**Вектор Турбо +**

Аннотация

Вектор Турбо + разработан в конце 90 годов в развитие бытового компьютера Вектор‑06Ц, при этом была сохранена практически полная обратная совместимость.

Основные отличительные особенности Вектор Турбо +:

– процессор Z80

– тактовая частот 12 МГц с возможностью программного понижения до 6 и 3 МГц;

– ОЗУ – 1 Мбайт, адресуется блоками по 64 кбайт;

– ПЗУ – 32 кбайт;

– статическое ОЗУ 32 кбайт;

– служебное статическое ОЗУ 16 кбайт, адресуемое через порты;

– расширенные способы адресации памяти;

– дополнительные графические режимы (выделены цветом):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X | Y | Кол-во цветов | Палитра | Экранная память, кбайт | Примечание |
| 256 | 256 | 16 | 256 | 32 |  |
| 512 | 256 | 4 | 256 | 32 |  |
| 256 | 256 | 256 | 32768 | 32+3 |  |
| 512 | 256 | 256 | 32768 | 64+3 |  |
| 512 | 512 | 256 | 32768 | 128+3 | Чересстрочный режим |
| 1024 | 256 | 64 | 32768 | 64+3 |  |
| 1024 | 512 | 64 | 32768 | 128+3 | Чересстрочный режим |

– интегрированы контроллеры дисковода и жёсткого диска, часы реального времени, а также музыкальный сопроцессор;

– в состав клавиатуры введена клавиша Fn, которая активирует множество сервисных функций используя механизм немаскируемых прерываний (NMI).

Карта распределения адресного пространства памяти

Динамическое ОЗУ

Мегабайт ОЗУ разделён на страницы по 64кбайт, доступ к которым выполняется через порты 40 и 41. При этом любая страница может использоваться для отображения (порт 43).

Базовые 64к ОЗУ расположены в старших разрядах, при попытке доступа к этой странице памяти с помощью портов 40 и 41, будет активировано ПЗУ или статическое ОЗУ (см.таблицу).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Диапазон адресов | Порты 40, 41 (биты 0-3) | Порт 43 (биты 0-4) | Комментарии | | | | | |
| 00000 – 07FFF | 08 | 10 |  | | | | | Квазидиск 800к |
| 08000 – 0FFFF | 11 |  | | | | |
| 10000 – 17FFF | 09 | 12 |  | | | | |
| 18000 – 1FFFF | 13 |  | | | | |
| 20000 – 27FFF | 0A | 14 |  | | | | |
| 28000 – 2FFFF | 15 |  | | | | |
| 30000 – 37FFF | 0B | 16 |  | | | | |
| 38000 – 3FFFF | 17 |  | | | | |
| 40000 – 47FFF | 0C | 18 | 0 | Порт 10  (биты 0,1 или 2,3) | | Квазидиск 238к (для совместимости) | |
| 48000 – 4FFFF | 19 |
| 50000 – 57FFF | 0D | 1A | 1 |
| 58000 – 5FFFF | 1B |
| 60000 – 67FFF | 0E | 1C | 2 |
| 68000 – 6FFFF | 1D |
| 70000 – 77FFF | 0F | 1E | 3 |
| 78000 – 7FFFF | 1F |
| 80000 – 87FFF | 00 | 00 |  | | | | |
| 88000 – 8FFFF | 01 |  | | | | |
| 90000 – 97FFF | 01 | 02 |  | | | | |
| 98000 – 9FFFF | 03 |  | | | | |
| A0000 – A7FFF | 02 | 04 |  | | | | |
| A8000 – AFFFF | 05 |  | | | | |
| B0000 – B7FFF | 03 | 06 |  | | | | |
| B8000 – BFFFF | 07 |  | | | | |
| C0000 – C7FFF | 04 | 08 | Квазидиск 800к | | | | | |
| C8000 – CFFFF | 09 | Для расширенной графики | | | | | |
| D0000 – D7FFF | 05 | 0A | Квазидиск 800к | | | | | |
| D8000 – DFFFF | 0B | Для расширенной графики | | | | | |
| E0000 – E7FFF | 06 | 0C | Экран начального загрузчика | | | | | |
| E8000 – EFFFF | 0D | Для расширенной графики | | | | | |
| F0000 – F7FFF | 07 | 0E | ПЗУ 32к | | Основные 64к | | | |
| F8000 – FFFFF | 0F | СОЗУ 32к | |  | | Основной экран | |

ПЗУ

ПЗУ имеет размер 32 кбайт в адресном пространстве 0000H-7FFFH. Доступ к ПЗУ возможен двумя способами:

– как к теневой странице памяти (см. таблицу выше). В этом режиме выполнение программы в ПЗУ рассматривается системой как в ОЗУ;

– в режиме начального загрузчика. В этом режиме в младших 32 кбайт на чтение адресуется ПЗУ вне зависимости от состояния портов 40H и 42H, но запись в этом адресном пространстве будет производиться в ОЗУ в соответствии с содержимым портов. При этом операции как записи, так и чтения из стека будут обращаться к странице ОЗУ, заданной портом 41H, или, если в 7 бите порта записан 0, к базовой странице ОЗУ. Это позволяет в режиме загрузчика иметь полный доступ ко всей памяти.

Вход в режим начального загрузчика может быть выполнен:

– с клавиатуры нажатием Ввод+Блк, будет выполнен сброс процессора с переходом на 0000H;

– с клавиатуры нажатием Fn+Блк, будет выполнено немаскируемое прерывание (если установлен бит 5 порта 45H) с переходом на 0066H;

– при выполнении маскируемого прерывания от кадрового синхроимпульса (если сброшен бит 6 порта 45H) с переходом на 0018H.

Примечание – При выполнении прерываний в ПЗУ, адрес возврата будет записан в вершину стека в ОЗУ и только после этого активировано ПЗУ.

Выход из режима начального загрузчика может быть выполнен:

– с клавиатуры нажатием Сбр+Блк, будет выполнен сброс процессора с переходом на 0000H;

– записью 0, а затем 1 в бит 1 порта 01H, будет выполнен сброс процессора с переходом на 0000H;

– выполнением команд RETI или RETN, будет выполнен выход из режима загрузчика, после чего прочитан адрес возврата из вершины стека в ОЗУ. Кроме корректного возврата из прерываний этот метод позволяет выходить из режима загрузчика с передачей управления на произвольный адрес в ОЗУ.

Примечание – При выполнении прерываний в ПЗУ активируется возможность использования автосброса установкой бита 1 порта 01H.

ПЗУ содержит:

– программу тестирования памяти. Программа может корректно работать даже при полном отсутствии микросхем динамического ОЗУ, визуализация дефектных микросхем при этом осуществляется цветом атрибутов;

– программу начального загрузчика, выполняющую загрузку с магнитной ленты, дисковода, жёсткого диска, с квазидиска и т.п.;

– подпрограммы обработки прерывания INT (адреса 0018H, 0038H);

– программу настройки системы и полуаппаратный отладчик программ (с использованием прерывания NMI). Параметры настройки системы сохраняются в энергонезависимой памяти часов реального времени;

– базовую операционную систему.

Статическое ОЗУ

Статическое ОЗУ (СОЗУ) расположено в адресном пространстве 8000H-FFFFH. Доступ к СОЗУ осуществляется как к теневой странице памяти (см. таблицу выше).

Распределение адресного пространства СОЗУ:

– 8000H-80FFH – последние значения, записанные в порты 00H-FFH;

– 8100H-83FFH – зарезервировано для служебных программ;

– 8400H-87FFH – таблица соответствия математического цвета физическому;

– 8800H-8FFFH – таблица атрибутов экрана;

– 9000H-FFFFH – произвольные значения для пользовательских программ.

**Механизм формирования цвета**

Вектор Турбо + в отличие от Вектор-06ц может формировать код цвета используя кроме данных с экранных плоскостей ещё и данные атрибутов. Каждый атрибут определяет цвет в прямоугольнике 8х8 или 8х16 точек в режимах с 256 или 512 строк, соответственно.

Значение атрибута содержится в его младшем полубайте, таким образом в рамках каждого атрибута может быть 16 значений и всего на экране возможно одновременное отображение 256 цветов. Старшие половинки байт в таблице атрибутов могут использоваться произвольным образом, например для хранения кодов инверсных цветов при организации мигающего знакоместа.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Математический цвет | | | | | | | | | | | |
| X | X | X | X | A3 | A2 | A1 | A0 | P3 | P2 | P1 | P0 |
| A0 – A3 – биты атрибута  P0 – P3 – биты экранных плоскостей | | | | | | | | | | | |

Привязка адресов атрибутов к экранным адресам на примере нулевой экранной плоскости базовой страницы для максимального разрешения экрана:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8800 | 8801 |  | 881E | 881F | 8820 | 8821 |  | 883E | 883F |
| DE000-  DE007, | DE100-  DE107, |  | DFE00-  DFE07, | DFF00-  DFF07, | CE000-  CE007, | CE100-  CE107 |  | CFE00-  CFE07 | CFF00-  CFF07 |
| FE000-  FE007 | FE100-  FE107 |  | FFE00-  FFE07 | FFF00-  FFF07 | EE000-  EE007 | EE100-  EE107 |  | EFE00-  EFE07 | EFF00-  EFF07 |
| 8840 | 8841 |  | 885E | 885F | 0860 | 8861 |  | 887E | 887F |
| DE008-  DE00F, | DE108-  DE10F, |  | DFE08-  DFE0F, | DFF08-  DFF0F, | CE008-  CE00F, | CE108-  CE10F, |  | CFE08-  CFE0F, | CFF08-  CFF0F, |
| FE008-  FE00F | FE108-  FE10F |  | FFE08-  FFE0F | FFF08-  FFF0F | EE008-  EE00F | EE108-  EE10F |  | EFE08-  EFE0F | EFF08-  EFF0F |
|  |  |  |  |  |  |  |  |  |  |
| 8F80 | 8F81 |  | 8F9E | 8F9F | 8FA0 | 8FA1 |  | 8FBE | 8FBF |
| DE0F0-  DE0F7, | DE1F0-  DE1F7, |  | DFEF0-  DFEF7, | DFFF0-  DFFF7, | CE0F0-  CE0F7, | CE1F0-  CE1F7, |  | CFEF0-  CFEF7, | CFFF0-  CFFF7, |
| FE0F0-  FE0F7 | FE1F0-  FE1F7 |  | FFEF0-  FFEF7 | FFFF0-  FFFF7 | EE0F0-  EE0F7 | EE1F0-  EE1F7 |  | EFEF0-  EFEF7 | EFFF0-  EFFF7 |
| 8FC0 | 8FC1 |  | 8FDE | 8FDF | 8FE0 | 8FE1 |  | 8FFE | 8FFF |
| DE0F8-  DE0FF, | DE1F8-  DE1FF, |  | DFEF8-  DFEFF, | DFFF8-  DFFFF, | CE0F8-  CE0FF, | CE1F8-  CE1FF, |  | CFEF8-  CFEFF, | CFFF8  CFFFF, |
| FE0F8-  FE0FF | FE1F8-  FE1FF |  | FFEF8-  FFEFF | FFFF8-  FFFFF | EE0F8-  EE0FF | EE1F8-  EE1FF |  | EFEF8-  EFEFF | EFFF8  EFFFF |
|  | – адрес атрибута в СОЗУ | | | | | | | | |
|  | – диапазон ячеек отображаемого ОЗУ для нечётных строк (1, 3 … 511) | | | | | | | | |
|  | – диапазон ячеек отображаемого ОЗУ для чётных строк (0, 2 … 510) | | | | | | | | |

**Структура таблицы соответствия математического цвета физическому**

Каждому математическому цвету соответствует 15 бит физического цвета, представленные в двух байтах – в первом байте записываются старшие биты цвета (B4\_B3\_G4\_G3\_G2\_R4\_R3\_R2), а во втором младшие (XX\_B2\_B1\_B0\_G1\_G0\_R1\_R0).

Для обеспечения совместимости структура первого байта соответствует принятому в Вектор‑06Ц.

| Адрес в СОЗУ | Адрес в ОЗУ палитры (математический цвет) | Физический цвет |
| --- | --- | --- |
| 8400, 8420 | 00 | B4\_B3\_G4\_G3\_G2\_R4\_R3\_R2 |
| 8401, 8421 | 00 | XX\_B2\_B1\_B0\_G1\_G0\_R1\_R0 |
| 8402, 8422 | 01 | B4\_B3\_G4\_G3\_G2\_R4\_R3\_R2 |
| 8403, 8423 | 01 | XX\_B2\_B1\_B0\_G1\_G0\_R1\_R0 |
| 841E, 843E | 0F | B4\_B3\_G4\_G3\_G2\_R4\_R3\_R2 |
| 841F, 843F | 0F | XX\_B2\_B1\_B0\_G1\_G0\_R1\_R0 |
| 8440, 8460 | 10 |  |
| 8441, 8461 | 10 |  |
| 8450, 8470 | 18 |  |
| 8480, 84A0 | 20 |  |
| 8490, 84B0 | 28 |  |
| 84C0, 84E0 | 30 |  |
| 84D0, 84F0 | 38 |  |
| 8500, 8520 | 40 |  |
| 8510, 8530 | 48 |  |
| 8540, 8560 | 50 |  |
| 8550, 8570 | 58 |  |
| 8580, 85A0 | 60 |  |
| 8590, 85B0 | 68 |  |
| 85C0, 85E0 | 70 |  |
| 85D0, 85F0 | 78 |  |
| 8600, 8620 | 80 |  |
| 8610, 8630 | 88 |  |
| 8640, 8660 | 90 |  |
| 8650, 8670 | 98 |  |
| 8680, 86A0 | A0 |  |
| 8690, 86B0 | A8 |  |
| 86C0, 86E0 | B0 |  |
| 86D0, 86F0 | B8 |  |
| 8700, 8720 | C0 |  |
| 8710, 8730 | C8 |  |
| 8740, 8760 | D0 |  |
| 8750, 8770 | D8 |  |
| 8780, 87A0 | E0 |  |
| 8790, 87B0 | E8 |  |
| 87C0, 87E0 | F0 |  |
| 87D0, 87F0 | F8 |  |
| 87DE, 87FE | FF | B4\_B3\_G4\_G3\_G2\_R4\_R3\_R2 |
| 87DF, 87FF | FF | XX\_B2\_B1\_B0\_G1\_G0\_R1\_R0 |

Примечание – Физический цвет необходимо записывать в два адреса из-за особенности схемы перебора адресов.

Программирование цветовой палитры возможно выполнять двумя способами:

– записью в порт 0СH, как это было реализовано в Вектор-06Ц. При этом значения атрибутов игнорируются и отображаться будет только 16 цветов.

В этом режиме на всём экране будет использоваться палитра, определённая для атрибута 00H, и если в выходных СОЗУ для неё были записаны цвета с ненулевыми значениями младших битов, то они будут влиять на формирование изображения.

Включение режима выполняется в момент записи в порт 0CH.

– автоматически. Во время кадрового синхроимпульса производится перезапись таблицы цветов из СОЗУ в ОЗУ палитры. Перезапись выполняется прозрачно для процессора и не замедляет его работу. В этом расширенном режиме при формировании цвета учитываются атрибуты.

Включение режима выполняется в момент записи в порт 47H.

При инициализации системы автоматически включится расширенный графический режим (поскольку в процессе инициализации происходит запись служебного СОЗУ через порт 47H). При необходимости его включения в дальнейшем следует выполнить следующую последовательность команд:

DI

OUT (47H),A ; Включаем атрибуты и выключаем управляющие сигналы (значение A – безразлично)

LD A,0E0H ; Тут нужно загрузить в A требуемое для программы значение

OUT (45H),A ; Сброс схемы записи в служебное СОЗУ

XOR A ; Не используем маску записи в ОЗУ

OUT (46H),A ; Включаем управляющие сигналы STACK и RET

EI

Примечания:

– Каждое нечётное обращение к порту 47H отключает запись в ОЗУ, для включения записи в ОЗУ нужна повторная запись в порт 47H (но при этом будет выполнена запись в служебное СОЗУ), либо можно сбросить схему в исходное записью в порт 45H.

– Обращение к порту 47H отключает формирование управляющих сигналов STACK и RET. Для их включения необходимо выполнить запись в порт 46A.

Описание портов

Порты Вектор-06ц

Поддерживается два режима адресации портов (см. порт 45H) – в полном диапазоне от 0000H до FFFFH и усечённом – от xx00H до xxFFH, где старший байт адреса игнорируется.

В каждом из этих режимов выделяются диапазоны адресов от xx40H до xx7FH и от xxC0H до xxFFH, в которых обращение к портам выполняется на максимально возможной скорости. По остальным адресам каждое обращение к порту удлиняется примерно до 500-600 нс, а также обеспечивается минимальный временной зазор между последовательными обращениями к портам – 800-1000 нс. Это необходимо для корректной работы медленных периферийных устройств.

**00H** – регистр управления внутреннего параллельного порта (КР580ВВ55)

Формат управляющего слова:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 |
| 0 | x x x | | | 000 – PC0 001 – PC1  010 – PC2 011 – PC3  100 – PC4 101 – PC5  110 – PC6 111 – PC7 | | | 0 – уст. «0»  1 – уст. «1» |
| 1 | Режим PA  00 – режим 0  01 – режим 1  1x – режим 2 | | PA0–PA7  0 – вывод  1 – ввод | PC4–PC7  0 – вывод  1 – ввод | Режим PB  0 – реж. 0  1 – реж. 1 | PB0–PB7  0 – вывод  1 – ввод | PC0–PC3  0 – вывод  1 – ввод |

Примечание – Как правило в этот порт записывается значение 88H, а при сканировании клавиатуры – 8AH.

**01H** – порт C внутреннего параллельного порта

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 |
| РУС/ЛАТ | УС | СС | Вход с  магн. | РУС  0 –погашен  1 – горит | Не исп. | Автосброс | Выход на магн. |
| 0 – нажата  1 – отжата | | |

Примечание – Автосброс работает только в режиме начального загрузчика, для его активации нужно изменить состояние соответствующего бита с 0 на 1.

**02H** – порт B внутреннего параллельного порта

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Режим | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 |
| Отображение, вывод | x | x | x | 256/512 | Код цвета бордюра | | | |
| Синхроимпульс, вывод | х | х | х | х | Код цвета при программировании палитры | | | |
| Синхроимпульс, ввод | Чтение результата сканирования клавиатуры | | | | | | | |

**03H** – порт A внутреннего параллельного порта

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Режим | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 |
| Отображение, вывод | Номер верхней отображаемой строки | | | | | | | |
| Синхроимпульс, вывод | Код сканирования клавиатуры | | | | | | | |

**04H** – регистр управления параллельного интерфейса (КР580ВВ55)

**05H** – порт C параллельного интерфейса

**06H** – порт B параллельного интерфейса

**07H** – порт A параллельного интерфейса

**08H** – регистр управления интегрального таймера (КР580ВИ53)

Формат управляющего слова:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 |
| Выбор канала  00 – канал 0  01 – канал 1  10 – канал 2 | | 00 – чтение на лету  01 – R/W младшего байта  10 – R/W старшего байта  11 – R/W слова | | 000 – режим 0  001 – режим 1  010 – режим 2  011 – режим 3  100 – режим 4  101 – режим 5 | | | 1 – двоично-десятичный счёт  0 – двоичный счёт |

Описание режимов работы таймера приведено в [приложении](#Режимы_работы_КР580ВИ53).

**09H** – канал 2 интегрального таймера

**0AH** – канал 1 интегрального таймера

**0BH** – канал 0 интегрального таймера

**0CH** (– **0FH)** – порт программирования палитры.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 |
| B1 | B0 | G2 | G1 | G0 | R2 | R1 | R0 |

Палитра может быть запрограммирована во время кадрового синхроимпульса. Для этого в порт 02H нужно занести код цвета, а затем в порт 0CH соответствующий ему физический цвет. Необходимо выполнить эту последовательность для всех шестнадцати кодов цвета.

Примечание – Значения атрибутов при этом методе программирования цвета будут игнорироваться.

Квазидиск 256к

**10H** (– **13H)** – порт управления квазидиском 256кбайт

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 |
| x | x | Теневое ОЗУ  0 – запрещено  1 – разрешено | Стековый доступ  0 – запрещён  1 – разрешён | Стековый доступ | | Теневое ОЗУ | |
| 00 – 40000–4FFFF  01 – 50000–5FFFF  10 – 60000–6FFFF  11 – 70000–7FFFF | | 00 – 4A000–4DFFF  01 – 5A000–5DFFF  10 – 6A000–6DFFF  11 – 7A000–7DFFF | |

Музыкальный сопроцессор

**14H,** **15H** – музыкальный сопроцессор

Описание сопроцессора приведено в [приложении](#AY_3_8910).

Контроллер дисковода

**18H** – регистр данных (Data) контроллера НГМД.

**19H** – регистр сектора дорожки (Sector) контроллера НГМД.

**1AH** – регистр дорожки (Track) контроллера НГМД.

**1BH** – регистр команд (Command) и статуса (Status) контроллера НГМД.

**1CH** – регистр выбора и управления контроллера НГМД

Описание контроллера приведено в [приложении](#Контроллер_дисковода).

Часы реального времени

**20H** – адрес регистра часов реального времени;

**21H** – чтение или запись данных для выбранного регистра.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Адрес | Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | Назначение | Диапазон |
| 00H | 0 | 10 секунд | | | Секунды | | | | Часы | 00-59 |
| 01H | 0 | 10 секунд | | | Секунды | | | | Будильник | 00-59 |
| 02H | 0 | 10 минут | | | Минуты | | | | Часы | 00-59 |
| 03H | 0 | 10 минут | | | Минуты | | | | Будильник | 00-59 |
| 04H | AM/PM | 0 | 0 | 10 ч | Часы | | | | Часы | 01-12 +AM/PM  00-23 |
| 0 | 10 часов | | Часы | | | |
| 05H | AM/PM | 0 | 0 | 10 ч | Часы | | | | Будильник | 01-12 +AM/PM  00-23 |
| 0 | 10 часов | | Часы | | | |
| 06H | 0 | 0 | 0 | 0 | 0 | День недели | | | Часы | 01-07 |
| 07H | 0 | 0 | 10 дней | | День | | | | Часы | 01-31 |
| 08H | 0 | 0 | 0 | 10 М | Месяц | | | | Часы | 01-12 |
| 09H | 10 лет | | | | Год | | | | Часы | 00-99 |
| 0AH | UIP | DV2 | DV1 | DV0 | RS3 | RS2 | RS1 | RS0 | Управление | - |
| 0BH | SET | PIE | AIE | UIE | SQWE | DM | 24/12 | DSE | Управление | - |
| 0CH | IRQF | PF | AF | UF | 0 | 0 | 0 | 0 | Управление | - |
| 0DH | VRT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Управление | - |
| 0EH-  7FH | X | X | X | X | X | X | X | X | ОЗУ | - |

Назначение битов управления приведено в [приложении](#RTC).

Порты Вектор Турбо+

**40H** – порт установки страницы расширенной памяти, предназначенной для замещения основной в адресных фрагментах, задаваемых портом 42H.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 |
| 0 – запрещение теневого ОЗУ  1 – разрешение теневого ОЗУ | x x x | | | 0000 – страница 00000–0FFFF  0001 – страница 10000–1FFFF  …  1110 – страница E0000–EFFFF  1111 – страница F0000–FFFFF | | | |

*Биты 0 – 3* – адрес страницы в 64к; бит 7 – разрешение режима теневого ОЗУ.

Работает аналогично порту 10H, биты 0, 1 и 5, соответственно.

**41H** – порт установки страницы расширенной памяти, к которой будет обращение на чтение/запись через стековые операции (PUSH/POP).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 |
| 0 – запрещение стековых операций  1 – разрешение стековых операций | x x x | | | 0000 – страница 00000–0FFFF  0001 – страница 10000–1FFFF  …  1110 – страница E0000–EFFFF  1111 – страница F0000–FFFFF | | | |

*Биты 0 – 3* – адрес страницы в 64к; бит 7 – разрешение режима.

Работает аналогично порту 10H, биты 2, 3 и 4, соответственно.

**42H** – порт маски памяти для режима теневого ОЗУ. Каждый установленный бит разрешает замещение соответствующего блока 8к:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 |
| E000-  FFFF | C000-  DFFF | A000-  BFFF | 8000-  9FFF | 6000-  7FFF | 4000-  5FFF | 2000-  3FFF | 0000-  1FFF |

Переключение между режимами доступа к расширенной памяти осуществляется по факту записи в порт 10H или в порты 40H, 41H.

По умолчанию, для совместимости со старым режимом, в порт следует записать 60H.

**43H** –порт установки отображаемой области памяти и режимов отображения.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 |
| 512 строк | 512 пикс. в строке | Порядок строк | 00000 – отображение 00000–07FFF  00001 – отображение 08000–0FFFF  …  11110 – отображение F0000–F7FFF  11111 – отображение F8000–FFFFF | | | | |

*Биты 0 – 4* – номер страницы отображаемой памяти с дискретностью 32к.

*Бит 5* – порядок отображения по вертикали (если =0, адреса снизу вверх (как в Векторе), если =1 – сверху вниз).

*Бит 6* – включение режима 512 пикселей в строке для каждой экранной плоскости. Экранная память, заданная портом 43H, отображается в левой половине экрана, а в правой отображается память смежной станицы в 64к (инвертируется бит 1 порта 43H).

*Бит 7* – включение режима 512 строк. Экранная память, заданная портом 43H отображается в чётных строках, а в нечётных отображается страница, адрес которой получается инвертированием бита 2 порта 43H. При этом при формировании цвета пикселей нечётных строк используются те же значения атрибутов, что и для чётных.

Пример – При записи в порт значения CFH, будет отображаться 512х512 пикселей в 256 возможных цветах, при этом в чётных строках в левой половине экрана будет отображён стандартный векторовский экран.

По умолчанию в порт следует записать 0FH.

**44H** – порт установки сдвига по горизонтали и тактовой частоты.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 |
| 00 – 12 МГц  01 – 6 МГц  (10 – 6 МГц)\*  11 – 3 МГц | | 000000 – без сдвига  000001 – сдвиг влево на 1 байт  …  111111 – сдвиг влево на 63 байта | | | | | |
| \* Такой код для 6 МГц использовать не рекомендуется | | | | | | | |

Биты 0 – 5 – фактически задают первый отображаемый байт в строке. При этом учитываются все возможные 64 байта. В режиме 256х256 при установке бита 5 будет отображаться смежный экран.

Биты 6 и 7 – задают эффективную скорость процессора. Фактически процессор всегда работает на частоте 12 МГц, для имитации работы на 6 МГц и 3 МГц добавляются такты ожидания.

Пример – если записать 01H, визуально произойдёт сдвиг изображения влево на 8 точек. В режиме 512х256, левые 8 точек отобразятся последними в строке. В режиме 256х256, справа отобразятся 8 первых точек смежного экрана (см.порт 43H).

По умолчанию в порт следует записать С0H.

**45H** – порт задания режимов записи в ОЗУ и системных модификаторов.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 |
| Порты  0 – 64к  1 – 256 | INT  0 – в ПЗУ  1 – в ОЗУ | NMI  0 – запр.  1 – разр. | 0 – без изм.  1 – отражение | 0 – без инв.  1 – с инв. | 000 – без сдвига  001…111 – циклический сдвиг влево на 1…7 бит | | |

*Биты 0 – 2* – задают значение циклического сдвига байта влево при записи в ОЗУ.

Пример – если биты содержат 03H, то при записи 01H фактически запишется 08H.

*Бит 3* – задаёт инвертирование записываемого байта (вместо 01H запишется FEH).

*Бит 4* – задаёт зеркальное отражение (вместо 01H запишется 80H).

Примечания к битам 0 – 4:

1 Биты 0 – 4 работают совместно в любых комбинациях. Порядок применения модификаторов: сдвиг, инвертирование, отражение;

2 Модификаторы записи влияют только на запись в ОЗУ (но не в СОЗУ).

3 Логический смысл функций преобразования, при необходимости, может быть задан произвольным образом (см. описание порта 47H), только для комбинации 00000 следует оставлять запись без модификации.

*Бит 5* – если =0, прерывания NMI запрещены, =1 – разрешены.

Примечание – Прерывания NMI всегда выполняются в ПЗУ (с адреса 0066H).

*Бит 6* – если =0, прерывания INT выполняются в ПЗУ, =1 – в ОЗУ. При этом переход на прерывание в ОЗУ будет по адресу 0038H, а в ПЗУ по адресу 0018H.

Примечание – Для программы, исполняемой в ПЗУ, по прерыванию будет переход по адресу 0038H. Это позволяет разместить в ПЗУ два разных обработчика прерываний – для загрузчика и программы в ОЗУ.

Примечания к битам 5 и 6:

1 При обработке прерываний NMI и INT в ПЗУ, фактически включается режим загрузчика, поэтому младшие 32кбайт ОЗУ в общем случае недоступны на чтение. Для доступа к этой области ОЗУ следует использовать стековые операции POP xx (PUSH xx тоже будут работать).

2 Для возврата в программу в ОЗУ из прерываний NMI и INT, выполняемых в ПЗУ, следует использовать команды RETN и RETI, соответственно;

3 Для возврата из прерывания в программу, выполняемую в ПЗУ (при работе начального загрузчика и других программ в ПЗУ), следует использовать команду RET.

4 Во время выполнения программ в ПЗУ, прерывания NMI автоматически блокируются.

5 Если кнопки **Fn+БЛК** удерживаются нажатыми, после выхода из прерывания NMI и выполнения одной команды в ОЗУ будет выполнено очередное прерывание.

*Бит 7* – задаёт режим работы адресации портов. Если = 0, адресуются все 65536 портов, если =1 – 256 портов, адресные линии A8 – A15 игнорируются.

По умолчанию в порт следует записать E0H.

При чтении из порта **45H**, *бит 7* будет содержать 1 во время кадрового синхроимпульса, в остальное время там будет читаться 0. Это можно использовать, например, для опроса клавиатуры при выполнении прерывания NMI чтобы исключить сдвиги экрана по вертикали в эти моменты.

Запись в порт **45H** сбрасывает в исходное механизм программирования служебного СОЗУ.

**46H** – порт маски записи.

Единица в каждом разряде порта блокирует запись соответствующего бита в память. Совместно с портом 45H это, позволяет значительно ускорить вывод текста и графики.

Примечание – перед началом работы с памятью с использованием портов 45H (биты 0-4) и 46H следует ОБЯЗАТЕЛЬНО запретить прерывания (INT и NMI).

По умолчанию в порт следует записать 00H.

Запись в порт 46H включает формирование системных сигналов (STACK и RET).

**47H** – порт доступа к служебному СОЗУ

Служебное СОЗУ содержит таблицу модификаторов записи в ОЗУ, а также данные для формирования системных сигналов – признаков обращения к стеку и возврата из прерываний. Без корректного программирования служебного СОЗУ нормальное функционирование компьютера невозможно.

Процедура записи в служебное СОЗУ:

1 запись в порт 45H (биты 0 – 5) адреса записи (биты A8 – A13);

2 запись в порт 47Н записываемого значения;

3 запись в порт 47H адреса записи (биты A0 – A7).

Сама запись выполняется на третьем шаге. Для последовательной записи 256 значений достаточно повторять только шаги 2 и 3.

Примечания.

1 После завершения записи в СОЗУ следует записать в порт 45H корректные режимы.

2 Запись в порт 47H всегда должна идти парой, либо завершаться записью в порт 45H (при этом схема управления служебным СОЗУ сбрасывается в исходное).

Распределение памяти служебного СОЗУ:

– 0000H – 1FFFH – таблица модификации записи;

– 2000H – 3FFFH – системные признаки.

Таблица модификации записи преобразует каждый записываемый в ОЗУ байт в значение, расположенное по адресу A0 – A13, где A0 – A7 – значение исходного байта, а A8 – A12 значения битов 0 – 4 порта 45H, A13 = 0 (значение бита 5 порта 45H безразлично).

В первом машинном цикле каждой команды производится чтение системных признаков из ячейки с адресом A0 – A13, где A0 – A7 – код прочитанной команды (или префикса), A8 – A12 значения битов 0 – 4 порта 45H, A13 = 1.

Системные признаки хранятся в битах:

– 0 – если = 1, прочитана команда чтения/записи из/в стек (POP/PUSH);

– 1 – если = 1, прочитана команда RETI или RETN (после префикса ED);

– 2 – если = 1, прочитан любой из префиксов CB, ED, DD или FD;

– 3 – если = 1, прочитан префикс ED.

– 4 – 7 не используются, должны быть сброшены в 0.

Запись в порт 47H включает расширенный графический режим с использованием атрибутов, а также выключает формирование системных сигналов STACK и RET.

Включение системных сигналов выполняется записью в порт 46H.

Программа начальной инициализации системы приведена в [приложении](#Начальная_инициализация).

Порты жёсткого диска

**50H** – R/W: Data (DR) — регистр данных (биты 0 – 7).

**58H** – R/W: Data (DR) — регистр данных (биты 8 – 15).

Фактически чтение или запись данных в HDD производится при обращении к порту 50H, порт 58H содержит старший байт 16-битного слова. Соответственно при чтении нужно сначала прочитать данные из порта 50H, а потом из 58H. При записи – наоборот, сначала записать в 58H порт, а потом в 50H.

**51H** – R: Error (ER) — регистр ошибок; W: Features (FR) — регистр свойств.

Регистр ошибок ER хранит состояние выполнения последней операции или диагностический код.

Назначение битов регистра ER:

– бит 0 — AMNF (Address Mark Not Found) — не найден адресный маркер данных в заголовке сектора;

– бит 1 — TK0NF (Track 0 Not Found) — указывает на то, что по команде Recalibrate не удалось найти нулевой трек;

– бит 2 — ABRT (Aborted Command) — устанавливается, если команда отвергнута как недействительная или в случае возникновения иной ошибки;

– бит 3 — MCR (Media Change Requested) — индикатор запроса смены носителя (после обнаружения запроса смены носителя команды Door Lock будут возвращать бит ошибки ERR и бит MCR, бит MCR сбрасывается командами Door Unlock, Media Eject или сигналом аппаратного сброса);

– бит 4 — IDNF (ID Not Found) — указывает на ненайденный идентификатор сектора;

– бит 5 — МС (Media Changed) — смена носителя (после смены носителя первая команда обращения отвергается и устанавливается данный бит, после сброса бита следующие команды будут выполняться нормальным образом);

– бит 6 — UNC (Uncorrectable Data Error) — неисправимая ошибка данных;

– бит 7 — зарезервирован.

После выполнения любого сброса или команды Execute Device Diagnostic регистр ошибок содержит диагностический код. Трактовка битов, за исключением бита 2 (ABRT), может меняться в зависимости от исполненной команды.

**52H** – R/W: Sector Count (SC) — регистр счетчика секторов.

Регистр счетчика секторов SC содержит число секторов, участвующих в обмене. Хост инициализирует этот регистр до подачи команды (нулевое значение соответствует 256 секторам). По успешному завершению операции доступа к данным регистр должен обнулиться. Если команда завершилась с ошибкой, в регистре будет число секторов, которые должны быть переданы для успешного завершения предыдущего запроса. Команды Initialize Device Parameters или Write Same могут переопределить значение этого регистра. В некоторых командах регистр используется для передачи иных параметров.

**53H** – R/W: Sector Number (SN) — регистр номера сектора / LBA[7:0] \*

**54H** – R/W: Cylinder Low (CL) — регистр младшего байта номера цилиндра / LBA[15:8] \*

**55H** – R/W: Cylinder High (CH) — регистр старшего байта номера цилиндра / LBA[23:16] \*

**56H** – R/W: Device/Head (D/H) — регистр номера устройства и головки / LBA[27:24] \*

Назначение битов регистра D/H:

– биты 7 и 5 вплоть до ATA-3 должны были быть единичными, в ATA/ATAPI-4 их объявили устаревшими;

– бит 6 – L – единичным значением указывает на применение режима адресации LBA, при нулевом значении бита используется режим CHS;

– бит 4 – DEV (Device) — выбор устройства, при DEV=0 выбрано ведущее, при DEV=1 – ведомое;

– биты [3:0] имеют двоякое назначение в зависимости от выбранной системы адресации, в режиме CHS они содержат номер головки, в режиме LBA — старшие биты логического адреса.

Как и предыдущие (SN, CH и CL), адресный регистр D/H инициализируется хост-адаптером, а в случае возникновения ошибки при операции устройство поместит в них адрес, по которому встретилась ошибка. До принятия спецификации ATA-2 считалось, что адресные регистры должны модифицироваться и после успешного выполнения операции, отражая текущее значение адреса в носителе.

**57H** – Status (SR) — регистр состояния; W: Command (CR) — регистр команд

Назначение битов регистра SR описано ниже.

• Бит 7 — BSY (Busy) указывает на занятость устройства, значение этого бита действительно всегда. При BSY=1 устройство игнорирует попытки записи в командный блок регистров, а чтение этих регистров дает неопределенный результат. При BSY=0 регистры командного блока доступны, в это время устройство не может устанавливать бит DRQ, изменять значение битов ЕRR и содержимое остальных командных регистров (могут меняться только значения битов IDX, DRDY, DF, DSC и CORR). Бит может устанавливаться на кратковременный интервал, так что хост может этого не заметить. Бит устанавливается:

• при сбросе устройства;

• по получении команды, если не устанавливается DRQ;

• между передачами блоков данных в режиме PIO и после них, пока не обнулился DRQ;

• во время передач данных в режиме DMA.

• Бит 6 — DRDY (Device Ready) указывает на готовность устройства к восприятию любых кодов команд. Если состояние бита изменилось, оно не может вернуться обратно до чтения регистра состояния. При DRDY=0 устройство воспринимает только команды Execute Device Diagnostic и Initialize Device Parameters, прекращая выполнение текущей команды и сообщая об этом флагом ABRT в регистре ошибок и флагом ERR в регистре состояния. Другие команды приводят к непредсказуемым результатам. Устройства ATAPI сбрасывают бит по любому сбросу и команде Execute Device Diagnostic. Бит устанавливается устройством ATA, когда оно готово к выполнению всех команд. Устройство ATAPI устанавливает бит до завершения выполнения команд, за исключением команд Device Reset и Execute Device Diagnostic.

• Бит 5 — DF (Device Fault) — индикатор отказа устройства.

• Бит 4 — DSC (Device Seek Complete) — индикатор завершения поиска трека. В командах, допускающих перекрытие, бит называется SERV (Service Required) — устройство требует обслуживания.

• Бит 3 — DRQ (Data Request) — индикатор готовности к обмену словом или байтом данных.

• Бит 2 — CORR (Corrected Data) — индикатор исправленной ошибки данных.

• Бит 1 — IDX (Index) — индекс, трактуется особо каждым производителем.

• Бит 0 — ERR (Error) — индикатор ошибки выполнения предыдущей операции.

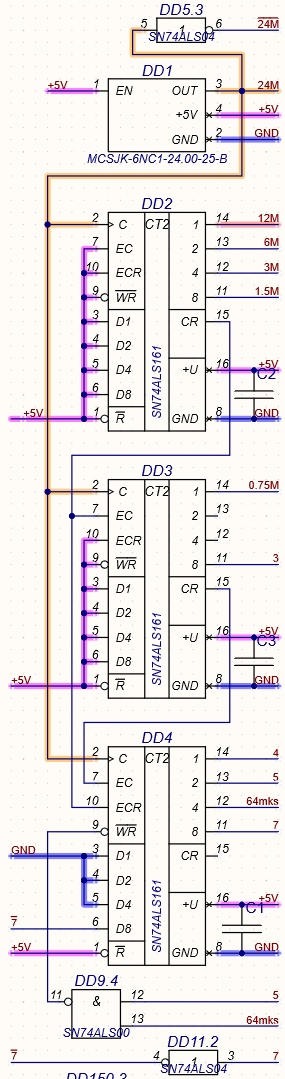
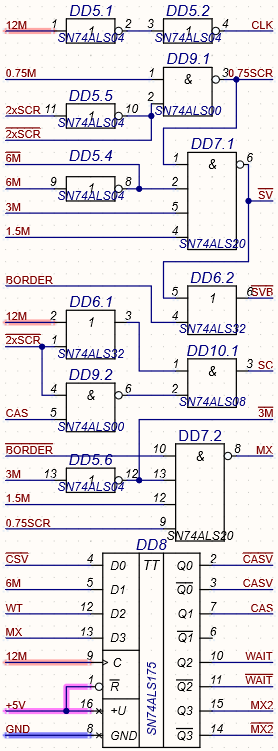
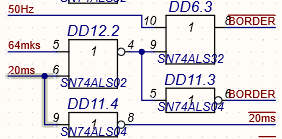
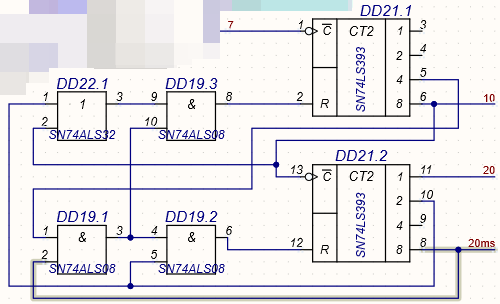
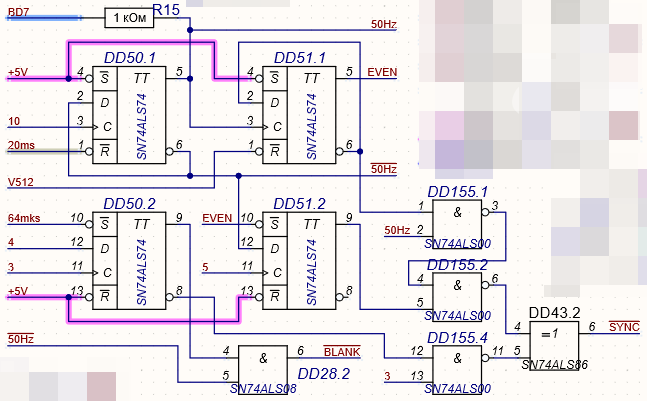
Дополнительная информация содержится в регистре ошибок. Если установлен бит ERR, до приема следующей команды, программного или аппаратного сброса устройство не изменит состояние этого бита, а также регистра ошибок, регистра количества секторов и регистров цилиндра, головки и номера сектора. Для команд Packet и Service бит называется CHK и служит признаком исключительной ситуации.

В стандарте ATA/ATAPI-4 для некоторых команд биты 4 и 5 могут иметь иное назначение, а биты 1 и 2 объявлены устаревшими.

Примечание – Регистры сектора, цилиндра и головки в режиме LBA содержат указанные биты логического адреса.

Назначение битов регистра команд (CR) описано ниже в [приложении](#Команды_HDD).

**5FH** – Системный сброс жёсткого диска.

Описание работы схемы

**Схема формирователя основных управляющих сигналов**

Базовая частота 24 МГц поступает с генератора DD1, после чего она делится счётчиками DD2, DD3, DD4, на которых формируются необходимые частоты, из которых далее формируются управляющие сигналы. Триггерами DD8 ряд сигналов сдвигаются и выравниваются по 12 МГц.

Описание сигналов:

***CLK*** – тактовая частота процессора (12 МГц);

***SV*** – выборка очередного отображаемого байта;

***SVB*** – тоже что и SV, но с учётом бордюра;

***SC*** – строб сдвиговых регистров экранных плоскостей;

***2xSCR*** – сигнал с порта 43H, бит 6 – включение режима 512 точек в строке для всех экранных плоскостей;

***CASV*** – сигнал CAS для выборки отображаемых байтов экранных плоскостей;

***MX2*** – сигнал переключения адресов между процессором и схемой отображения;

***BORDER*** – строчный и кадровый бордюры;

***64mks*** – строчный бордюр;

***20ms*** – кадровый бордюр;

***50Hz*** – кадровый синхроимпульс;

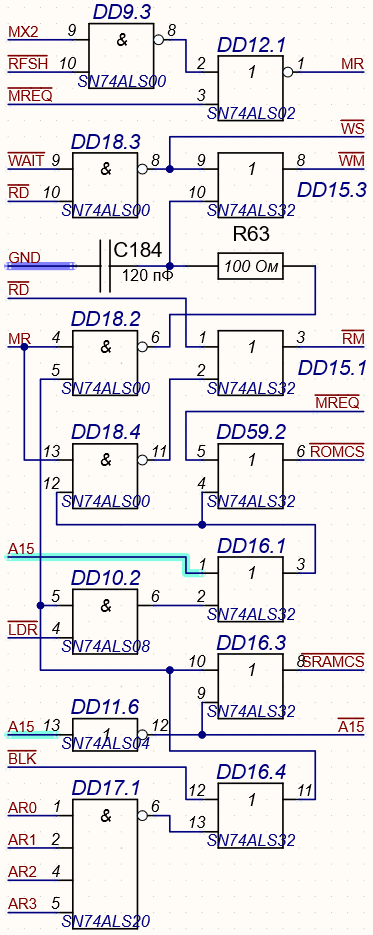
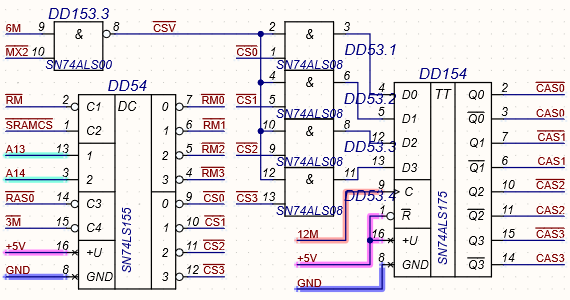
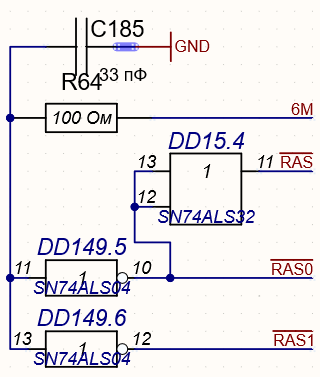
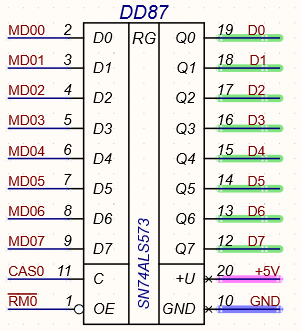
***V512*** – сигнал с порта 43H, бит 7 – включение режима 512 строк;

***EVEN*** – признак чётных/нечётных строк;

***BLANK*** – сигнал гашения;

***SYNC*** – сигнал синхронизации.

Назначение сигналов CSV, CASV, CAS, WAIT описано ниже.

**Схема управления динамическим ОЗУ**

Доступ к ОЗУ разделяется между процессором и схемой отображения. Для предотвращения конфликтов сигнал обращения процессора к ОЗУ (**MR**) может быть активен только в соответствующие моменты (MX2 = 1).

Сигнал записи в память (/WR) формируется процессором Z80 слишком поздно (во втором такте) что не позволяет его использовать на 12МГц без задержек.

В схеме вместо /WR используется сигнал /MREQ при неактивном /RD, таким образом сигнал записи появляется уже в первом такте (**/WM**).

Объединение /RD с /WAIT на DD18.3 нужно чтобы результирующий сигнал (**/WM**) был активен только одновременно с сигналами **/CAS0** **- /CAS3**.

R/C цепочка C184, R63 фильтрует паразитные импульсы из-за задержек на вентилях.

Во время обращения к базовой странице памяти через порты 40H или 41H (**AR0-AR3** = 1 и **/BLK**=0), сигналы обращения к ОЗУ (**/WM**, **/RM**) блокируются и активируются **/ROMCS** (A15=0) или **/SRAMCS** (A15=1).

Сигнал **/WS** заменяет /WR для записи в СОЗУ.

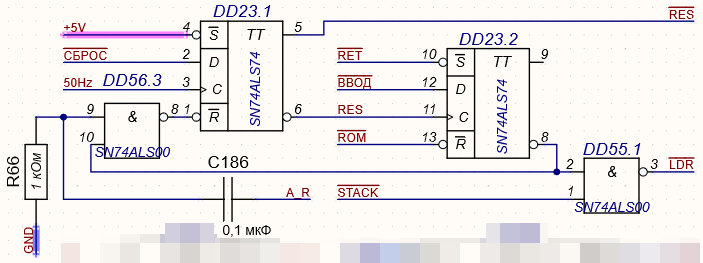
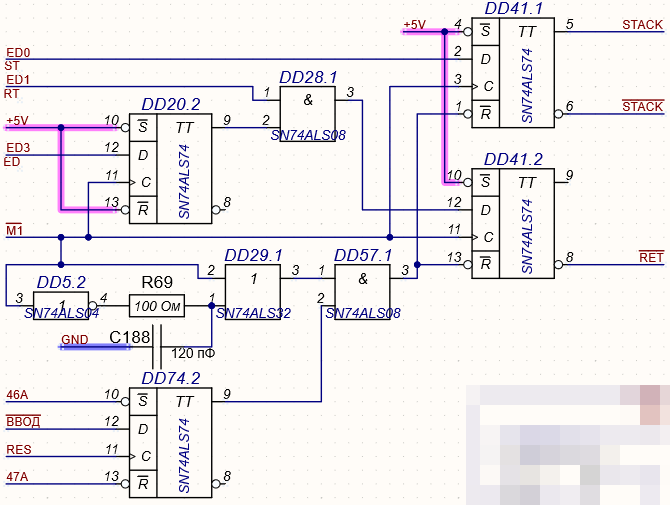
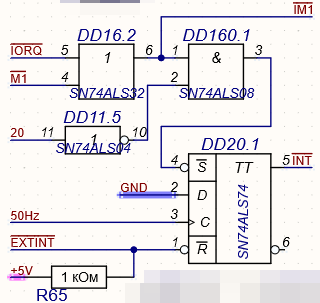
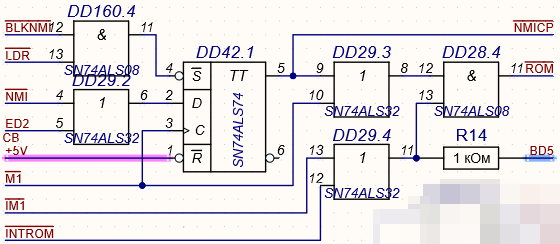
При активации режима загрузчика (**/LDR** = 0) активируется выборка ПЗУ (**/ROMCS**) и отключается чтение из ОЗУ в диапазоне 0000H-7FFFH.

Дешифратор DD54 коммутирует сигналы выборки буферных регистров (**/RM0 - /RM3**) и сигналы активации /CASx линеек ОЗУ (**/CS0 - /CS3**).

Для исключения конфликта между ОЗУ и СОЗУ – при активном /SRAMCS на входе дешифратора чтение из ОЗУ блокируется.

R/C цепочка C185, R64 сдвигает сигнал /RAS чтобы к его приходу успевал сформироваться адрес.

**Схема управления сбросом, прерываниями и системными сигналами**



**ST** – PUSH, POP, EX (SP),HL;

**RT** – RETI, RETN;

**CB** – префиксы CB, ED, DD, FD;

**ED** – префикс ED.

Сигнал STACK формируется из ST, стробированием M1 (чтобы это точно был код команды, а не данные), этот сигнал нужен для работы «квазидиска» в одном из режимов адресации.

Поскольку команды RETI и RETN двухбайтовые, формирование сигнала RET выполняется в два этапа из ED и RT.

Сигнал RET служит для деактивации ПЗУ при выходе из подпрограммы прерывания командами RETI и RETN.

Сигнал ROM активирует ПЗУ при поступлении сигналов маскируемого или немаскируемого прерываний (см. порт 45H). При этом переключение должно выполняться строго в промежутке между командами (фиксация сигналов прерываний процессором происходит в последнем такте последнего цикла каждой команды).

Поступивший сигнал NMI защёлкивается в триггере по заднему фронту M1, при этом пропускаются возможные префиксы команды (сигнал CB, иначе возможна ситуация, когда часть байтов команды будет прочитано из ОЗУ, а часть из ПЗУ), на процессоре появляется сигнал запроса NMI (NMICP = лог.0). Сигнал ROM появляется в начале первого цикла следующей команды. Таким образом синхронизируется начало обработки прерывания и активация ПЗУ.

Активация ПЗУ при обработке маскируемого прерывания происходит по запросу вектора прерывания (IM1 = M1 + IORQ), поскольку при этом не происходит обращения к ОЗУ, тут нет жёсткого требования к моменту переключения.

Приложения

Программа начальной инициализации системы

di

im 0

ld a,0e0h

ld bc,0045h

out (c),a ; 256 портов, INT в ОЗУ, NMI разрешены, без модификации записи

xor a

out (40h),a ; Адрес теневой страницы расширенной памяти

out (41h),a ; Адрес страницы памяти для обращения через стек

out (44h),a ; Первый отображаемый байт по горизонтали

ld a,0fh

out (43h),a ; Номер отображаемой страницы памяти

ld a,88h

out (00h),a ; ВВ55 основной

ld a,81h

out (04h),a ; ВВ55 внешний порт

ld a,0ffh

out (03h),a ; Задаём вертикальный скроллинг в исходное

out (05h),a

out (06h),a

out (07h),a

;--------------------------------------

; Формируем таблицу модификации записи

ld c,0e0h

ld d,0

wrset1: ld a,c

out (45h),a ; Записываем номер функции преобразования байта

wrset2: ld b,0

wrset3: ld hl,wrset4

ld a,07h

and c

ld e,a

sbc hl,de ; В зависимости от позиции сдвига формируем адрес перехода

ld a,b

jp (hl) ; Переходим на …

;-------------------------------------------------------------------

rlca ; <- сдвиг влево на 7 позиций

rlca ; <- сдвиг влево на 6 позиций

rlca ; <- сдвиг влево на 5 позиций

rlca ; <- сдвиг влево на 4 позиции

rlca ; <- сдвиг влево на 3 позиции

rlca ; <- сдвиг влево на 2 позиции

rlca ; <- сдвиг влево на 1 позицию

wrset4: ld e,a ; <- без сдвига

ld a,c

and 08h

jr z,wrset5

ld a,e ; Инвертирование байта

cpl

ld e,a

wrset5: ld a,c

and 10h

ld a,e

jr z,wrset6

xor a ; Зеркальное отражение битов байта

rlc e

rra

rlc e

rra

rlc e

rra

rlc e

rra

rlc e

rra

rlc e

rra

rlc e

rra

rlc e

rra

wrset6: out (47h),a ; Записываемое в ОЗУ значение

ld a,b

out (47h),a ; Значение от процессора

djnz wrset3

inc c

jr nz,wrset1

;--------------------------------------

; Формируем таблицу системных сигналов

ld c,0c0h

wrset7: ld b,0

ld a,c

out (45h),a

wrset8: ld a,b

cp 45h ; RETN

jr z,wrset9 ; 02h

cp 4Dh ; RETI

jr z,wrset9 ; 02h

cp 55h ; RETN

jr z,wrset9 ; 02h

cp 5Dh ; RETI

jr z,wrset9 ; 02h

cp 65h ; RETN

jr z,wrset9 ; 02h

cp 6Dh ; RETI

jr z,wrset9 ; 02h

cp 75h ; RETN

jr z,wrset9 ; 02h

cp 7Dh ; RETI

jr z,wrset9 ; 02h

cp 0C1h ; POP BC

jr z,wrset10 ; 01h

cp 0C5h ; PUSH BC

jr z,wrset10 ; 01h

cp 0D1h ; POP DE

jr z,wrset10 ; 01h

cp 0D5h ; PUSH DE

jr z,wrset10 ; 01h

cp 0E1h ; POP HL

jr z,wrset10 ; 01h

cp 0E3h ; EX (SP),HL

jr z,wrset10 ; 01h

cp 0E5h ; PUSH HL

jr z,wrset10 ; 01h

cp 0F1h ; POP AF

jr z,wrset10 ; 01h

cp 0F5h ; PUSH AF

jr z,wrset10 ; 01h

cp 0CBh ; CB

jr z,wrset11 ; 04h

cp 0DDh ; DD

jr z,wrset11 ; 04h

cp 0EDh ; ED

jr z,wrset12 ; 0ch

cp 0FDh ; FD

jr z,wrset11 ; 04h

xor a

jp wrset13

wrset9: ld a,02h

jp wrset13

wrset10:

ld a,01h

jp wrset13

wrset11:

ld a,04h

jp wrset13

wrset12:

ld a,0ch

jp wrset13

wrset13:

out (47h),a ; Системные сигналы

ld a,b

out (47h),a ; Код команды или префикса

djnz wrset8

inc c

ld a,20h

and c

jr z,wrset7

;--------------------------------------

xor a

out (46h),a ; Маска записи в ОЗУ, а также включение системных сигналов

ld a,0e0h

out (45h),a ; 256 портов, INT в ОЗУ, NMI разрешены, без модификации записи

clrSRAM:

ld a,0f0h

out (42h),a ; Замещаем верхние 32кб статическим ОЗУ

ld a,87h

out (40h),a ; Активируем теневое ОЗУ

ld hl,8000h

ld c,00h

clrSRAM3:

ld (hl),c

inc hl

ld a,l

or h

jr nz,clrSRAM3

Режимы работы интегрального таймера (КР580ВИ53)

**Режим 0** – выдача сигнала прерывания по конечному числу. При работе в этом режиме на выходе канала появляется уровень "0" сразу же после установления режима работы. После загрузки числа в счетчик канала выход остается в "0" и счетчик начинает считать, если на входе разрешения установлен уровень "1". После того как достигается конечное число, на выходе устанавливается уровень "1" и остается до тех пор, пока канал не будет перезагружен режимом работы или новым числом.

**Режим 1** – ждущий мультивибратор с программно-устанавливаемой длительностью сигнала. В этом режиме выход канала после загрузки числа в счетчик канала устанавливается в уровень "0" после первого тактового сигнала, следующего за передним фронтом на управляющем входе. Одновременно начинается счет, а при достижении конечного числа на выходе устанавливается уровень "1".

**Режим 2** – генератор тактовых сигналов. В этом режиме на выходе канала через число периодов тактовой частоты, записанное в счетчике канала, появляется уровень "0" длительностью в один период тактовой частоты.

**Режим 3** – генератор прямоугольных сигналов. В этом режиме на выходе канала будет уровень "1" в течение первой половины интервала времени, определяемого числом в счетчике, и уровень "0" в течение второй половины.

**Режим 4** – программно-управляемый строб. После установки режима 4 на выходе канала появляется уровень "1". Когда число полностью загружено в счетчик канала и на управляющий вход подан уровень "1", начинается счет, и при достижении конечного числа на выходе появляется импульс уровня "0" длительностью в один период тактовой частоты.

**Режим 5** – аппаратно-управляемый строб. Работа канала в этом случае аналогична работе в режиме 4 с той лишь разницей, что счетчик канала после загрузки начинает счет только по переднему фронту на управляющем входе. Кроме того, если во время счета на управляющем входе снова появится передний фронт сигнала, то счет будет начат сначала.

Программируемый генератор звуков AY-3-8910

Программируемый генератор звуков AY-3-8910/8912 (ISG) фирмы General instrument представляет собой БИС, позволяющую воспроизводить широкую гамму сложных звуковых эффектов, и легко сопрягается с любым 8/16 разрядным микропроцессором. Он имеет 3 программно-независимых звуковых канала и два (для 8912 - один) 8 разрядных универсальных порта ввода/вывода. ISG может выдавать звуковые сигналы в диапазоне 8 ократор звуков AY-3-8910 и AY-3-8912.

Программирование AY-3-8910/8912

ISG является регистро ориентированным генератором звуков. Его функции выполняются посредством 16 внутренних регистров. Номер регистра задается 4 младшими разрядами при подаче команды "фиксация адреса" и остается действительным до получения команды о смене этого адреса.

-----------------------------------------

N регистра | Назначение

-----------------------------------------

0, 2, 4 | Нижние 8 бит частоты голосов A, B, C; может принимать значения от 0 до 255.

1, 3, 5 | Верхние 4 бита частоты голосов A, B, C; может принимать значения от 0 до 15.

6 | Управление частотой генератора шума; может принимать значения от 0 до 31.

7 | Управление смесителем и вводом/выводом; может принимать значения от 0 до 255.

8, 9, 10 | Управление амплитудой каналов A, B, C; может принимать значения от 0 до 16.

11 | Нижние 8 бит управления периодом пакета; может принимать значения от 0 до 255.

12 | Верхние 8 бит управления периодом пакета; может принимать значения от 0 до 255.

13 | Выбор формы волнового пакета; может принимать значения от 0 до 15.

14, 15 | Регистры портов ввода/вывода могут принимать значения от 0 до 255.

Основным при работе ISG является регистр 7. Его главное назначение - определять какие каналы должны участвовать в образовании звука и определять направление обмена портов ввода/вывода.

7 | 6 | 5 | 4 | 3 | 2 | 1 | 0

------+---+-----+---+---+-----+---+------

Порт A|- B|Шум C|- B|- A|Тон C|- B|- A

----------+-------------+----------------

ввод/вывод|кан. для шума| канала для тона

-----------------------------------------

При установлении в регистрах величины 16, амплитуда в канале управляется встроенным, общим для всех трех каналов, генератором огибающей. Выбор типа огибающей и ее затухание осуществляется в регистре 13.

bit 0 - затухание bit 1 - изменение

bit 2 - нарастание bit 3 - продолжение

Примеры:

OUT 65533,Reg\_Numb OUT 49149,Data

Выстрел:

10 OUT 65533,6 : OUT 49149,31

20 OUT 65533,7 : OUT 49149,7

30 OUT 65533,8 : OUT 49149,16

40 OUT 65533,9 : OUT 49149,16

50 OUT 65533,10: OUT 49149,16

60 OUT 65533,12: OUT 49149,18

70 OUT 65533,13: OUT 49149,0

Программирование контроллера дисковода

В схеме управления дисководом используется контроллер MB8877A или его отечественный аналог КР1818ВГ93.

Контроллер содержит пять внутренних регистров:

– Data (запись, чтение) – служит для обмена данными между процессором и контроллером;

– Track (запись, чтение) – содержит номер текущей дорожки;

– Sector (запись, чтение) – содержит номер сектора;

– Status (чтение) – содержит статус выполнения команды;

– Command (запись) – служит для задания команд.

Примечание – Чтение статуса возможно не ранее чем через 30мс после записи команды.

Контроллер обеспечивает прием и выполнение 11 команд. Все команды условно разделены на 4 типа: вспомогательные, записи и чтения информации, поиска и чтения индексного поля и принудительного прерывания.

**Вспомогательные команды.**

Команды этого типа выполняются вне зависимости от сигнала готовности накопителя:

RESTORE – обеспечивает переход головки на нулевую дорожку. Если нулевая дорожка не достигнута после 256 шагов, выполнение команды прекращается;

SEEK – обеспечивает перемещение головки до тех пор, пока не будет достигнута требуемая дорожка. При этом регистр Data должен содержать номер требуемой дорожки, а Track - текущей. По окончании регистр Track содержит новый номер текущей дорожки;

STEP – обеспечивает перемещение головки на один шаг. Направление перемещения совпадает с направлением предыдущего движения головки;

STEPF – обеспечивают перемещение головки на один шаг вперед;

STEPB – обеспечивают перемещение головки на один шаг назад.

Состояние битов регистра Status в процессе и после выполнения этих команд:

R7: 1 = накопитель не готов.

R6: 1 = диск защищён от записи.

R5: 1 = головка опущена.

R4: 1 = ошибка поиска.

R3: 1 = ошибка в контрольном коде заголовка сектора

R2: 1 = головка на нулевой дорожке.

R1: 1 = индексный импульс.

R0: 1 = идет выполнение команды.

**Команды чтения и записи данных.**

Перед выполнением этих команд необходимо в регистры Track и Sector записать номер требуемой дорожки и сектора соответственно. Длина сектора задается кодом в индексной области (заголовке):

00 = 128 байт на сектор;

01 = 256 байт на сектор;

02 = 512 байт на сектор;

03 = 1024 байт на сектор.

Команда READSCT выполняется, когда прочитан заголовок сектора с параметрами, совпадающими с запрошенными в Track и Sector. Если за 10 оборотов диска запрошенный сектор не найден, вырабатывается признак СЕКТОР НЕ НАЙДЕН и прекращается выполнение команды. Если сектор найден, он байт за байтом считывается и передается в регистр Data. В момент передачи устанавливается сигнал DRQ и признак ЗАПРОС ДАННЫХ в Status. Регистр Data должен быть считан в течение 20мкс после установки признака. Если это не произойдет, то в Data записывается следующий байт и вырабатывается признак ПОТЕРЯ ДАННЫХ. В конце считывания проверяется контрольная сумма. В случае несовпадения выставляется признак ОШИБКА КОНТРОЛЬНОГО КОДА и выполнение команды прекращается, даже в случае многосекторного режима (m=1).

Команда WRITESCT выполняется аналогично. Если регистр Data не будет записан в течение 20мс после установки признака ЗАПРОС ДАННЫХ, то вырабатывается признак ПОТЕРЯ ДАННЫХ, а на диск записывается байт нулей.

Значение битов регистра Status:

READSCT WRITESECT

R7: 1 = накопитель не готов. накопитель не готов.

R6: всегда 0 1 = защита записи

R5: 1 = стёртые данные ошибка записи

R4: 1 = сектор не найден сектор не найден

R3: 1 = ошибка в контрольном коде ошибка в контрольном коде

R2: 1 = потеря данных потеря данных

R1: 1 = запрос данных запрос данных

R0: 1 = идёт выполнение команды идёт выполнение команды.

**Команды поиска и чтения индексного поля.**

Эти команды предназначены для поиска информации на диске и для форматирования.

Команда READADDR предназначена для определения положения головки. По этой команде последовательно считываются с диска и передаются процессору 6 байт индексной области первого обнаруженного на диске сектора: номер дорожки, номер стороны, номер сектора, код длины сектора, двухбайтная контрольная сумма. В процессе выполнения этой команды содержимое Sector разрушается (туда копируется содержимое Track). Если за 10 оборотов диска ни один сектор не будет найден, устанавливается признак СЕКТОР НЕ НАЙДЕН.

Команда READTRK предназначена для отладочных целей. По этой команде дорожка считывается целиком (начало дорожки определяется по индексному импульсу) и передается процессору. Проверка контрольного кода не производится.

Команда WRITETRK предназначена для форматирования дорожки. Информация для этой процедуры должна содержать все пробелы, индексные метки и т.д. Байты F5H-FEH - служебные и предназначены для записи контрольных кодов, меток и т.п. Запись дорожки начинается в момент прихода индексного импульса.

Содержимое регистра Status:

READADDR READTRK WRITETRK

R7: 1 = накопитель не готов 1 = накопитель не готов 1 = накопитель не готов

R6: всегда 0 всегда 0 1 = защита записи

R5: всегда 0 всегда 0 1 = ошибка записи

R4: 1 = сектор не найден всегда 0 всегда 0

R3: 1 = ошибка контроля всегда 0 всегда 0

R2: 1 = потеря данных 1 = потеря данных 1 = потеря данных

R1: 1 = запрос данных 1 = запрос данных 1 = запрос данных

R0: 1 = идет выполнение 1 = идет выполнение 1 = идет выполнение

**Команды прерывания.**

По этим командам выполнение текущей команды прекращается и генерируется сигнал INTRQ. В зависимости от кода команды, возможны различные условия генерации этого сигнала.

**Описание команд:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Команда | Биты | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RESTORE | 0 | 0 | 0 | 0 | h | V | r1 | r0 |
| SEEK | 0 | 0 | 0 | 1 | h | V | r1 | r0 |
| STEP | 0 | 0 | 1 | u | h | V | r1 | r0 |
| STEPF | 0 | 1 | 0 | u | h | V | r1 | r0 |
| STEPB | 0 | 1 | 1 | u | h | V | r1 | r0 |
| READSCT | 1 | 0 | 0 | m | S | E | C | 0 |
| WRITESCT | 1 | 0 | 1 | m | S | E | C | a0 |
| READADDR | 1 | 1 | 0 | 0 | 0 | E | 0 | 0 |
| REASTRK | 1 | 1 | 1 | 0 | 0 | E | 0 | 0 |
| WRITETRK | 1 | 1 | 1 | 1 | 0 | E | 0 | 0 |
| INTERRUPT | 1 | 1 | 0 | 1 | I3 | I2 | I1 | I0 |

h = 1 – опускать головку, h = 0 – не опускать.

V = 0 – не проверять положение головки, V = 1 – проверять.

u = 0 – не изменять Track во время выполнения команды; u = 1 – изменять.

m = 0 – работать с одним сектором, m = 1 – работать до конца дорожки.

S = 0 – нижняя сторона, S = 1 – верхняя сторона.

C = 0 – не проверять совпадение стороны, C = 1 – проверять.

Е = 1 – задержка 15мс после приема команды (подвод головки), Е = 0 – без задержки.

r1 = 0, r0 = 0 – время на шаг 6 мс

r1 = 0, r0 = 1 – время на шаг 12 мс

r1 = 1, r0 = 0 – время на шаг 20 мс

r1 = 1, r0 = 1 – время на шаг 30 мс.

а0 = 0 – обычная запись, а0=1 – запись стёртых данных.

I0 = 1 – прерывание по готовности накопителя.

I1 = 1 – прерывание по неготовности накопителя.

I2 = 1 – прерывание по индексному импульсу.

I3 = 1 – немедленное прерывание.

Структура данных для форматирования дорожки (MFM):

|  |  |  |  |
| --- | --- | --- | --- |
| Данные | Кол-во байт |  | Описание |
| 4E | 80 |  | Пробел от начала индексного импульса |
| 00 | 12 | Синхронизация |
| F6 | 3 | Запись байтов C2 |
| FC | 1 | Индексная метка начала дорожки |
| 4E | 50 |  | Пробел перед записью сектора |
| 00 | 12 |  | Синхронизация |
| F5 | 3 |  | Запись байтов A1 |
| FE | 1 |  | Индексная метка начала сектора |
| xx | 1 |  | Номер дорожки (00 – 4F) |
| xx | 1 |  | Номер стороны (00, 01) |
| xx | 1 |  | Номер сектора (01 – 05/08/0F/1A) |
| xx | 1 |  | Размер сектора (00 – 128; 01 – 256; 02 – 512; 03 – 1024) |
| F7 | 1 |  | Запись контрольной суммы |
| 4E | 22 |  | Пробел перед данными |
| 00 | 12 |  | Синхронизация |
| F5 | 3 |  | Запись байтов A1 |
| FB | 1 |  | Метка данных |
| E5 | xx |  | Запись данных (xx = размеру сектора) |
| F7 | 1 |  | Запись контрольной суммы |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

┌────────┬──────┬───────────────────────────────────────┐

│ Число │ Байт │ Назначение │

│ байтов │ │ │

│ десят. │ │ │

├────────┼──────┼───────────────────────────────────────┤

│ 80 │ 4Е │ Пробел от начала индексного импульса │

│ 12 │ 00 │ │

│ 3 │ F6 │ Запись байтов C2 │

│ 1 │ FC │ Индексная метка начала дорожки │

│ 50 ─┐ │ 4Е │ Пробел перед сектором │

│ 12 │ │ 00 │ │

│ 3 │ │ F5 │ Запись байтов A1 │

│ 1 │ │ FE │ Метка заголовка сектора │

│ 1 │ │ ХХ │ Номер дорожки │

│ 1 │ │ ХХ │ Номер стороны │

│ 1 │ │ ХХ │ Номер сектора │

│ 1 │ │ ХХ │ Код длины сектора │

│ 1 │ │ F7 │ Запись контрольной суммы │

│ 22 │ │ 4Е │ Пробел перед данными │

│ 12 │ │ 00 │ │

│ 3 │ │ F5 │ Запись байтов A1 │

│ 1 │ │ FB │ Метка данных │

│ ХХХ │ │ E5 │ Сектор │

│ 1 ─┘ │ F7 │ Запись контрольной суммы │

│ ### │ │ Запись остальных секторов │

│ &&& │ 4Е │ Продолжение записи до конца дорожки. │

└────────┴──────┴───────────────────────────────────────┘

Назначение служебных байтов (MFM):

F5: запись байта A1, запуск вычисления контрольного кода (суммы);

F6: запись байта C2;

F7: запись контрольного кода (контрольной суммы).

В режиме MFM на одну дорожку можно записать:

- пять секторов размером 1024 байта;

- девять (на некоторых накопителях - 10) секторов размером 512 байт;

- шестнадцать секторов размером 256 байт;

- двадцать шесть секторов размером 128 байт.

**Команды поиска и чтения индексного поля.**

Эти команды предназначены для поиска информации на диске и для форматирования.

Контроллеры Кишиневского варианта, Омского варианта и Кристы-2. Номер порта управления - 1CН. Доступен только по записи.

┌─────┬────────────────┬──────────────────┬──────────────────────┐

│ │ Кишиневский │ Омский │ Криста-2 │

├─────┼────────────────┴──────────────────┴──────────────────────┤

│ R0: │ выбор накопителя (0=A/C, 1=B/D) │

│ R1: │ Выбор 0=AB,1=CD│ Не задействован; всегда АВ │

│ R2: │ 1=нижняя сторона, 0=верхняя сторона │

│ R3: │ Не задействован │

│ R4: │ 0=8", 1=5" │ Не задействован; всегда 5" │

│ R5: │ 0=FM, 1=MFM │ Не задействован; всегда MFM │

│ R6: │ Не задействован │

│ R7: │ Не задействован│ Не задействован │ 0=стандартный, │

│ │ │ │ 1=совмещенный режим │

└─────┴────────────────┴──────────────────┴──────────────────────┘

Для запуска двигателя необходимо записать байт в регистр управления. После этого двигатель работает в течение 2.5 сек. Перезапись регистра возобновляет отсчет времени с момента последней записи.

Для того, чтобы что-то сделать с дисководом (прочитать/записать сектор), нужно уметь включать мотор, выбирать накопитель, позиционировать головку и обмениваться данными. Однако, как правило, человек, имеющий дисковод и контроллер, использует его не просто так, а под операционной системой (напр. CP/M). И система обычно не любит, чтобы пользовательская программа сама работала с регистрами контроллера - такие действия часто заканчиваются "повисанием". Поэтому рекомендуется в таких программах сохранять состояние всей используемой дисковой аппаратуры - регистров Sector и Track, положение головок всех используемых накопителей.

Вот пример программы, определяющей текущую дорожку, на которой находится головка накопителя, вне зависимости от того, какая дискета установлена (даже неформатированная).

SAFE: IN TRACK ; Сохраняем регистр Track

STA TRKBUF ; в ячейке TRKBUF,

IN SECTOR ; а регистр Sector -

STA SCTBUF ; в SCTBUF.

WAIT: CALL MOTOR ; Запустим мотор, выберем накопитель.

; Реализация этой процедуры зависит

; от типа контроллера

IN STATUS ; Накопитель готов (дверь закрыта)?

RLC ; Бит готовности -> флаг переноса

JC WAIT ; Нет, еще не готов.

MVI A,STEPF ; Делаем шаг вперед

CALL DO

MVI B,255 ; B - счетчик шагов

LOOP: MVI A,STEPB ; Делаем шаг назад

CALL DO

INR B ; Инкрементируем счетчик шагов.

IN STATUS ; Читаем результат.

ANI 4 ; Нулевая дорожка?

JZ LOOP ; Нет еще.

MOV A,B ; Иначе B содержит номер дорожки, на которой

STA TRK0 ; стояла головка

CALL STOP ; Остановим мотор (для контроллера Sphera+)

RET ; Закончим.

;

DO PROC ; Процедура выполнения вспомогательной команды

OUT COMMAND ; Пошлем команду в ВГ93

DO\_1: IN STATUS ; Команда уже выполняется?

RRC ; Бит выполнения -> флаг переноса

JNC DO\_1 ; Еще не выполняется.

DO\_2: IN STATUS

RRC ; Команда еще выполняется?

JC DO\_2 ; Да.

RET ; Команда выполнена.

DO ENDP

Для восстановления положения головки можно пользоваться, например, та-

кой программой:

REST: CALL MOTOR ; Запустим мотор

MVI A,RESTORE ; Головку - на нулевую дорожку

CALL DO

LDA TRK0 ; Номер требуемой дорожки ->

OUT DATA ; -> в регистр данных

MVI A,SEEK

CALL DO ; Позиционирование на сохраненную дорожку.

LDA TRKBUF ; Восстанавливаем регистры

OUT TRACK

LDA SCTBUF

OUT SECTOR

RET ; Вот и все.

Как было сказано выше, перед началом чтения нужно установить головку на нужную дорожку. Для этой цели можно использовать программу, аналогичную вышеприведенной.

Но необходимо помнить, что в дисковод может быть установлена дискета не того типа, на который рассчитан накопитель, например, в 80-дорожечный накопитель может быть вставлена 40-дорожечная дискета. В этом случае для перемещения головки на соседнюю дорожку необходимо перемещать ее на 2 шага, а значение в регистре Track изменять на 1. Чтобы определить, какая дискета установлена, можно воспользоваться информацией, содержащейся в системной области диска. Однако этот способ не универсальный. Гораздо надежнее, например, позиционировать головку на дорожку 2 и прочитать адрес (команда READADR). Если на самом деле головка стоит на 1 дорожке - то это ситуация 40дор. диск в 80дор. накопителе.

Предположим, что Вы установили головку на нужную дорожку и регистр Sector содержит номер требуемого сектора. Процесс чтения происходит так:

Запускается мотор дисковода.

Когда накопитель будет готов, посылается команда READSCT.

В подтверждение выполнения контроллер устанавливает бит выполнения.

Теперь нужно следить за состоянием бита ЗАПРОС ДАННЫХ.

Когда этот бит равен 1, необходимо прочитать содержимое регистра Data и записать его в память.

Так продолжать до тех пор, пока бит выполнения не очистится (можно для проверки окончания процесса чтения проверять состояние сигнала INTRQ (в контроллерах Coman и Sphere+), который равен 1 в момент завершения выполнения команды.

Из-за довольно низкой тактовой частоты процессора процесс чтения критичен ко времени(особенно на "Кристе-2"). Во время чтения прерывания должны быть запрещены. Более того, без применения программных ухищрений невозможно одновременно следить за битами ЗАНЯТО и ЗАПРОС ДАННЫХ. Как эта ситуация обходится - читайте ниже.

При чтении возможно возникновение ошибок. Наиболее часто возникают ошибки "Сектор не найден" и "Ошибка контрольного кода". Разберемся, почему они возникают.

Ошибка "Сектор не найден" возникает, если по каким-то причинам не произойдет идентификация сектора. Это случится, например, если в дисковод вставлена неформатированная дискета, неверно позиционирована головка или заголовок сектора поврежден. При чтении контроллер пытается прочитать заголовок сектора с требуемыми параметрами - номером дорожки, стороны (если затребована такая проверка) и сектора. Если он не будет найден за 10 оборотов диска, устанавливается признак СЕКТОР НЕ НАЙДЕН. Поскольку диск вращается со скоростью 5 оборотов в секунду, то для того, чтобы эта ошибка не приводила к повисанию, время выбега двигателя должно быть не менее 2.5 с (это относится только к контроллеру МикроДОС). ОШИБКА КОНТРОЛЬНОГО КОДА происходит, если, например, загрязнена головка дисковода, неверная юстировка, некачественная запись, наводки, помехи и т.п. Иногда для ее устранения полезно делать несколько попыток чтения сектора.

Приведем здесь два варианта программ чтения сектора. Первая из них при-

меняется в почти всех программах, работающих с диском (по-видимому, она -

ровесница ДОС). Вторая была разработана в Омском центре программирования и

применяется в Универсальном загрузчике Spase.

Будем считать, что мотор запущен и регистры Track и Sector содержат

верные параметры.

RSECT: DI ; Запретим прерывания - критично по времени

LXI H,BUF ; HL - адрес буфера, кратный 256

MVI A,READSCT ; Команду READSCT

OUT COMMAND ; пошлем в регистр команд

RRDY: IN STATUS ; Ожидаем начала выполнения:

RRC ; Выделим бит выполнения

JNC RRDY ; Он сброшен - ждем.

MVI B,2 ; Маска бита ЗАПРОС ДАННЫХ

LDA SCTSIZE ; Читаем код размера сектора

; (должен быть записан сюда заранее)

DCR A ; =1 ?

JZ R256 ; Да, сектор в 256 байт

DCR A ; =2 ?

JZ R512 ; Да, сектор в 512 байт

; Иначе - 1024 байта

R1024: IN STATUS ; Читаем статус

ANA B ; Выделяем бит запроса данных.

JZ R1024 ; Он сброшен - запроса нет.

IN DATA ; Читаем порт данных

MOV M,A ; Записываем в ОЗУ

INR L ; Инкрементируем смещение

JNZ R1024 ; 256 байт еще не прочитаны

INR H ; Читаем следующие 256 байт

R768: IN STATUS ; Читаем статус

ANA B ; Выделяем бит запроса данных.

JZ R768 ; Он сброшен - запроса нет.

IN DATA ; Читаем порт данных

MOV M,A ; Записываем в ОЗУ

INR L ; Инкрементируем смещение

JNZ R768 ; 256 байт еще не прочитаны

INR H ; Читаем следующие 256 байт

R512: IN STATUS ; Читаем статус

ANA B ; Выделяем бит запроса данных.

JZ R512 ; Он сброшен - запроса нет.

IN DATA ; Читаем порт данных

MOV M,A ; Записываем в ОЗУ

INR L ; Инкрементируем смещение

JNZ R512 ; 256 байт еще не прочитаны

INR H ; Читаем следующие 256 байт

R256: IN STATUS ; Читаем статус

ANA B ; Выделяем бит запроса данных.

JZ R256 ; Он сброшен - запроса нет.

IN DATA ; Читаем порт данных

MOV M,A ; Записываем в ОЗУ

INR L ; Инкрементируем смещение

JNZ R256 ; 256 байт еще не прочитаны

REND: IN STATUS ; Ждем завершения

RRC ; Завершено?

JC REND ; Нет еще. Ждем.

IN STATUS ; В аккумуляторе - код завершения.

Очевидно, что приведенная выше программа обладает недостатками. Основ-

ной из них - требование заранее знать размер сектора и размещать его по ад-

ресу, кратному 256 байт. Да и размер ее довольно велик. Ниже предлагается

программа, работающая по так называемому паритетному алгоритму.

RSECT: LXI H,BUF ; HL - адрес буфера

DI

MVI A,READSCT

OUT COMMAND

RRDY: IN STATUS

RRC

JNC RRDY ; до этого момента - аналогично предыдущему

MVI B,3 ; Маска ВЫПОЛНЕНИЕ & ЗАПРОС ДАННЫХ

JMP LOOP2 ; Обходим участок...

LOOP1: MOV M,A ; Заносим байт в ОЗУ

INX H ; Инкремент адреса

LOOP2: IN STATUS ; Читаем статус

ANA B ; Выделим нужные биты

; Флаг паритета процессора = 0, если только один из двух выделенных битов -

; единичный. Поскольку ситуация, при которой ЗАПРОС ДАННЫХ = 1, а ВЫПОЛНЕНИЕ

; =0 - невозможна, то (PF=0) обозначает, что команда еще выполняется и зап-

; роса данных нет.

JPO LOOP2

IN DATA ; Читаем порт данных

; Флаг паритета будет равен 1, если либо оба выделенных бита =1, и тогда

; нужно обработать запрос данных, либо оба равны 0, и тогда выполнение ко-

; манды закончено. Но в последнем случае дополнительно флаг Z=1. Поэтому

JNZ LOOP1

IN STATUS ; На выходе в аккумуляторе - код завершения.

Поскольку выполнение команды рано или поздно кончается, эта программа

не "зависает", если не найден сектор, как предыдущая. Кроме того, она зна-

чительно короче.

Операция записи выполняется на всех контроллерах, кроме "Криста-2",

аналогично. Приведем здесь паритетный вариант программы записи сектора.

WSECT: LXI H,BUF

DI

MVI A,WRITESCT

OUT COMMAND

IN STATUS

RRC

JNC $-3

MVI B,3

LOOP: IN STATUS

ANA B

JPO LOOP

MOV A,M ; Эти команды

OUT DATA ; на флаги процессора

INX H ; не влияют.

JNZ LOOP

DCX H ; Был лишний инкремент

IN STATUS

Программирование часов реального времени

Регистр управления A

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 |
| UIP | DV2 | DV1 | DV0 | RS3 | RS2 | RS1 | RS0 |

Бит 7 – в процессе обновления (Update-In-Progress). Когда UIP = 1, вскоре будет обновление регистров часов, когда UIP = 0 – обновление не наступит как минимум в течение 244 мкс. Все регистры часов полностью доступны, когда UIP = 0.

Бит UIP доступен только на чтение. Запись бита SET регистра управления B запрещает любые обновления регистров и очищает бит UIP.

Биты 6, 5 и 4 – DV2, DV1, DV0. Эти три бита используются для включения и выключения внутреннего генератора и сброса схемы счётчика. Значение 010 единственное, когда генератор включен и часы считают время. Значения 11x включают генератор, но удерживают цепи счётчика в состоянии сброса. После записи 010, следующее обновление регистров наступит через 500 мс.

Биты 3 – 0 – биты делителя

Bits 3 to 0: Rate Selector (RS3, RS2, RS1, RS0).

These four rate-selection bits select one of the 13 taps

on the 15-stage divider or disable the divider output.

The tap selected can be used to generate an output

square wave (SQW pin) and/or a periodic interrupt. The

user can do one of the following:

1) Enable the interrupt with the PIE bit;

2) Enable the SQW output pin with the SQWE bit;

3) Enable both at the same time and the same rate;

or

4) Enable neither.

Table 3 lists the periodic interrupt rates and the squarewave

frequencies that can be chosen with the RS bits.

than tBUC allow valid time and date information to be

reached at each occurrence of the periodic interrupt.

SELECT BITS

REGISTER A

RS3 RS2 RS1 RS0

tPI PERIODIC

INTERRUPT

RATE

SQW OUTPUT

FREQUENCY

0 0 0 0 None None

0 0 0 1 3.90625ms 256Hz

0 0 1 0 7.8125ms 128Hz

0 0 1 1 122.070μs 8.192kHz

0 1 0 0 244.141μs 4.096kHz

0 1 0 1 488.281μs 2.048kHz

0 1 1 0 976.5625μs 1.024kHz

0 1 1 1 1.953125ms 512Hz

1 0 0 0 3.90625ms 256Hz

1 0 0 1 7.8125ms 128Hz

1 0 1 0 15.625ms 64Hz

1 0 1 1 31.25ms 32Hz

1 1 0 0 62.5ms 16Hz

1 1 0 1 125ms 8Hz

1 1 1 0 250ms 4Hz

1 1 1 1 500ms 2Hz

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Биты регистра A | | | | tPI периодичность прерываний | Выходная частота |
| RS3 | RS2 | RS1 | RS0 |
| 0 | 0 | 0 | 0 | Нет | Нет |
| 0 | 0 | 0 | 1 | 3,90625 мс | 256 Гц |
| 0 | 0 | 1 | 0 | 7,8125 мс | 128 Гц |
| 0 | 0 | 1 | 1 | 122,070 мкс | 8192 кГц |
| 0 | 1 | 0 | 0 | 244,141 мкс | 4096 кГц |
| 0 | 1 | 0 | 1 | 488,281 мкс | 2048 кГц |
| 0 | 1 | 1 | 0 | 976,5625 мкс | 1024 кГц |
| 0 | 1 | 1 | 1 | 1,953125 мс | 512 Гц |
| 1 | 0 | 0 | 0 | 3,90625 мс | 256 Гц |
| 1 | 0 | 0 | 1 | 7,8125 мс | 128 Гц |
| 1 | 0 | 1 | 0 | 15,625 мс | 64 Гц |
| 1 | 0 | 1 | 1 | 31,25 мс | 32 Гц |
| 1 | 1 | 0 | 0 | 62,5 мс | 16 Гц |
| 1 | 1 | 0 | 1 | 125 мс | 8 Гц |
| 1 | 1 | 1 | 0 | 250 мс | 4 Гц |
| 1 | 1 | 1 | 1 | 500 мс | 2 Гц |

These four read/write bits are not affected by RESET.

Control Register B

BIT 7 BIT 6 BIT 5 BIT 4 BIT 3 BIT 2 BIT 1 BIT 0

SET PIE AIE UIE SQWE DM 24/12 DSE

Bit 7: SET. When the SET bit is 0, the update transfer

functions normally by advancing the counts once per

second. When the SET bit is written to 1, any update

transfer is inhibited, and the program can initialize the

time and calendar bytes without an update occurring in

the midst of initializing. Read cycles can be executed in

a similar manner. SET is a read/write bit and is not

affected by RESET or internal functions of the device.

Bit 6: Periodic Interrupt Enable (PIE). The PIE bit is a

read/write bit that allows the periodic interrupt flag (PF) bit

in Register C to drive the IRQ pin low. When the PIE bit is

set to 1, periodic interrupts are generated by driving the

IRQ pin low at a rate specified by the RS3–RS0 bits of

Register A. A 0 in the PIE bit blocks the IRQ output from

being driven by a periodic interrupt, but the PF bit is still

set at the periodic rate. PIE is not modified by any internal

device functions, but is cleared to 0 on RESET.

Bit 5: Alarm Interrupt Enable (AIE). This bit is a

read/write bit that, when set to 1, permits the alarm flag

(AF) bit in Register C to assert IRQ. An alarm interrupt

occurs for each second that the three time bytes equal

the three alarm bytes, including a don’t-care alarm

code of binary 11XXXXXX. The AF bit does not initiate

the IRQ signal when the AIE bit is set to 0. The internal

functions of the device do not affect the AIE bit, but is

cleared to 0 on RESET.

Bit 4: Update-Ended Interrupt Enable (UIE). This bit is

a read/write bit that enables the update-end flag (UF)

bit in Register C to assert IRQ. The RESET pin going

low or the SET bit going high clears the UIE bit.

The internal functions of the device do not affect the

UIE bit, but is cleared to 0 on RESET.

Bit 3: Square-Wave Enable (SQWE). When this bit is

set to 1, a square-wave signal at the frequency set by

the rate-selection bits RS3–RS0 is driven out on the SQW

pin. When the SQWE bit is set to 0, the SQW pin is held

low. SQWE is a read/write bit and is cleared by RESET.

SQWE is low if disabled, and is high impedance when

VCC is below VPF. SQWE is cleared to 0 on RESET.

Bit 2: Data Mode (DM). This bit indicates whether time

and calendar information is in binary or BCD format.

The DM bit is set by the program to the appropriate format

and can be read as required. This bit is not modified

by internal functions or RESET. A 1 in DM signifies

binary data, while a 0 in DM specifies BCD data.

Bit 1: 24/12. The 24/12 control bit establishes the format

of the hours byte. A 1 indicates the 24-hour mode

and a 0 indicates the 12-hour mode. This bit is

read/write and is not affected by internal functions or

RESET.

Bit 0: Daylight Saving Enable (DSE). This bit is a

read/write bit that enables two daylight saving adjustments

when DSE is set to 1. On the first Sunday in

April, the time increments from 1:59:59 AM to 3:00:00

AM. On the last Sunday in October when the time first

reaches 1:59:59 AM, it changes to 1:00:00 AM. When

DSE is enabled, the internal logic test for the first/last

Sunday condition at midnight. If the DSE bit is not set

when the test occurs, the daylight saving function does

not operate correctly. These adjustments do not occur

when the DSE bit is 0. This bit is not affected by internal

functions or RESET.

BIT 7 BIT 6 BIT 5 BIT 4 BIT 3 BIT 2 BIT 1 BIT 0

SET PIE AIE UIE SQWE DM 24/12 DSE

Real-Time Clock

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 17

Bit 7: Interrupt Request Flag (IRQF). This bit is set to

1 when any of the following are true:

PF = PIE = 1

AF = AIE = 1

UF = UIE = 1

Any time the IRQF bit is 1, the IRQ pin is driven low.

This bit can be cleared by reading Register C or with a

RESET.

Bit 6: Periodic Interrupt Flag (PF). This bit is readonly

and is set to 1 when an edge is detected on the

selected tap of the divider chain. The RS3 through RS0

bits establish the periodic rate. PF is set to 1 independent

of the state of the PIE bit. When both PF and PIE

are 1s, the IRQ signal is active and sets the IRQF bit.

This bit can be cleared by reading Register C or with a

RESET.

Bit 5: Alarm Interrupt Flag (AF). A 1 in the AF bit indicates

that the current time has matched the alarm time.

If the AIE bit is also 1, the IRQ pin goes low and a 1

appears in the IRQF bit. This bit can be cleared by

reading Register C or with a RESET.

Bit 5: Update-Ended Interrupt Flag (UF). This bit is

set after each update cycle. When the UIE bit is set to

1, the 1 in UF causes the IRQF bit to be a 1, which

asserts the IRQ pin. This bit can be cleared by reading

Register C or with a RESET.

Bits 3 to 0: Unused. These bits are unused in Register

C. These bits always read 0 and cannot be written.

BIT 7 BIT 6 BIT 5 BIT 4 BIT 3 BIT 2 BIT 1 BIT 0

IRQF PF AF UF 0 0 0 0

Control Register C

BIT 7 BIT 6 BIT 5 BIT 4 BIT 3 BIT 2 BIT 1 BIT 0

VRT 0 0 0 0 0 0 0

Control Register D

Bit 7: Valid RAM and Time (VRT). This bit indicates

the condition of the battery connected to the VBAT pin.

This bit is not writeable and should always be 1 when

read. If a 0 is ever present, an exhausted internal lithium

energy source is indicated and both the contents of

the RTC data and RAM data are questionable. This bit

is unaffected by RESET.

Bits 6 to 0: Unused. The remaining bits of Register D

are not usable. They cannot be written and they always read 0.

Структура жёсткого диска

Первый сектор жёсткого диска содержит программу загрузчика (необязательно) и таблицу разделов MBR. Сектор заканчивается сигнатурой AA55h.

Для файловой системы FAT16 основные данные на диске распределены следующим образом:

|  |  |  |
| --- | --- | --- |
| Сектор LBA | Смещение | Описание |
| 0000 | 01С6 | **VolShift** Начало первого тома на диске (в секторах) |
| 01CE | **VolSize** Размер первого тома (в секторах) |
| VolShift | 000B | **SectSize** Размер сектора в байтах |
| 000D | **ClasterSize** Размер кластера в секторах |
| 000E | **FAT\_Shift** Положение первой таблицы FAT (с учётом начала тома) |
| 0010 | **FAT\_Count** Количество копий FAT |
| 0011 | **DirSize** Количество элементов корневого каталога |
| 0016 | **FAT\_Size** Размер каждой таблицы FAT в секторах |
| 0036 | Идентификатор файловой системы “FAT16 “ |
| FAT\_Shift | 0000 | Первая таблица FAT |
| FAT\_Size | Вторая таблица FAT |
| FAT\_Shift + FAT\_Count \* FAT\_Size | 0000 | **DirShift** Корневой каталог |
| DirSize \* 20 | **DataShift** Начало кластеров данных (Нулевой кластер адресует начало корневого каталога) |

Далее приведено подробное описание отдельных структур на диске.

Таблица разделов MBR

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Структура главной загрузочной записи (MBR) | | | | | | | |
| Смещение | Размер | Описание | | | | | |
| 0000h | 1BEh | Код загрузчика | | | | | |
| 01BEh | 10h | Таблица разделов | Раздел 1 | Смещение | Размер | Описание | |
| 00h | 01h | Признак активности раздела (80h – активен, 00h – не активен) | |
| 01h | 01h | Начало раздела | Номер головки |
| 02h | 01h | Номер сектора (биты 0 – 5),  номер дорожки (биты 6, 7) |
| 03h | 01h | Номер дорожки (биты 8 и 9 – в предыдущем байте) |
| 04h | 01h | Код типа раздела | |
| 05h | 01h | Конец раздела | Номер головки |
| 06h | 01h | Номер сектора (биты 0 – 5),  номер дорожки (биты 6, 7) |
| 07h | 01h | Номер дорожки (биты 8 и 9 – в предыдущем байте) |
| 08h | 04h | Смещение первого сектора (LBA) | |
| 0Ch | 04h | Количество секторов раздела (LBA) | |
| 01CEh | 10h | Раздел 2 | Описание второго раздела | | | |
| 01DEh | 10h | Раздел 3 | Описание третьего раздела | | | |
| 01EEh | 10h | Раздел 4 | Описание четвёртого раздела | | | |
| 01FEh | 02h | 55h, AAh | | Сигнатура | | | |

Блок параметров диска FAT

|  |  |  |
| --- | --- | --- |
| Смещ. | Пример | Описание |
| 00–02 | EB, 3E, 90 | Ассемблерная команда перехода к загрузочному коду |
| 03–0A | “MSDOS622” | Имя OEM в кодировке ASCII |
| 0B,0C | 0200 | Количество байтов в секторе. Допустимые значения – 512, 1024, 2048 и 4096 |
| 0D | 10 | Количество секторов в кластере (блоке данных). Допустимые значения задаются степенями 2, но размер кластера не должен превышать 32 Кбайт |
| 0E,0F | 0001 | Размер зарезервированной области в секторах (от начала раздела до таблицы FAT) |
| 10 | 02 | Количество копий FAT. Обычно в системе хранятся две копии, но, по документации Microsoft, для устройств малой ёмкости допускается хранение только одной копии |
| 11,12 | 0200 | Максимальное количество файлов в корневом каталоге для FAT12 и FAT16. В FAT32 поле равно 0, а в FAT16 оно обычно равно 512 |
| 13,14 | 0000 | 16-разрядное количество секторов в файловой системе. Если количество секторов не может быть представлено двухбайтовой величиной, позднее в структуре данных следует альтернативное четырёхбайтовое поле (а двухбайтовое должно быть равно нулю) |
| 15 | F8 | Тип носителя. Согласно документации Microsoft, для стационарных дисков используется значение 0F8h, а для съемных – 0F0h |
| 16,17 | 00F6 | 16-разрядный размер (в секторах) каждой копии FAT в FAT12 и FAT16. В FAT32 поле равно 0 |
| 18,19 | 003F | Количество секторов в дорожке |
| 1A,1B | 0010 | Количество головок |
| 1C-1F | 0000003F | Количество секторов перед началом раздела |
| 20-23 | 000F5961 | 32-разрядное количество секторов в файловой системе. Либо это поле, либо 16-разрядное поле (19-20) должно быть равно 0 |
| 24 | 80 | Номер диска BIOS INT13h |
| 25 | 00 | Не используется |
| 26 | 29 | Расширенная сигнатура, которая показывает, действительны ли следующие три значения. Сигнатура равна 0x29 |
| 27-2A | 1704456A | Серийный номер тома, в некоторых версиях Windows вычисляется на основании даты и времени создания |
| 2B-35 | “LABEL  VOL ” | Метка тома в кодировке ASCII. Выбирается пользователем при создании файловой системы |
| 36-3D | “FAT16 “ | Метка типа файловой системы в кодировке ASCII. Стандартные значения – «FAT», «FAT12» и «FAT16», но ни одно из них не является обязательным |
| 3E-1FD | xx…xx | Не используется |
| 1FE-1FF | AA55h | Сигнатура системного сектора. |

Таблицы FAT располагаются в следующих секторах за блоком параметров.

Команды жесткого диска с AT-интерфейсом

Назначение битов регистра команд (CR)

| Код (Hex) | Команда | Описание |
| --- | --- | --- |
| 1X | Recalibrate | Головка отводится на цилиндр 0 |
| 2X | Read Sector | Группа команд: читает от 1 до 256 секторов, начиная с заданного сектора |
| 20 | Read Sector with retry | Читает сектора; при ошибке чтения об ошибке сообщается только после нескольких безуспешных повторов чтения |
| 21 | Read Sector no retry | Читает сектора; об ошибке сообщается сразу при появлении ошибки чтения |
| 22 | Read Sector long with retry | Читает сектора; после байтов данных передаются также байты коррекции ошибки для текущего сектора (используется для тестирования) |
| 23 | Read Sector long no retry | Как и команда 22h, но без повторного чтения |
| 3X | Write Sector | Группа команд: записывает от 1 до 256 секторов с заданного сектора. Команды 30h, 31h, 32h и 33h как и при "Read Sector ..." |
| 4X | Read Verify Sector | Группа команд: проверяют корректную читаемость заданных секторов |
| 40 | Read Verify Sector with retry | Проверяет читаемость с повторением чтения при ошибках чтения |
| 41 | Read Verify Sector no retry | Проверяет читаемость без повторений чтения |
| 50 | Format Track | Форматирует заданную дорожку |
| 7X | Seek | Переводит головку на заданную дорожку |
| 90 | Exec Drive | Hакопитель начинает собственный тест |
| 91 | Set Drive Parameter | Передает ожидаемые параметры накопителя встроенному контроллеру |
| C4\* | Read Multiple | Соответствует команде 20h, но несколько секторов передаются как блок |
| C5\* | Write Multiple | Соответствует команде 30h, но несколько секторов передаются как блок |
| C6\* | Set Multiple | Передает число секторов в блоке для команд C4h и C5h в регистр 1F2h |
| E4 | Read Sect Buffer | Читает 512 байт из буфера сектора |
| E8 | Write Sect Buffer | Записывает 512 байт в буфер сектора |
| EC | Identify Drive | Читает информацию о накопителе (модель, версия, серийный номер, тип контроллера, число цилиндров, головок, секторов на дорожку и т.д.) |
| EF\* | Cache On/Off | Включение/выключение кэш-памяти контроллера: "55h" в 1F1h - кэш выключен, "AAh" в 1F1h - кэш включен |
| EX\* | Power Commands | Команды для снижения мощности |
| E0\* | Standby Mode | Выключает двигатель шпинделя, при доступе снова автоматически включается |
| E1\* | Idle Mode | Выключает двигатель шпинделя |
| E2\* | Auto Power Down | Выключает двигатель шпинделя, после доступа снова автоматически выключается, время устанавливается шагами по 5 секунд (1F2h) |
| E3\* | Auto Power Down | Включает двигатель шпинделя и далее как команда E2h |
| E5\* | Read Power Mode | Читает в 1F2h состояние двигателя шпинделя |
| E6\* | Sleep Mode | Выключает накопитель, включение только через сброс (программный или аппаратный) |
| FX\* | Power Save | Команды F8h, F9h, FAh, FBh, FDh как и команды E0h...E5h, но время устанавливается шагами по 0,1 секунды |
| F0\*\* | Set Configuration | Кэш-стратегия и обработка ошибок |
| F0\*\* | Read Configuration | Читает текущую конфигурацию |
| F0\*\* | Defect List | Читает список дефектных мест |

\* - Специальные команды, зависящие от изготовителя.

\*\* - Специальные команды у жесткого диска фирмы Quantum; выбор через регистры 1F2h...1F6h.