BC – исходная позиция Y мыши;
DE – исходная позиция X мыши;
- На выходе: "CY" – признак нахождения исходных значений X,Y вне окна, или признак переполнения буфера сохранения информации под указателем;
- иначе: BC – выходная позиция Y мыши;
DE – выходная позиция X мыши;
A – код нажатой кнопки:
 - 80H – нажата левая кнопка;
 - 40H – нажата правая кнопка.

При вызове функции A=0 на экране появляется указатель мыши, который перемещается до тех пор, пока не нажата хотя бы одна из ее кнопок. Указатель мыши может перемещаться также с помощью клавиш управления курсором; клавиша BK соответствует нажатию левой кнопки мыши, клавиша AP2 – правой кнопки.

Цвет указателя мыши зависит от установленного режима отображения мыши (смотри ниже) и от текущих цветов фона и переднего плана (смотри описание драйвера "TV-PRO").

Замечание

Данная функция A=0 использует вектор MOUSE (0F84EH) для опроса состояния мыши. При написании собственной подпрограммы обработки вектора MOUSE (например, в случае использования нестандартной мыши) следует учитывать, что адрес размещения ее кодов должен быть больше 1FFFH, так как она вызывается из ROM1.

A=1 – Установка режима отображения указателя мыши

- На входе:
 - C – код режима;
 - D7 – признак перемещения по всему экрану (при D7=0 – в пределах текущего окна);
 - D3 – признак сохранения информации экрана под указателем мыши,

- D2 – признак инверсии (цвета фона и переднего плана меняются местами,
D1, D0 – режим наложения указателя:
00 – с полной заменой фона и изображения;
01 – наложение по ИЛИ (цветом переднего плана закрашиваются те точки экрана, которым в шаблоне указателя мыши соответствуют единицы),
10 – наложение по И (цветом фона закрашиваются те точки экрана, которым в шаблоне указателя мыши соответствуют нули)
11 – наложение по исключающему ИЛИ (независимо от текущих цветов фона и переднего плана инвертируется цвет точек экрана, которым в шаблоне указателя мыши соответствуют единицы):

Рекомендуемые режимы отображения указателя мыши: по ИЛИ – 09H (89H), или по XOR – 03H (83H).

A=2 – Получение режима отображения указателя мыши

На выходе:

C – код режима;

A=3 – Установка размеров указателя мыши

На входе:

E – высота указателя мыши в пикселях;

D – ширина указателя мыши в пикселях;

C – шаг перемещения мыши по Y;

B – шаг перемещения мыши по X;

L – смещение шаблона указателя мыши в пределах левого столбца байтов (0..7);

A=4 – Получение размеров указателя мыши

На выходе:

E – высота указателя мыши в пикселях;

D – ширина указателя мыши в пикселях;

C – шаг перемещения мыши по Y;

B – шаг перемещения мыши по X;

L – смещение шаблона указателя мыши в пределах левого столбца байтов (0..7);

A=5 – Установка адреса размещения шаблона мыши

На входе:

C – номер сегмента с шаблоном указателя;

HL – начальный адрес размещения шаблона в сегменте (в пределах сегмента или в пределах страницы ОЗУ);

A=6 – Получение адреса размещения шаблона мыши

На выходе:

C – номер сегмента с шаблоном указателя;

HL – начальный адрес размещения шаблона в сегменте (в пределах сегмента или в пределах страницы ОЗУ);

A=7 – Установка адреса буфера сохранения информации под указателем мыши

На входе:

CBED – номера сегментов для сохранения информации экрана под указателем мыши (C – для монохромного и псевдоцветного режимов доступного экрана, CB – для 4-цветного режима и 16-цветного режима с групповым

кодированием, CBE – для 8-цветного режима, CBED – для 16-цветного режима);

HL – начальный адрес буфера в сегментах (во всех сегментах одинаковый);

A=8 – Получение адреса буфера сохранения информации под указателем мыши

На выходе:

CBED – номера сегментов для сохранения информации экрана под указателем мыши (C – для монохромного и псевдоцветного режимов доступного экрана, CB – для 4-цветного режима и 16-цветного режима с групповым кодированием, CBE – для 8-цветного режима, CBED – для 16-цветного режима);

HL – начальный адрес буфера в сегментах (во всех сегментах одинаковый);

A=9 – Получение длины буфера, необходимой для сохранения информации под указателем мыши

На выходе:

DE – длина буфера, необходимая для сохранения информации под указателем мыши (одинакова для всех сегментов буфера);

A=10 – Поиск выбранного объекта по базе

На входе:

BC – координата Y объекта;

DE – координата X объекта;

HL – адрес начала таблицы с описанием базы;

На выходе: "CY" – объект не найден,
иначе: A – порядковый номер объекта (0, 1...);
HL – адрес начала описания объекта в базе.

Таблица с описанием базы объектов имеет следующую структуру:

```
TAB: DB n      ; Число объектов в базе
      ;
      DW x1     ; X, Y левого верхнего угла 0-го объекта
      DB y1
      DW dx1    ; Размеры 0-го объекта
      DB dy1
      ;
      ...
      DW xp     ; X, Y левого верхнего угла (n-1)-го объекта
      DB yp
      DW dxn    ; Размеры (n-1)-го объекта
      DB dyn
EOFTAB:      ; Конец таблицы
```

Первый байт таблицы содержит количество объектов в базе. Далее следует соответствующее число описателей объектов, каждый из которых занимает 6 байт.

Заметим, что количество объектов в базе: $n = [EOFTAB - TAB - 1] / 6$

Приложения

Перечень элементов

Перечень элементов ПК "Орион-ПРО" V3.20

Наименование	Колич. v3.20	Аналог	Позиция
1533АП6	2	74ALS245	D68,D79
1533ЛА1	1	74ALS12	D66
1533ЛА2	3	74ALS30	D85,D89,D100
1533ЛА3	6	74ALS00	D1,D10,D11,D14,D38,D86
1533ЛА4	2	74ALS10	D103,D119
1533ЛА8	1	74ALS01	D32
1533ЛЕ1	3	74ALS02	D102,D120,D121
1533ЛЕ4	1	74ALS27	D*
1533ЛИ1	7	74ALS08	D15-D18,D104,D112,D118
1533ЛЛ1	7	74ALS32	D19-D25
1533ЛН1	4	74ALS04	D101,D195-D107
1533ЛН2	3	74ALS05	D37,D44,D45
1533ИЕ5	5	74ALS93	D2,D3,D6,D7,D84
1533ИЕ7	3	74ALS193	D31,D46, D122
1533КП11	4	74ALS257	D36,D39,D42,D43
1533КП12	9	74ALS253	D4,D5,D8,D9,D12,D13,D88,D10, D109
1533ТМ2	10	74ALS74	D26-D29,D41,D47,D48,D83, D115, D*
531ТМ2 (Возможно придется установить вместо 1533ТМ2)	0/1	74S74	D47
1533ТМ8	4	74ALS175	D87,D110,D113,D117
1533ТМ9	3	74ALS174	D111,D114,D116

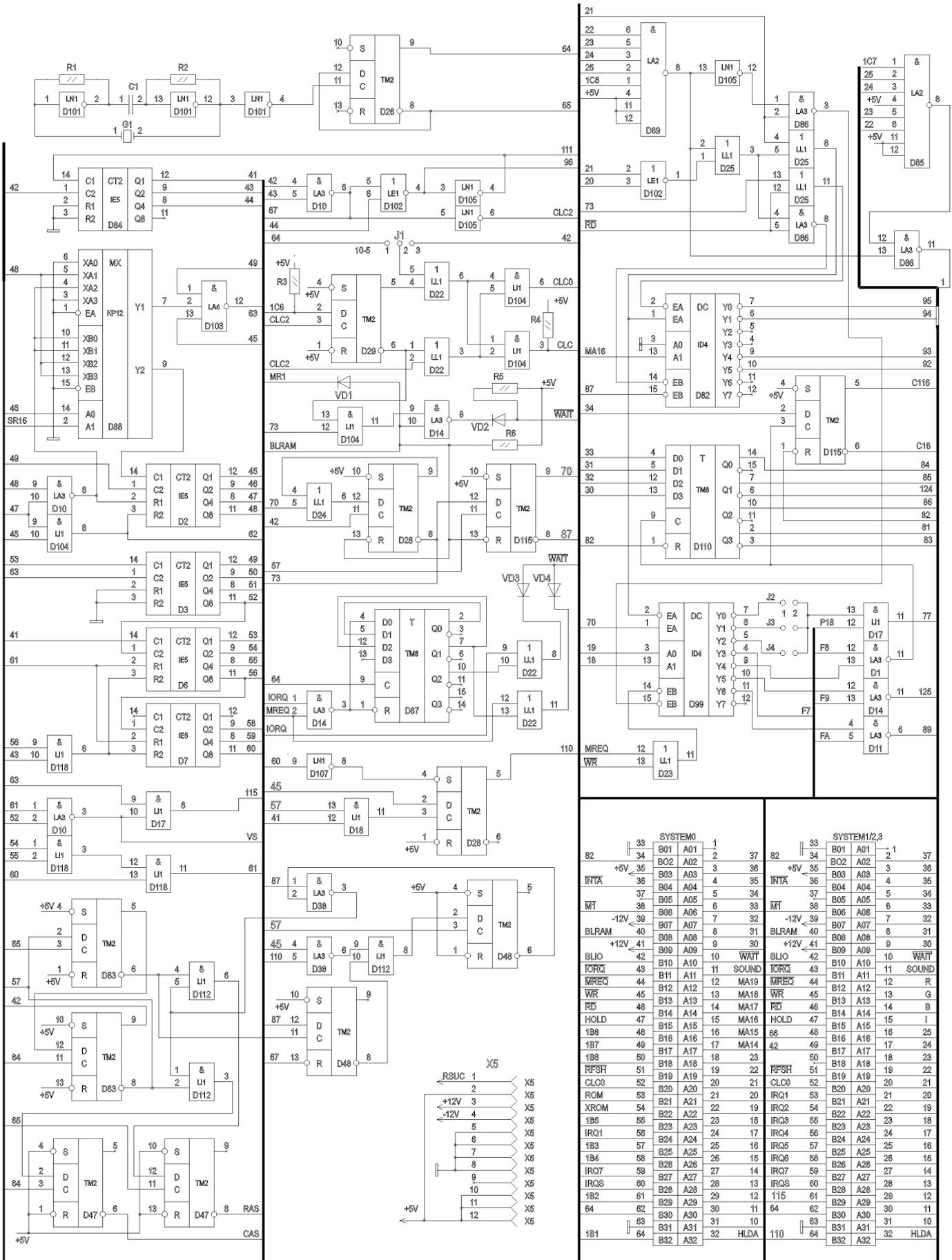
1533ИД4	2	74ALS155	D82, D99
1533ИД7	2	74ALS138	D76, D93
1533ИР10	4	74ALS166	D49, D50, D51, D58
555ИР16	1	74ALS295	D34
1533ИР22	9	74ALS373	D40, D75, D90-D92, D94-D98
1533ИР23	1	74ALS374	D35
155ИВ1/555ИВ1	1	74LS148	D56
КР580ВВ55А	4	8255	D55, D69, D80, D81
Z80H	1	--	D77
КР1818ВГ93	1	WDC1793	D33
27C64	1	--	D67
27C256	1	--	D78
КР565РУ7И	16	41256	D52-D54, D57, D59-D65, D70-D74
SW-8 dip перекл.	1	--	D38
КР1006ВИ1	1	555	D30
Разъем клав. PS/2	1	?	X2
Разъем FDD 34-pin	1	--	X4
Разъем MOLEX 8980 (мама) или Molex Mini-Fit Jr (мама)	1		X5
Разъем IDE 40-Pin	1	--	X3
Разъем DIN25 Centronics	1		X1
Разъем DIN 41612 64pin 2x32 female	4	--	SYSTEM0-3
КТ315Б	2	2SC1959	VT1, VT3
КТ972	1	BD875	VT2
КД522	12	1N4148	VD1-VD4, VD6, VD8-VD12, VD14
NR1-4-9(3.3-10K)	3		NR1-NR3

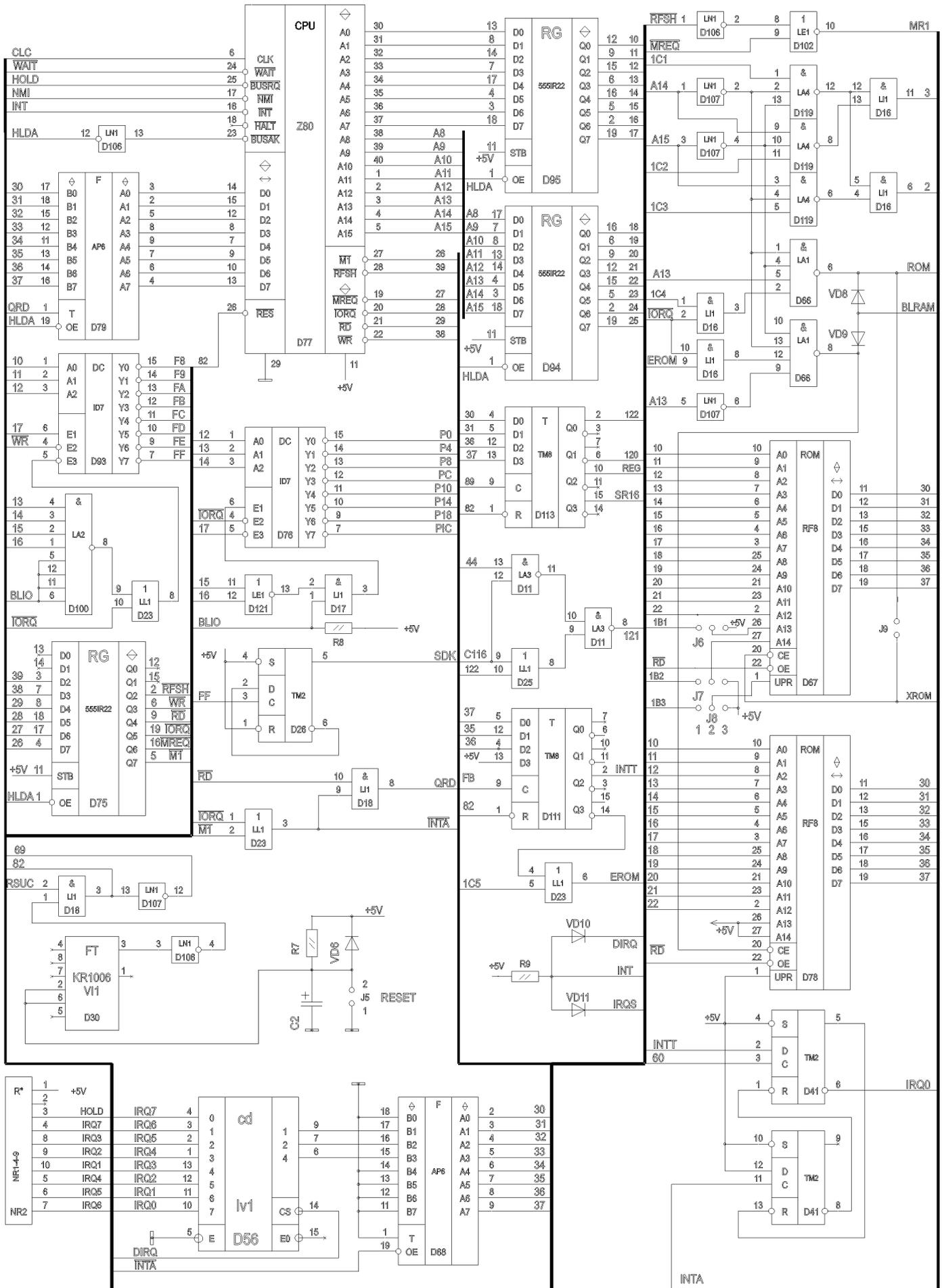
ДПМ 0.4-30	1		L1
КС168А	1	1N4736A	VD13
Кварц 20.0МГц	1	HC-49U	Q1
Кварц 8.0МГц	1	HC-49U	Q2
Конденсатор кер. 68пФ	1		C1
Конденсатор 1мкФ	1		C2
Конденсатор кер. 200пФ	1		C3
Конденсатор кер. 20пФ	2		C6, C9
Конденсатор кер. многосл. 0,1мкФ — 0,68мкФ	36		C8, C11, C13, C15-C47
Конденсатор эл. 10мкФ	5		C4, C5, C7, C10, C12
Резистор 0,125Вт 75Ω	1	1/8W	R30
Резистор 0,125Вт 150Ω	2		R31, R33
Резистор 0,125Вт 510Ω	6		R1, R2, R16, R17, R26, R28, R34
Резистор 0,125Вт 1кΩ	12		R4-R6, R9-R15, R25, R35
Резистор 0,125Вт 2кΩ	16		R3, R8, R18-R24, R32, R36, R38-R42
Резистор 0,125Вт 3.9кΩ	1		R37
Резистор 0,125Вт 5.1кΩ	1		R27
Резистор 0,125Вт 15кΩ	2		R7, R29

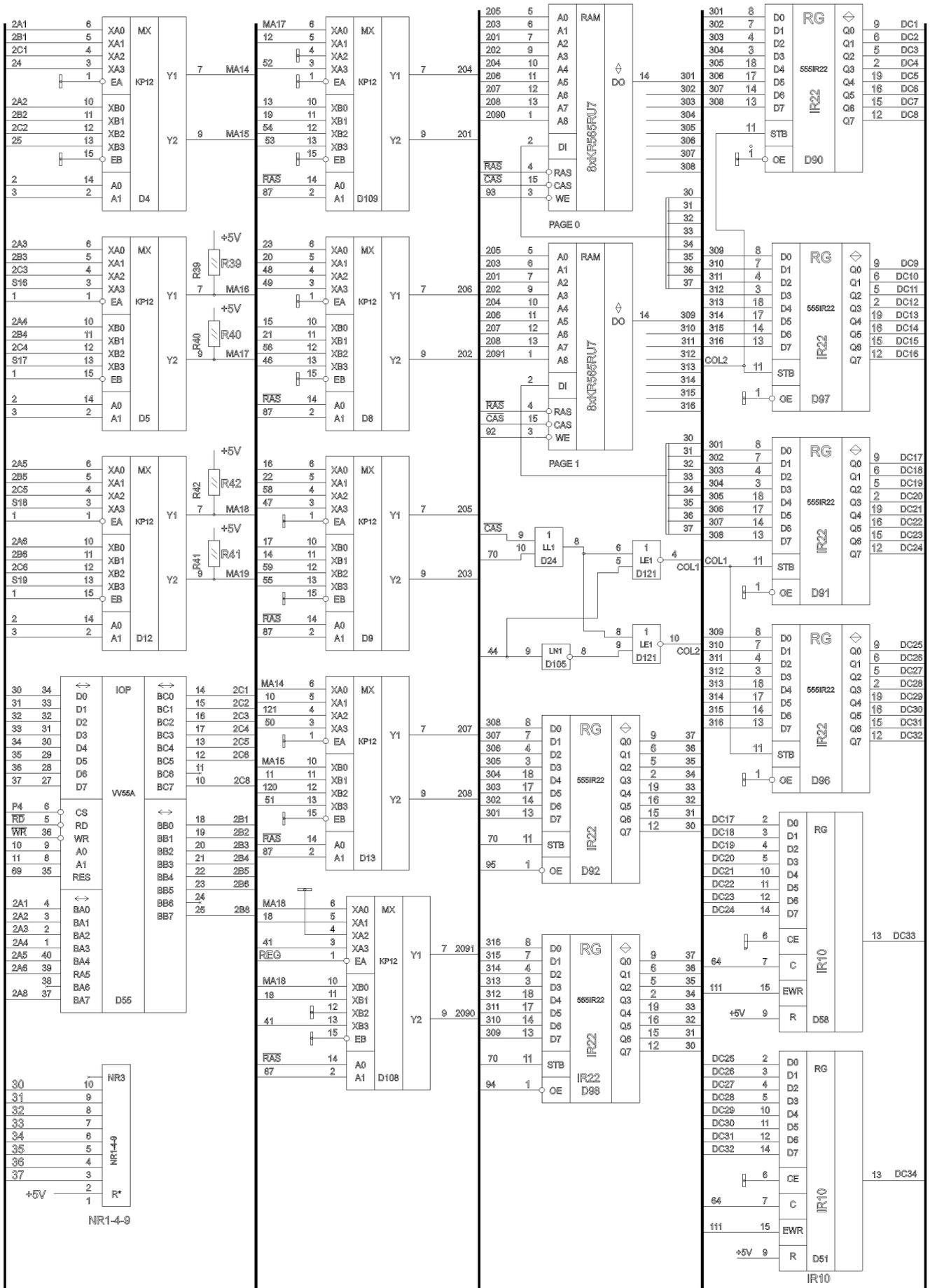
Принципиальные схемы

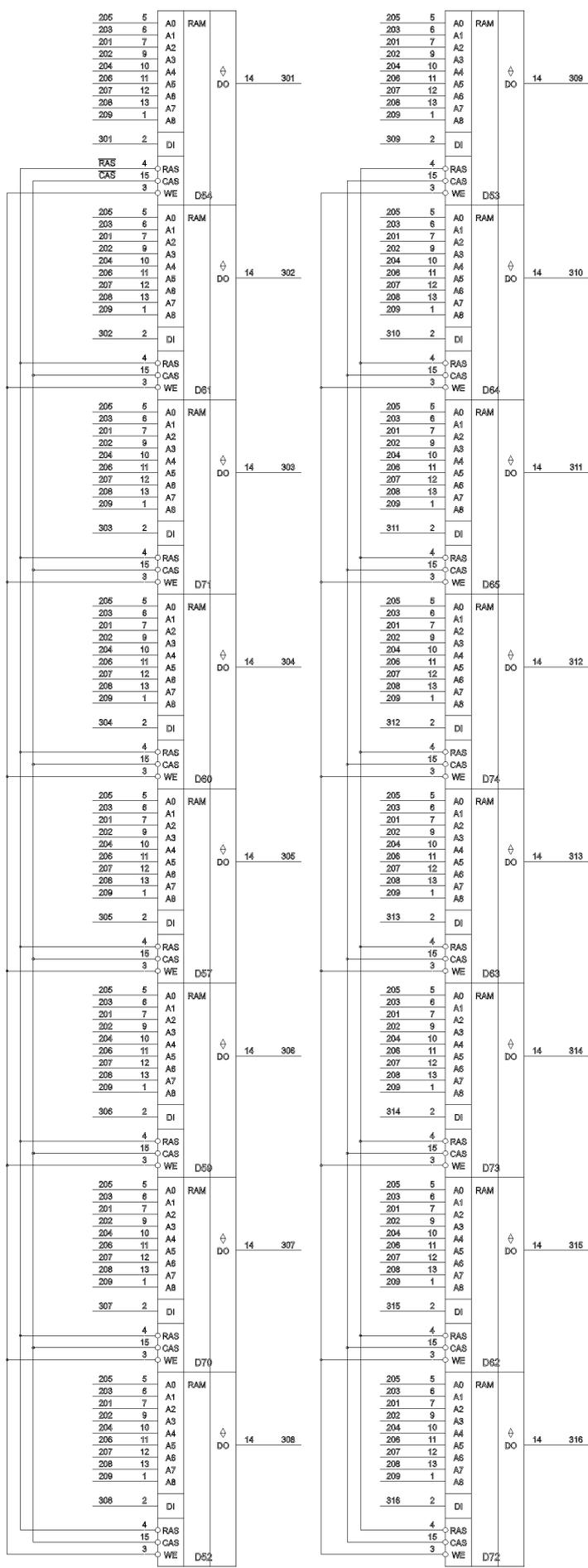
ORION-POWER VERSION 3.10 (C) 1993,1996,1998 ORIONSOFT

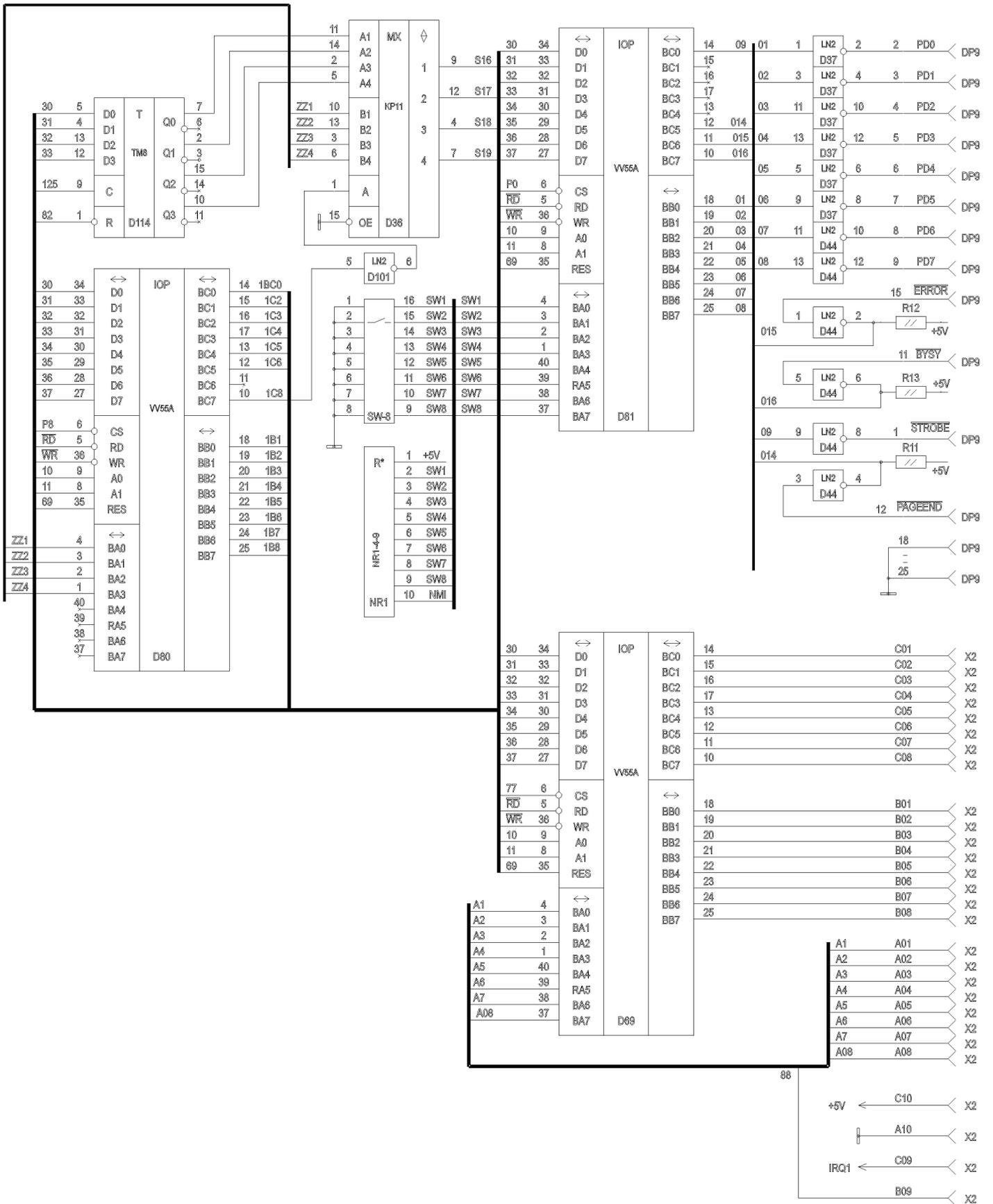
Лист 1

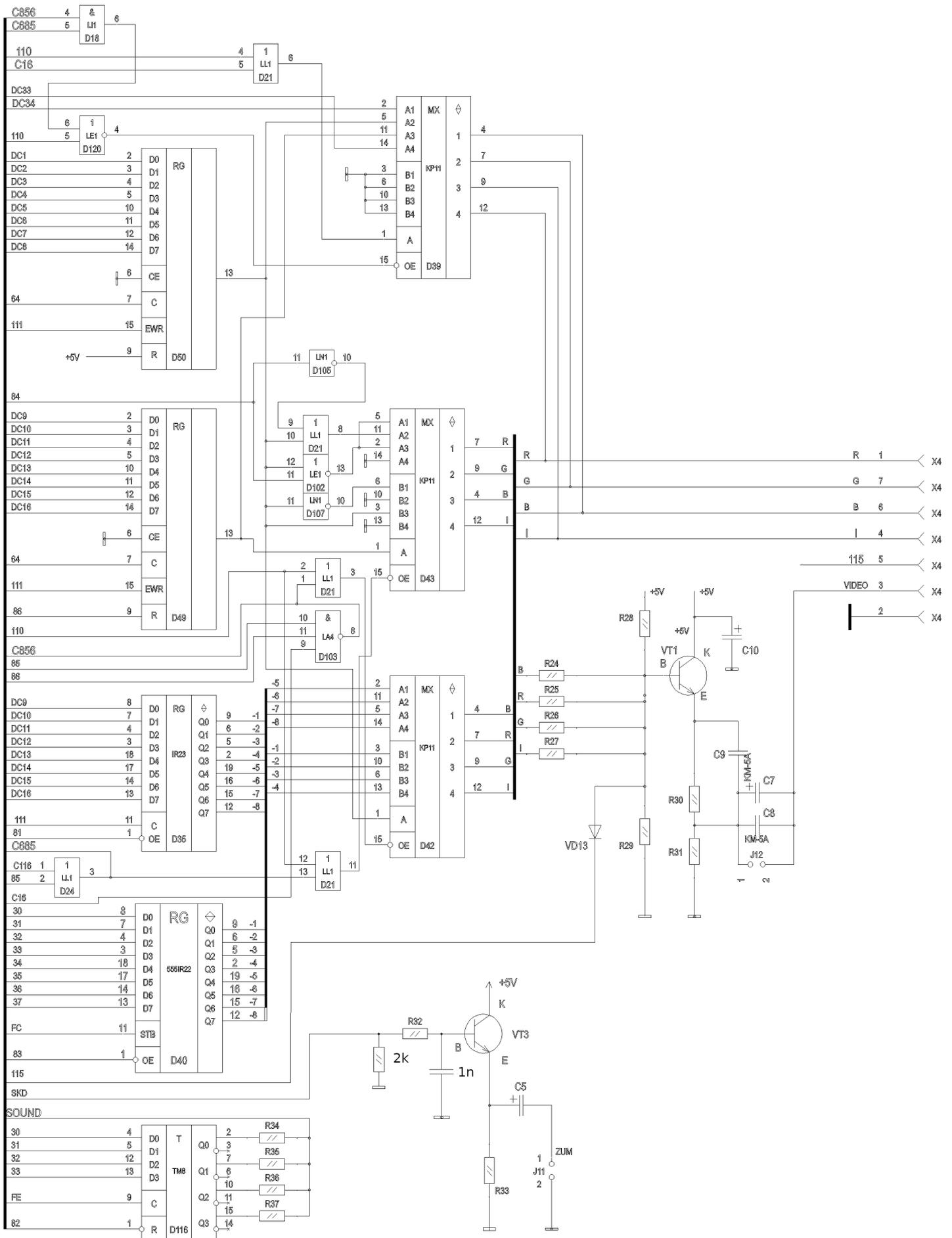


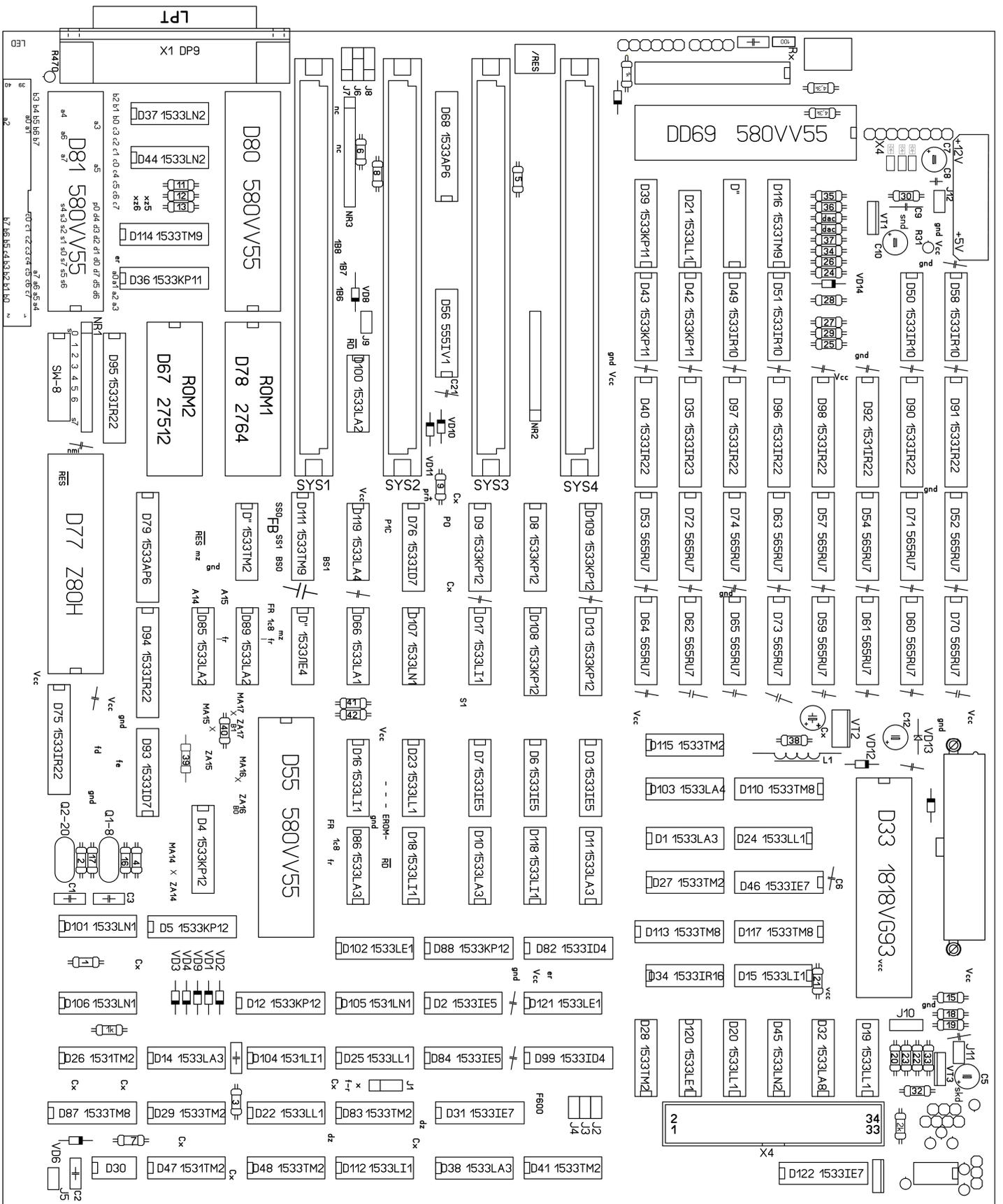












Оглавление

От авторов.....	3
Итак, новый компьютер.....	5
РАДИОЛЮБИТЕЛЬСКИЙ КОМПЬЮТЕР «Орион-ПРО».....	8
Системная плата.....	9
Принципиальная схема.....	12
Рекомендации по сборке и наладке.....	15
РЕЖИМЫ РАБОТЫ КОМПЬЮТЕРА "ORION-PRO".....	17
Режим "Monitor".....	17
Режим "ORION-PRO".....	21
Режим "ORION-128".....	22
Режим "TEST".....	24
ОРГАНИЗАЦИЯ ОПЕРАТИВНОЙ ПАМЯТИ.....	25
ОРГАНИЗАЦИЯ ЭКРАННОЙ ПАМЯТИ.....	27
ОРГАНИЗАЦИЯ ПОСТОЯННОЙ ПАМЯТИ.....	29
ПОРТЫ ВВОДА-ВЫВОДА.....	30
Порты ПК "Орион-ПРО:.....	30
BIOS компьютера "ORION-PRO".....	33
РАСПРЕДЕЛЕНИЕ НЕПЕРЕКЛЮЧАЕМОЙ ПАМЯТИ.....	34
ТАБЛИЦА ВЕКТОРОВ BIOS.....	34
ОСОБЕННОСТИ ДРАЙВЕРА КЛАВИАТУРЫ.....	39
ДРАЙВЕР ДИСПЛЕЯ.....	45
Управление экраном (ВЕКТОР 0F80CH).....	46
Цвета доступного экрана.....	49
Цвета (палитры) видимого экрана.....	49
Управление выводом символа и курсора (ВЕКТОР 0F80FH).....	53
Вывод графики (ВЕКТОР 0F82AH).....	61
СИСТЕМА ДИНАМИЧЕСКОГО РАСПРЕДЕЛЕНИЯ ОЗУ.....	67
Таблица описания области памяти.....	68
Таблица описания системы.....	68
Инициализация системы.....	69
Вызов функции распределения памяти (ВЕКТОР 0F82DH).....	69
ДРАЙВЕР МЫШИ.....	71
Приложения.....	74
Перечень элементов.....	74
Принципиальные схемы.....	77

УВАЖАЕМЫЙ ЧИТАТЕЛЬ

Ваши замечания о содержании книги, ее оформлении и обнаруженных ошибках прошу присылать по адресу:

142440, Московская обл.,
п. Обухово, а/я13
Сугоняко В.П.

Владислав Петрович Сугоняко

Радиолобительский

компьютер

"Орион-ПРО"

Издано "ОРИОНСОФТ", 1998г.

Редакция от 01.01.1998г.

Технический редактор и корректор В.О. Пушков

Часть разделов этой книги написана по материалам, подготовленных В.О. Пушковым для руководства по ROM-BIOS