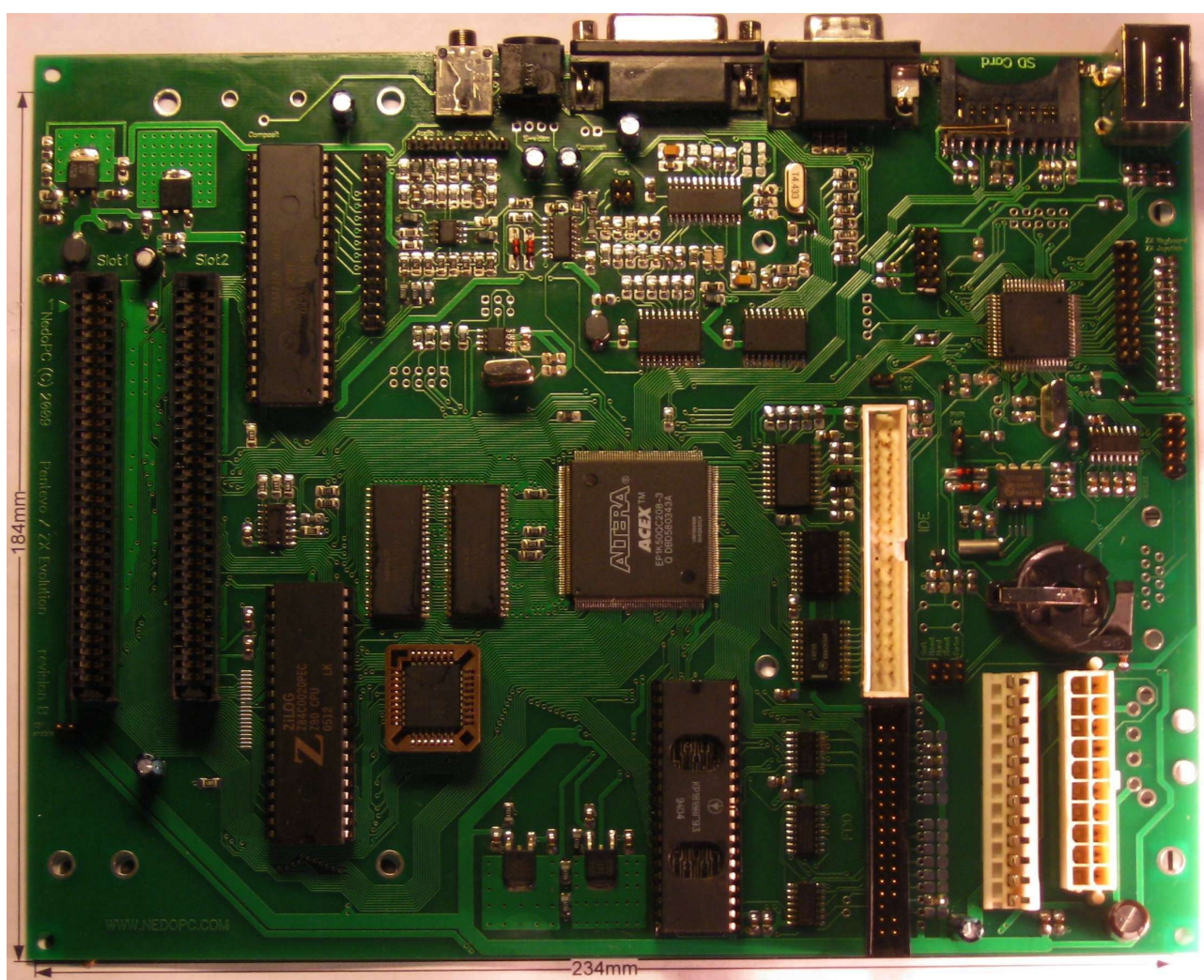


# Базовая конфигурация для ZX Evolution.

## Руководство пользователя.



(версия от 01.03.2012)

[www.nedopc.com](http://www.nedopc.com)

## Оглавление

1 Введение.....	4
2 Управляющие контакты компьютера.....	5
2.1 Hard reset (полный сброс).....	5
2.2 Soft reset (мягкий сброс).....	5
2.3 NMI (подача немаскируемого прерывания на процессор Z80).....	5
3 Дополнительные функции PS2 клавиатуры.....	6
3.1 Мягкий сброс компьютера (soft reset).....	6
3.2 Полный сброс компьютера (hard reset).....	6
3.3 Переключение режима видеовыхода (TV/VGA).....	6
3.4 Переключение режима аудиовыхода (beeper/tapeout).....	6
3.5 Подача NMI (немаскируемого прерывания) на процессор Z80.....	7
3.6 Регулировка разрешения PS/2 мыши.....	7
4 Архитектура: общие положения.....	8
5 Архитектура: управление памятью.....	9
6 Архитектура: управление скоростью процессора.....	13
7 Архитектура: видеорежимы.....	13
7.1 Выбор видеорежимов.....	13
7.2 Палитра.....	14
7.3 Бордюр.....	15
7.4 Адресация видеорежимов.....	15
7.4.1 Стандартный ZX-режим.....	15
7.4.2 ZX hardware multicolor.....	15
7.4.3 256x192 16 цветов.....	15
7.4.4 640x200 hardware multicolor.....	16
7.4.5 320x200 16 цветов.....	16
7.4.6 Текстовый режим 80x25.....	16
7.4.7 Загрузка шрифта для текстового режима.....	17
8 Архитектура: порты чтения конфигурации и NMI.....	18
8.1 Порт #BF (чтение).....	18
8.2 Порт #xxBE (чтение).....	18
8.3 Порт #BE (запись).....	19
8.4 Вход в NMI.....	19
8.5 Выход из NMI.....	20
9 Архитектура: порты ввода-вывода.....	20
9.1 Порт #FE.....	20
9.2 Порт COVOX (#xxFB, WO always).....	21
9.3 Порты AY/YM.....	21
9.4 Порт Kempston Joystick.....	21
9.5 Порты Kempston Mouse.....	21
9.6 Порты управления часами и NVRAM.....	22
9.7 Чтение версии базовой конфигурации/бутлоадера.....	23
9.8 Доступ к PS/2 клавиатуре.....	24
9.9 Доступ к SD-карте.....	25
9.10 Доступ к IDE-устройствам.....	26

## *Базовая конфигурация для ZX Evolution*

9.11	Порты контроллера дискового.....	28
9.12	Порт RS232 (коммуникационный интерфейс).....	29
9.12.1	Регистр данных (DAT).....	30
9.12.2	Регистры делителя (DLL,DLM).....	30
9.12.3	Регистр управления FIFO (FCR).....	31
9.12.4	Регистр управления линией (LCR).....	31
9.12.5	Регистр управления модемом (MCR).....	31
9.12.6	Регистр статуса линии (LSR).....	32
9.12.7	Регистр статуса модема (MSR).....	32
9.12.8	Пользовательский регистр (SPR).....	32
10	История версий.....	33
10.1	Версия 29.09.2011.....	33
10.2	Версия 11.06.2011.....	33
10.3	Версия 11.05.2011.....	33
10.4	Версия 26.04.2011.....	33
10.5	Версия 30.03.2011.....	33
10.6	Версия 07.12.2010.....	33
10.7	Версия 29.11.2010.....	34
10.8	Версия 17.10.2010.....	34
10.9	Версия 30.03.2010.....	34
10.10	Версия 28.03.2010.....	34
10.11	Версия 24.03.2010.....	34
10.12	Версия 10.03.2010.....	34
10.13	Версия 04.02.2010.....	35

## **1 Введение**

Под базовой конфигурацией компьютера ZX Evolution подразумевается конфигурация, которая поставляется и поддерживается производителем. Производитель осуществляет поддержку и развитие этой конфигурации и исправление найденных в ней ошибок. Группа NedoPC не поддерживает конфигурации сторонних разработчиков, в случае проблем с такими конфигурациями, обращайтесь непосредственно к их разработчикам.

Базовая конфигурация состоит из следующих компонент:

- Конфигурация для FPGA EP1K50QC208;
- Программа для микроконтроллера ATMEGA128.

Конфигурация поставляется в виде бинарного файла (zxevo\_fw.bin), который можно загрузить в микроконтроллер ATMEGA128 с помощью BOOTLOADER<sup>1</sup>. Обновление конфигураций, также производится с помощью BOOTLOADER.

Конфигурация FPGA интегрирована в программу микроконтроллера. Конфигурирование FPGA производится при включении или железном сбросе компьютера.

Изменение версии базовой конфигурации подразумевает изменение любой из или обеих компонент.

---

<sup>1</sup> BOOTLOADER — базовый загрузчик. Неизменяемая часть памяти микроконтроллера ATMEGA128, которая предназначена для загрузки или обновления памяти программ ATMEGA128.

## 2 Управляющие контакты компьютера

Управляющие группы контактов компьютера, предназначены для управления работой компьютера с помощью подсоединенных к ним элементов управления (например кнопок на корпусе компьютера).

### 2.1 Hard reset (полный сброс)

Группа контактов «Hard reset» осуществляет полный сброс компьютера (с перезапуском прошивки микроконтроллера и перепрограммированием FPGA) при замыкании контактов. Функционирование не зависит от используемой конфигурации компьютера.

Рекомендуется при использовании стандартного (AT или ATX) корпуса подсоединить к кнопке «Reset» корпуса.

**Внимание!** Hard Reset *не обеспечивает* сохранность содержимого памяти компьютера (например, RAM-дисков или страничек с исходными текстами в аласме).

### 2.2 Soft reset (мягкий сброс)

Функции группа контактов «Soft reset» зависят от конфигурации.

В базовой конфигурации на эту группу контактов возложены две функции:

- мягкий сброс компьютера (сброс процессора и внутренних регистров fpga);
- включение/выключение ATX блока питания.

Рекомендуется при использовании стандартного ATX корпуса подсоединить к кнопке «Power» корпуса. В этом случае включение компьютера осуществляется однократным нажатием кнопки. Отключение компьютера осуществляется долговременным нажатием в течении 5 секунд.

Кратковременное замыкание контактов будет осуществлять мягкий сброс компьютера и установку нулевой страницы ROM.

Мягкий сброс обеспечивает сохранность содержимого памяти компьютера.

### 2.3 NMI (подача немаскируемого прерывания на процессор Z80)

Функции группы контактов «NMI» зависят от конфигурации.

В базовой конфигурации на эту группу контактов возложены функции подачи немаскируемого прерывания процессор на процессор Z80 (аналогично описанному в 3.5).

**Внимание!** Для revision A и B плат эта группа контактов называется «Turbo», но используется как контакты для управления NMI.

### 3 Дополнительные функции PS2 клавиатуры

При использовании PS2 клавиатуры с компьютером ZX Evolution можно использовать дополнительные возможности по управлению компьютером:

- мягкий сброс компьютера;
- полный сброс компьютера;
- переключение режима видеовыхода (TV/VGA);
- переключение режима аудиовыхода (beeper/tapeout);
- подача NMI на процессор Z80.

#### 3.1 Мягкий сброс компьютера (soft reset)

Мягкий сброс компьютера осуществляется с помощью кратковременного (менее 5 секунд) нажатия клавиши «F12».

#### 3.2 Полный сброс компьютера (hard reset)

Полный сброс компьютера с клавиатуры можно осуществить двумя способами:

- Длительным нажатием клавиши «F12» (длительностью 5 сек), отключение питания в этом случае не производится;
- Нажатием комбинации «Ctrl-Alt-Del», отключение питания в этом случае не производится.

#### 3.3 Переключение режима видеовыхода (TV/VGA)

Так как используется объединенный видеовыход (в связи с ограничением количества ног микросхемы EP1K50QC208), то необходимо установить режим видеовыхода:

- TV – скандалер отключен, видеовыход работает с телевизионной разверткой;
- VGA – включен скандалер, видеовыход работает с VGA разверткой.

Переключение режима видеовыхода (TV/VGA) осуществляется с помощью клавиши «Scroll Lock». Индикация режима осуществляется с помощью светодиода «Scroll Lock» клавиатуры. Светящееся состояние светодиода соответствует VGA режиму компьютера, отсутствие свечения — TV режиму.

**Внимание!** Чтобы состояние режима сохранялось при выключении компьютера, необходимо установить батарейку.

#### 3.4 Переключение режима аудиовыхода (beeper/tapeout)

Так как используется объединенный аудиовыход (в связи с ограничением количества ног

микросхемы EP1K50QC208), то необходимо установить режим аудиовыхода:

- beeper – на аудиовыход подается сигнал соответствующий выходу beeper;
- tapeout – на аудиовыход подается сигнал соответствующий выходу tapeout.

Переключение режима аудиовыхода (beeper/tapeout) осуществляется с помощью клавиши «Num Lock». Индикация режима осуществляется с помощью светодиода «Num Lock» клавиатуры. Светящееся состояние светодиода соответствует tapeout режиму аудиовыхода, отсутствие свечения — beeper режиму.

**Внимание!** Чтобы состояние режима сохранялось при выключении компьютера, необходимо установить батарейку.

### 3.5 Подача NMI (немаскируемого прерывания) на процессор Z80

Подача NMI на процессор Z80 осуществляется двумя способами:

- с помощью клавиши «Print Screen» PS/2 клавиатуры
- Переводом бита 3 порта #xxBF из 1 в 0

### 3.6 Регулировка разрешения PS/2 мыши

Регулировка разрешения PS/2 мыши возможна только для мышей, которые поддерживают эту функцию. Как правило это оптические и лазерные мыши.

Диапазон поддерживаемых разрешений PS/2 мыши:

Значение	Разрешение
0	1 отсчет на мм
1	2 отсчета на мм
2	4 отсчета на мм
3	8 отсчетов на мм

Регулировка происходит одновременным нажатием двух клавиш PS/2 мыши (левой и правой) и управляющей клавиши на PS/2 клавиатуре:

- keypad '\*' - сброс значения разрешения в 0 (по умолчанию);
- keypad '+' - увеличение значения разрешения на 1;
- keypad '-' - уменьшение значения разрешения на 1.

**Внимание!** Чтобы значение разрешения сохранялось при выключении компьютера, необходимо установить батарейку.

## 4 Архитектура: общие положения

ZX Evolution имеет на борту 4 мегабайта ОЗУ и 512 килобайт ПЗУ, которые доступны через следующие модели управления памятью: ZX Spectrum 48k, ZX Spectrum 128k (порт **#7FFD**), pentagon 1024k (адресация через порт **#7FFD**), ATM2 (адресация мегабайта ОЗУ), а также дополнительный режим адресации всех 4 мегабайт ОЗУ. Все 512 килобайт ПЗУ адресуются по модели ATM2, при этом дополнительно существует возможность записи в ПЗУ, которое реализовано на микросхеме flash-памяти.

В архитектуре контроллеров дисководов beta-disk появляется понятие теневого порта, доступных только при определённых условиях. В компьютере ATM2 появляются дополнительный набор теневого порта (используемых, в частности, для управления памятью) и дополнительные способы включения таких портов (в частности, режим удержания теневого режима при выполнении из ОЗУ). Однако, в ATM2 остаётся однозначная связь между теньевым режимом и включением ПЗУ TR-DOS. В архитектуре ZX Evolution при полной поддержке режимов ATM2 появляется возможность удержания теневого режима портов *без* включения ПЗУ TR-DOS.

Концепция портов ввода-вывода для ZX Evolution существенно отличается от одной в других спектрум-совместимых компьютерах. Прежде всего, дешифрация всегда происходит как минимум по 8 младшим битам. Существенно отличается логика блокирования портов картами ZX-Bus: если в других спектрум-совместимых компьютерах карты блокируют порты материнской платы (и друг друга, при наличии нескольких слотов), то в ZX Evolution, наоборот, все порты, присутствующие на материнской плате, не 'доходят' до карт ZX-Bus. Механизм IORQGE используется картами лишь для блокирования друг друга (в порядке расположения в слотах). Отсюда следует важный вывод — любые карты, дублирующие какую-либо функциональность материнской платы (имеющие порты, совпадающие с существующими в ZX Evolution), такие как контроллеры клавиатуры, мышки, IDE, SD-карт и т. д., работать *не будут*.

В дальнейшем для каждого порта указываются режим его видимости (теньевой доступен всегда: **«shadow»**, **«noshad»**, **«always»**), доступность его для чтения или записи (только чтение, только запись, чтение и запись: **«RO»**, **«WO»**, **«RW»**). Также указывается адрес порта либо в виде **«#xxFE»** (дешифрация только по младшим 8 битам), либо в виде **«#7FFD»** (дешифрация не только по младшим 8 битам). Первый вариант указания адреса порта применяется также и в случае, если какие-либо адресные линии A8..A15 используются как информационные входы (например, набор портов для переключения страниц в 16k окнах процессора: **#3FF7**, **#7FF7**, **#BFF7**, **#FFF7**), при этом в описании порта указываются конкретные значения старшего байта адреса, которые только и следует использовать. В случае указания 16-битного адреса порта, конкретные биты старшего адресного байта, используемые при дешифрации, *намеренно не указываются* и могут изменяться от прошивки к прошивке.



## 5 Архитектура: управление памятью.

Управление памятью в ZX Evolution во многом схоже с управлением памятью в ATM2. Принципиальные изменения касаются возможности адресации мегабайта ОЗУ через порт #7FFD (модель pentagon 1024k) и адресации 4 мегабайт ОЗУ.

В каждое 16k окно процессора можно произвольно включать любую из 16k страниц ОЗУ или ПЗУ. Существует 2 независимых карты памяти, которые переключаются битом 4 в порте #7FFD. В режиме совместимости с ZX Spectrum 128k или pentagon 1024k эти две карты памяти программируются идентично за исключением номера включенной страницы ПЗУ в нулевом окне процессора, чтобы данный бит работал, как в ZX Spectrum 128k. Отображение страниц памяти в каждом окне процессора программируется независимо через специальные теневые порты. Кроме того, для каждого такого окна существует опция подмены младших 3 или 6 бит номера включенной страницы ОЗУ данными из соответствующих бит порта #7FFD: таким образом обеспечивается совместимость с режимами ZX Spectrum 128k и pentagon 1024k. При отображении же страницы ПЗУ существует опция подмены младшего бита номера страницы на сигнал DOSEN (сигнал включения TRDOS). Включение данного сигнала («переход в TR-DOS») происходит при исполнении кода из ПЗУ в каком-либо окне процессора по смещению #3Dxx в этом окне при установленном бите D4 в порте #7FFD. Выключение происходит при выполнении кода из ОЗУ («выход из TR-DOS»). Кроме того, при помощи соответствующего теневого порта можно запретить выключение сигнала DOS, таким образом оставляя теневые порты доступными при исполнении из ОЗУ. Вдобавок, существует возможность включения теневого порта безотносительно сигнала DOS.

Порт	Описание
#xxBF RW always	<p>Включение теневого режима портов; разрешение записи в ПЗУ.</p> <p>7..4: устанавливайте в 0 для совместимости</p> <p>3: Генерация NMI для процессора. NMI выдаётся при переходе бита из 1 в 0.</p> <p>2: если 1, то разрешается загрузка в font rom. 0 после сброса.</p> <p>1: если 1, то запись в ПЗУ разрешена. 0 после сброса.</p> <p>0: если 1, то включены теневые порты. 0 после сброса.</p> <p>Бит 0 этого порта имеет приоритет над другими способами включения теневого режима портов: теновый режим включается безусловно при записи 1 в этот бит. При записи 0 включение теневого режима управляется другими способами, например портом #xx77 или переходом в TR-DOS. Данный бит <i>не включает</i> TR-DOS.</p> <p>Данный порт <i>отсутствует</i> в ATM2.</p> <p>При чтении: биты 2:0 — последнее записанное значение, биты 7:3 — не определены.</p>

*Базовая конфигурация для ZX Evolution*

<i><b>Порт</b></i>	<i><b>Описание</b></i>		
<b>#xx77 WO shadow</b>	<p>Включение теневого режима портов; разрешение работы диспетчера памяти.</p> <table border="1"> <tr> <td>А8: если 0, то выключение диспетчера памяти. В каждое окно процессора устанавливается последняя страница ПЗУ. 0 после сброса.</td><td>А9: если 0, то “наильно” включение TR-DOS и теневого портов. 0 после сброса.</td></tr> </table> <p><i>Внимание!</i> Остальные биты (как биты данных, так и биты старшего байта адреса) этого write-only порта используются для других целей и описаны в других местах данной документации.</p> <p>Включение TR-DOS и теневого портов битом А9 имеет приоритет над включением TR-DOS и теневого портов исполнением со смещения #3Dxx страницы ПЗУ при установленном бите D4 порта <b>#7FFD</b> («переход в TR-DOS»). Когда А9=0, то теньовые порты и TR-DOS включены независимо от места исполнения процессора (в т.ч. и из ОЗУ).</p> <p>Допускается использовать следующие адреса этого порта:</p> <p><b>#FD77</b> — включение теневого портов и TR-DOS,</p> <p><b>#BD77</b> — то же самое, плюс разрешение перезаписи палитры (А14=0),</p> <p><b>#BF77</b> – разрешение перезаписи палитры без включения теневого портов и TR-DOS</p> <p><b>#FF77</b> – выключение теневого портов и TR-DOS.</p> <p>Для установки А8 в 0 из любого из вышеуказанных адресов вычесть #100.</p>	А8: если 0, то выключение диспетчера памяти. В каждое окно процессора устанавливается последняя страница ПЗУ. 0 после сброса.	А9: если 0, то “наильно” включение TR-DOS и теневого портов. 0 после сброса.
А8: если 0, то выключение диспетчера памяти. В каждое окно процессора устанавливается последняя страница ПЗУ. 0 после сброса.	А9: если 0, то “наильно” включение TR-DOS и теневого портов. 0 после сброса.		

*Базовая конфигурация для ZX Evolution*

<i>Порт</i>	<i>Описание</i>
<b>#xFF7 WO shadow</b>	<p>Управление диспетчером страниц.</p> <p>5..0: <i>инверсные</i> биты номера страницы, включаемого в окно процессора. Доступны 16k страницы с номерами 0..63 ОЗУ (всего 1 мегабайт) и страницы 0..31 ПЗУ (всего 512 килобайт, бит 5 игнорируется).</p> <p>6: бит «gamptom»: если 0, то в данном окне процессора включается ПЗУ, если 1 — ОЗУ.</p> <p>7: бит «dos7ffd»: если 1, то:  для ОЗУ — в данном окне происходит подмена младших 3 или 6 бит (в зависимости от режима ZX Spectrum 128k или pentagon 1024k) номера страницы <i>не инверсными</i> битами из порта #7FFD.  Для ПЗУ — происходит подмена младшего бита номера страницы сигналом включения TR-DOS (1 если TR-DOS включен). Кроме того, происходит включение теневых портов и TR-DOS («вход в TR-DOS»), если в данном окне произойдёт выполнение кода со смещения #3Dxx.</p> <p>Биты A14, A15 определяют окно процессора, для которого происходит включение требуемой страницы. Следует использовать следующие адреса этого порта:  <b>#3FF7</b> — для окна #0000..#3FFF,  <b>#7FF7</b> — для окна #4000..#7FFF,  <b>#BFF7</b> — для окна #8000..#BFFF,  <b>#FFF7</b> — для окна #C000..#FFFF.</p> <p>Запись в данный порт меняет страницу в каком либо окне текущей (активной) карты памяти. При этом неактивная карта памяти остаётся неизменной. Какая карта памяти активна, задаёт бит 4 порта <b>#7FFD</b>.</p>

<i>Порт</i>	<i>Описание</i>
<b>#x7F7 WO shadow</b>	<p>Управление диспетчером страниц.</p> <p>7..0: <i>инверсные</i> биты номера страницы ОЗУ, включаемого в окно процессора. Доступны 16k страницы с номерами 0..255 (всего 4 мегабайта).</p> <p>Биты A14, A15 определяют окно процессора, для которого происходит включение требуемой страницы. Следует использовать следующие адреса этого порта:  <b>#37F7</b> — для окна #0000..#3FFF,  <b>#77F7</b> — для окна #4000..#7FFF,  <b>#B7F7</b> — для окна #8000..#BFFF,  <b>#F7F7</b> — для окна #C000..#FFFF.</p> <p>Запись в данный порт меняет страницу в каком либо окне текущей (активной) карты памяти. При этом неактивная карта памяти остаётся неизменной. Какая карта памяти активна, задаёт бит 4 порта <b>#7FFD</b>.</p> <p><i>Внимание!</i> При записи в этот порт остаётся неизменным режим подмены младших бит номера страницы ОЗУ (бит «dos7ffd» порта <b>#xFF7</b>). Однако, запись в <b>#x7F7</b> безусловно включает ОЗУ в окне (очищает бит «gamptom» порта <b>#xFF7</b>).</p> <p>Данный порт <i>отсутствует</i> в ATM2.</p>

## Базовая конфигурация для ZX Evolution

Порт	Описание				
#7FFD WO always	В режиме ZX Spectrum 128k:				
	6..7: не исп.	5: при записи 1 блокируется дальнейшая запись в порт #7FFD, до перехода в режим pentagon 1024k или до сброса. 0 после сброса.	4: выбор карты памяти (см. описание портов #xxF7). 0 после сброса.	3: выбор экрана. 0 после сброса.	2..0: замещают младшие 3 бита номера страницы, если такое замещение разрешено для какого-либо окна (см. описание портов #xxF7). 0 после сброса.
	В режиме pentagon 1024k:				
	7..5, 2..0: замещают при необходимости младшие 6 бит номера страницы.		4: выбор карты памяти.	3: выбор экрана.	

Программой начальной загрузки, хранящейся в ПЗУ ZX Evolution (reset service) производится настройка карт памяти таким образом, что бит 4 порта #7FFD работает как выбор страницы ПЗУ в окне #0000..#3FFF (basic128/basic48), биты 2..0 (режим ZX Spectrum 128k) или 7..5,2..0 (режим pentagon 1024k) — замещают соответствующие биты номера страницы в окне #C000..#FFFF (более старшие биты номера страницы при этом — 0), в остальных двух окнах выставлены страницы, соответствующие конфигурации памяти ZX Spectrum 128k.

Порт	Описание	
#EFF7 WO noshad	2: при выставлении в 1 — режим ZX Spectrum 128k, иначе — режим pentagon 1024k. Значение после сброса — 0.	3: При выставлении в 1 в окно #0000..#3FFF принудительно включается нулевая страница ОЗУ. Этот бит <i>имеет приоритет</i> над всеми другими способами переключать страницы памяти в данном окне. Значение после сброса — 0.
	<b>Внимание!</b> Остальные биты этого write-only порта используются для других целей и описаны в других местах данной документации.	

## 6 Архитектура: управление скоростью процессора

Реализованы следующие скорости процессора:

- 3.5 МГц (nowait)
- 7.0 МГц (nowait)
- 14.0 МГц (with wait)

В режиме 14 МГц эффективная скорость процессора в 1.66 раз больше, чем в режиме 7.0 МГц (по замерам программы turbotest v1.0).

Для переключения скорости процессора используются биты в портах:

Порт #xx77, бит 3	Порт #EFF7, бит 4	Частота процессора, МГц
0	1	3.5
0	0	7.0
1	1	14.0
1	0	Запрещённая комбинация, последствия непредсказуемы

После сброса процессор работает на частоте 7.0 МГц.

## 7 Архитектура: видеорежимы

ZX Evolution поддерживает следующие видеорежимы:

- Стандартный ZX-режим 6912 байт
- ZX hardware multicolor, отличающийся от стандартного ZX-режима наличием отдельного байта атрибута на каждый байт пикселей
- 256x192 16 цветов по стандарту Alone Coder'a
- 320x200 16 цветов по стандарту ATM
- 640x200 hardware multicolor по стандарту ATM
- Текстовый режим 80x25 по стандарту ATM

Кроме того, поддерживается палитра по стандарту ATM.

### 7.1 Выбор видеорежимов

Выбор видеорежима осуществляется комбинацией бит в портах #EFF7 (noshad, WO) и #xx77 (shadow, WO) в соответствии с таблицей:

Порт #EFF7, бит 3	Порт #EFF7, бит 4	Порт #xx77, бит 3	Видеорежим
-------------------	-------------------	-------------------	------------

## Базовая конфигурация для ZX Evolution

бит 0	бит 5	биты 2..0	
0	0	011	Стандартный ZX-режим
0	1	011	ZX hardware multicolor
1	0	011	256x192 16 цветов
0	0	010	640x200 hardware multicolor
0	0	000	320x200 16 цветов
0	0	110	Текстовый режим 80x25

Все другие комбинации указанных бит являются невалидными, поведение компьютера при установке таких комбинаций не определено и может измениться в последующих версиях прошивки FPGA.

## 7.2 Палитра

Палитра задаёт отображение 16 доступных для каждого видеорежима и бордюра цветов в 64 аппаратных цвета (каждая из компонент R,G и B имеет 2-битные градации яркости, от чёрного до максимально насыщенного). При сбросе компьютера ПО в ПЗУ устанавливает палитру в режим совместимости с ZX-Spectrum, т. е. цвета, отображаемые в стандартном ZX-режиме, соответствуют цветам ZX-Spectrum.

Программное изменение палитры осуществляется следующим образом. Текущий отображаемый цвет определяет элемент палитры, который будет изменён. Для разрешения изменения палитры необходимо произвести запись в порт **#xx77 (WO, shadow)** с очищенным битом адреса A14 (*Внимание* — остальные биты (как биты данных, так и биты старшего байта адреса) этого порта используются для других целей и описаны в других местах данной документации). Новое значение элемента палитры устанавливается записью в порт **#FF (WO, shadow)**, который при очищенном бите A14 порта **#xx77** начинает *одновременно* выполнять функции порта контроллера дисководов и порта установки палитры.

Интерпретация записываемого в порт **#FF** значения:

Бит записываемого байта	Значение
0	B
1	R
2	Устанавливать в 1 для корректной работы контроллера дисководов
3	Устанавливать в 1 для корректной работы контроллера дисководов
4	G
5	b
6	r
7	g

Градации яркости компонент R, G и B для устанавливаемого цвета:  $\sim\{Rr\}$ ,  $\sim\{Gg\}$ ,  $\sim\{Bb\}$ , где  $\sim$  обозначает побитовую инверсию.

Примечание: удобно изменять палитру в момент отображения бордюра: в этом случае текущий отображаемый цвет является цветом бордюра. Для того, чтобы процесс изменения палитры не был виден на экране, его следует проводить в процедуре обработки кадрового прерывания (вариант — после команды **HALT**).

### 7.3 Бордюр

Бордюр — видимая область экрана, находящаяся за пределами пикселей, определяемых каким-либо видеорежимом. Цвет бордюра задаётся битами 2..0 записываемых значений в порты **#FE (RW, always)** или **#F6 (WO, always)**: при записи в порт **#FE** возможно установить цвета 0..7, при записи в **#F6** — 8..15.

### 7.4 Адресация видеорежимов

Для каждого видеорежима существует по два набора страниц, из которых осуществляется выборка видеоданных. Выбор набора страниц определяет бит 3 порта **#7FFD (WO, always)**. Страницы, соответствующие единичному состоянию этого бита, приведены в описаниях режимов в скобках.

#### 7.4.1 Стандартный ZX-режим

Выборка из страницы 5(7). Пиксели каждой строки состоят из бит непрерывной байтовой строки (порядок выдачи бит из отдельного байта — от старшего бита к младшему). Начальные смещения байтовых строк вычисляются из номера  $Y$  строки ( $Y=0..191$ ,  $Y=0$  для верхней строки) следующим образом:  $\text{pixoffset} = \{ Y[7:6], Y[2:0], Y[5:3] \} \ll 5$ , где  $Y[m:n]$  — выборка соответствующей битовой последовательности из величины  $Y$ ,  $\{ \}$  - конкатенация битовых последовательностей в число,  $\ll$  - сдвиг влево на указанное число бит с заполнением младших бит нулями.

Битовая карта режима делится на квадраты 8x8 пикселей, каждый такой квадрат имеет свой байтовый атрибут, расположенный по смещению  $\text{attroffset} = \#1800 + Y*32 + X$ , где  $X, Y$  — координаты квадрата 8x8 пикселей (0,0 — верхний левый квадрат, 1,0 — его сосед справа и т.д.).

Цвет пикселя, соответствующего единичному биту, определяется байтом атрибута  $\text{attr}$  как  $\{ \text{attr}[6], \text{attr}[2:0] \}$ . Цвет пикселя, соответствующего нулевому биту — как  $\{ \text{attr}[6], \text{attr}[5:3] \}$ . Установленный бит  $\text{attr}[7]$  приводит к аппаратной периодической (с периодом ~1 Гц) инверсии всех считанных из памяти бит пикселей в пределах квадрата 8x8 пикселей соответствующего атрибута.

#### 7.4.2 ZX hardware multicolor

Данный режим совпадает со стандартным ZX-режимом в части выборки байтов пикселей. Атрибут, в отличие от стандартного ZX-режима, индивидуальный у каждого байта пикселей. Смещение атрибута  $\text{attroffset} = \text{pixoffset} + \#2000$ , где  $\text{pixoffset}$  — смещение соответствующего байта пикселей. Интерпретация байта атрибута такая же, как в стандартном ZX-режиме.

#### 7.4.3 256x192 16 цветов

В данном режиме для каждого блока пикселей шириной 8 и высотой 1 пиксель приведём в

соответствие байт пикселей стандартного ZX-режима, который имеет смещение `zxoffset`. Для отображения блока пикселей считываются 4 байта:

- `byte0` — страница 4(6), смещение `zxoffset + #0000`
- `byte1` — страница 5(7), смещение `zxoffset + #0000`
- `byte2` — страница 4(6), смещение `zxoffset + #2000`
- `byte3` — страница 5(7), смещение `zxoffset + #2000`

Каждый пиксель блока пикселей (слева направо) имеет цвета: `{byte0[6], byte0[2:0]}`, `{byte0[7], byte0[5:3]}`, `{byte1[6], byte1[2:0]}`, `{byte1[7], byte1[5:3]}`, `{byte2[6], byte2[2:0]}`, `{byte2[7], byte2[5:3]}`, `{byte3[6], byte3[2:0]}`, `{byte3[7], byte3[5:3]}`.

#### 7.4.4 640x200 hardware multicolor

Разделим массив пикселей 640x200 на блоки, каждый высотой 1 пиксель и шириной 16 (блоки 16x1). Введём координаты таких блоков `X,Y` (верхний левый блок — 0,0; его сосед справа — 1,0; сосед снизу — 0,1 и т. д., `X=0..39`, `Y=0..199`). Введём смещение `offset = X + Y*40`. Для отображения блока пикселей считываются 4 байта по следующим адресам:

- `byte0` — страница 1(3), смещение `offset + #0000`
- `byte1` — страница 5(7), смещение `offset + #0000`
- `byte2` — страница 1(3), смещение `offset + #2000`
- `byte3` — страница 5(7), смещение `offset + #2000`

В пределах каждого блока есть два байта пикселей и два байта атрибутов. Для левых 8 пикселей блока байт пикселей — `byte1`, байт атрибутов — `byte0`; для правых 8 пикселей, соответственно, `byte3` и `byte2`.

Интерпретация байта атрибутов: `{attr[6], attr[2:0]}` — цвет, соответствующий единичному биту из байта пикселей, `{attr[7], attr[5:3]}` — цвет, соответствующий нулевому биту.

#### 7.4.5 320x200 16 цветов

Массив пикселей 320x200 разделяем на блоки 8x1 и вводим их координаты `X,Y` аналогично режиму 640x200 hardware multicolor. Требуемые для отображения блока 4 байта считываются по тем же адресам, что и в режиме 640x200 hardware multicolor. Составление пикселей из этих 4 байт производится так же, как и в режиме 256x192 16 цветов.

#### 7.4.6 Текстовый режим 80x25

Вводим разбиение массива *символов* на блоки 2x1, вводим координаты таких блоков `X,Y` (`X=0..39`, `Y=0..24`). В пределах блока коды левого и правого символов `lsum` и `rsum`, а также соответствующие им байты атрибутов `lattr` и `rattr` считываются по адресам:

- `lsum` — страница 5(7), смещение `#01C0 + Y*64 + X`



- lattr — страница 1(3), смещение #21C0 + Y\*64 + X
- rsym — страница 5(7), смещение #21C0 + Y\*64 + X
- ratrr — страница 1(3), смещение #01C1 + Y\*64 + X

После холодного старта код символа определяет отображаемый символ в соответствии с АТМ-кодировкой по умолчанию, приведённой на Рис. 1. На каждый символ байт атрибутов применяется аналогично режиму 640x200 hardware multicolor.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Рис. 1: Кодировка шрифта по умолчанию

## 7.4.7 Загрузка шрифта для текстового режима

Разрешение загрузки шрифта включается битом 2 порта #BF (см. стр. 9). ОЗУ шрифта имеет объём 2 килобайта, каждый символ хранится по адресам  $n*8..n*8+7$ , где  $n$  – номер символа (0..255). Образ символа 8x8 пикселей хранится построчно сверху вниз, биты в байтах представляют пиксели символа слева направо.

После включения разрешения загрузки шрифта *любая* запись процессором в адресное пространство *памяти* по адресу A ( $A=0..65535$ ) приводит к записи также и в ОЗУ шрифта по адресу  $A \& 2047$ .

### *Базовая конфигурация для ZX Evolution*

*Внимание!* В процессе загрузки ОЗУ шрифта рекомендуется выключать прерывания или производить загрузку в определённые моменты относительно сигнала `int`, так как вход в процедуру обработки прерывания при включенном разрешении загрузки шрифта приведёт к загрузке в ОЗУ шрифта случайных значений по случайным адресам.

## 8 Архитектура: порты чтения конфигурации и NMI

Данный раздел описывает средства, позволяющие программно считать конфигурацию памяти и палитру компьютера. Данные средства предназначены в основном для использования в процедуре обработки NMI.

### 8.1 Порт #BF (чтение)

Порт #BF позволяет читать своё содержимое (см. стр. 9).

### 8.2 Порт #xxBE (чтение)

Порт #xxBE предназначен для чтения конфигурации памяти и палитры компьютера.

Порт	Описание
#00BE.. <b>#07BE</b> <b>RO</b> <b>always</b>	Инверсный номер включенной страницы. Распределение: #00BE.. <b>#03BE</b> – номера страниц для окон #0000.. <b>#3FFF</b> , #4000.. <b>#7FFF</b> , #8000.. <b>#BFFF</b> и #C000.. <b>#FFFF</b> соответственно для нулевой карты памяти, #04BE.. <b>#07BE</b> – номера страниц для единичной карты памяти.
#08BE <b>RO</b> <b>always</b>	Биты «gamptom» (см. стр. 11). Порядок битов: бит 0 – соответствует номеру страницы, считанному из порта #00BE, бит 1 – номеру из порта #01BE и т.д.
#09BE <b>RO</b> <b>always</b>	Биты «dos7ffd» (см. стр. 11). Порядок такой же, как для порта #08BE.
#0ABE <b>RO</b> <b>always</b>	Последнее записанное значение в порт #7FFD.
#0BBE <b>RO</b> <b>always</b>	Последнее записанное значение в порт #EFF7.
#0CBE <b>RO</b> <b>always</b>	Последнее записанное значение в порт #xx77. Распределение бит: биты 3:0 – соответствуют битам данных 3:0 порта #xx77, бит 7 – соответствует биту адреса A14 при записи в #xx77, бит 6 — биту адреса A9, бит 5 – биту адреса A8.
#0DBE <b>RO</b> <b>always</b>	Чтение текущего отображаемого цвета. Предназначен для чтения палитры. Формат считываемого значения совпадает с форматом записываемого значения в порт #FF при установке палитры (см. стр. 14), за исключением битов 2 и 3, которые <i>не определены</i> . Процедура считывания палитры аналогична процедуре записи (см. стр. 14), но производится чтение порта #0DBE вместо записи порта #FF.

Алгоритм восстановления состояния диспетчера страниц:

1. Записывая в порт #xFF7, восстановить биты «dos7ffd».
2. Если страница в данном окне данной карты памяти — ROM или RAM с номером менее 64, то восстановить её запись в #xFF7 одновременно с пунктом 1.
3. Для всех страниц RAM с номером не менее 64 восстановить номер *дополнительной* записью в порт #x7F7.

### 8.3 Порт #BE (запись)

Запись в порт #BE происходит при выходе из обработчика NMI. Подробнее см. стр. 21.

### 8.4 Вход в NMI

После подачи NMI (см. стр. 7) процессор производит следующие действия:

1. Сохраняет адрес следующей команды на стеке
2. Запоминает состояние разрешённости прерываний INT в флаге IFF2, при этом запрещая дальнейшие прерывания INT очисткой флага IFF1.
3. Начинает выполнять программу с адреса #0066

Одновременно с подачей NMI на процессор компьютер включает в окно процессора #0000..#3FFF последнюю (с номером #FF) страницу ОЗУ. Таким образом, процедура обработки NMI должна находиться в этой странице по смещению #0066.

Так как компьютер производит подключение последней страницы ОЗУ одновременно с началом сигнала NMI, а вход в обработку происходит только после окончания выполнения текущей команды, то вход будет выполнен некорректно (может разрушиться состояние процессора, содержимое ОЗУ программы и содержимое последней страницы ОЗУ) в случае, если сигнал NMI совпал с одним из следующих случаев:

- Процессор выполнял команду по адресам #0000..#3FFF
- Стек был установлен в адреса #0000..#3FFF
- Команда процессора считывала или записывала память по адресам #0000..#3FFF
- Запись на стек программы нарушает её функционирование

Начало сигнала NMI совпадает с началом сигнала INT, таким образом в случае разрешённых прерываний INT, будет выполнен вход в NMI *вместо* входа в INT. При выходе из обработчика NMI следует иметь это в виду и при необходимости продолжать выполнение программы с обработчика INT.

Рекомендуемый алгоритм входа в NMI для случая, когда требуется продолжить выполнение программы:

1. Сразу после входа в NMI (команда по смещению #0066 в последней странице ОЗУ) сохранить содержимое стека в предназначенную для этого ячейку последней страницы ОЗУ
2. Переставить стек в место, выделенное в последней странице ОЗУ
3. Сохранить все регистры процессора на стек

4. Узнать состояние разрешённости прерываний путём считывания флага IFF2 командой **LD A,R** или **LD A,I**.

## 8.5 Выход из NMI

Рекомендуемый алгоритм выхода из обработчика NMI:

1. Если обработчик NMI выполнял команды **DI** или **EI**, то следует восстановить флаг IFF2 процессора (теми же командами). При этом надо следить, чтоб до момента выхода из NMI не произошло прерывания INT.
2. Восстановить регистры процессора.
3. Переставить указатель стека на стек программы
4. Выполнить команду **OUT (#BE),A**. Данная команда не влияет на содержимое флагов и регистров процессора. Записываемое значение игнорируется.
5. Сразу же после команды **OUT (#BE),A** выполнить команду **RETN**. В процессе выполнения этой команды в окне процессора #0000..#3FFF включится вместо последней страницы ОЗУ заданная диспетчером страница и процессор продолжит выполнять прерванную программу.

*Примечание:* выключение подстановки последней страницы ОЗУ происходит на 2ой по порядку сигнал M1 процессора после записи в порт #BE независимо от того, какие команды выполняются в этот момент.

В момент выполнения команды **RETN** происходит копирования состояния разрешённости прерываний INT из флага IFF2 в флаг IFF1.

Команды **DI** и **EI** устанавливают флаги IFF1 и IFF2 в одинаковое состояние, таким образом в случае выполнения этих команд в обработчике NMI, последующая команда **RETN** не изменит состояние разрешённости прерываний.

## 9 Архитектура: порты ввода-вывода

### 9.1 Порт #FE

Порт считывания состояния клавиатуры и магнитофонного входа, порт установки бита бипера, бита магнитофонного выхода, цвета бордюра.

Порт	Описание			
#xxFE RW always	Чтение:			
	7: 1	6: tape_in	5: 0	4..0: keyboard
	Запись:			
	7..5: не исп.	4: beeper	3: tape_out	2..0: цвет бордюра

Чтение клавиатуры происходит стандартным для ZX-Spectrum способом — установкой в 0 адресных бит A15..A8, соответствующих требуемым рядам, данные всех выбранных рядов складываются по AND, результат возвращается в битах 4..0. Чтение происходит без wait'ов.

Магнитофонный вход — отдельный вход ZX Evolution.

## 9.2 Порт COVOX (#xxFB, WO always)

Порт предназначен для вывода 8-битного звука. Звук выводится на биперный выход при помощи широтно-импульсной модуляции с частотой около 218 кГц. Записываемые значения — беззнаковые отсчёты от 0 до 255. После записи в порт #xxFB биперный выход начинает работать в режиме COVOX'a вплоть до последующей записи в порт #FE, после чего биперный выход начинает работать в обычном режиме.

## 9.3 Порты AY/YM

Порт	Описание
#FFFD RW always	Запись — выбор номера регистра AY/YM Чтение — чтение содержимого регистра AY/YM
#BFFD WO always	Запись — запись в регистр AY/YM

## 9.4 Порт Kempston Joystick

Kempston Joystick — пятикнопочный манипулятор. Позволяет управлять программами, в которых реализована поддержка этого манипулятора. Обращение к порту манипулятора происходит в безвайтовом режиме.

Порт	Описание							
#xx1F RO noshad	Чтение значения из порта позволяет получить состояние игрового манипулятора. Формат байта полученного значения:							
	<table><tr><td>7: 0</td><td>6: 0</td><td>5: 0</td><td>4: Fire</td><td>3: Up</td><td>2: Down</td><td>1: Left</td><td>0: Right</td></tr></table> <ul style="list-style-type: none"><li>• Fire – 1: нажата кнопка «Огонь» на манипуляторе, 0: не нажата;</li><li>• Up – 1: нажата кнопка «Вверх» на манипуляторе, 0: не нажата;</li><li>• Down – 1: нажата кнопка «Вниз» на манипуляторе, 0: не нажата;</li><li>• Left – 1: нажата кнопка «Влево» на манипуляторе, 0: не нажата;</li><li>• Right – 1: нажата кнопка «Вправо» на манипуляторе, 0: не нажата.</li></ul>	7: 0	6: 0	5: 0	4: Fire	3: Up	2: Down	1: Left
7: 0	6: 0	5: 0	4: Fire	3: Up	2: Down	1: Left	0: Right	

## 9.5 Порты Kempston Mouse

Kempston Mouse — интерфейс манипулятора типа «мышь». Позволяет управлять программами, в которых реализована поддержка этого манипулятора. Обращение к порту манипулятора происходит в безвайтовом режиме.

Поддерживаются трехкнопочные мыши, а так же мышь с колесом.

Порт	Описание					
#FADF RO always	Чтение порта позволяет определить состояние кнопок и получить счетчик поворота колеса. Формат байта полученного значения:					
	<table><tr><td>7..4: Wheel counter</td><td>3: 1</td><td>2: MMB</td><td>1: RMB</td><td>0: LMB</td></tr></table>	7..4: Wheel counter	3: 1	2: MMB	1: RMB	0: LMB
	7..4: Wheel counter	3: 1	2: MMB	1: RMB	0: LMB	
<ul style="list-style-type: none"><li>Wheel counter – счетчик поворота колеса;</li><li>MMB – 0: нажата средняя кнопка мыши, 1: не нажата;</li><li>RMB – 0: нажата правая кнопка мыши, 1: не нажата;</li><li>LMB – 0: нажата левая кнопка мыши, 1: не нажата;</li></ul>						
#FBDF RO always	Чтение порта позволяет определить значение счетчика координаты X положения мыши.					
#FFDF RO always	Чтение порта позволяет определить значение счетчика координаты Y положения мыши.					

В случае если мышь не обнаружена, из портов считывается значение 0xFF.

## **9.6 Порты управления часами и NVRAM**

Доступ и управление часами и NVRAM программно совместим с широкой распространенной в экс-СССР схемой Gluk Clock (основанной на микросхеме K512ВИ1 или DALLAS 12877, 12877A).

## Базовая конфигурация для ZX Evolution

Порт	Описание
<b>#EFF7 WO noshad</b>	<p>7ой бит этого порта управляет доступом к ячейкам часов в не теновом режиме:</p> <ul style="list-style-type: none"> <li>установка бита 7 — разрешает доступ;</li> <li>снятие бита 7 — запрещает доступ.</li> </ul> <p>Значение после сброса — 0.</p> <p><i>Внимание!</i> Остальные биты этого write-only порта используются для других целей и описаны в других местах данной документации.</p>
<b>#DFF7 WO noshad</b>  <b>#DEF7 WO shadow</b>	<p>Запись в порт устанавливает адрес ячейки (0x00..0xEF).</p> <p>Примечание: в теновом режиме порт <b>#DEF7</b> доступен независимо от состояния бита 7 порта <b>#EFF7</b>.</p>
<b>#BFF7 RW noshad</b>  <b>#BEF7 WO shadow</b>	<p>Чтение/запись ячейки.</p> <p>Примечание: в теновом режиме порт <b>#BEF7</b> доступен независимо от состояния бита 7 порта <b>#EFF7</b>.</p>

Так как часы компьютера ZX Evolution основаны на микросхеме PCF8583, то совместимость обеспечивается путем эмуляции.

Особенности эмуляции:

- Поддерживается запись/чтение всех ячеек NVRAM (0x0E...0x3F), добавлена возможность доступа к дополнительным пользовательским ячейкам (0x40..0xEF);
- Поддерживается чтение всех ячеек , несущих дополнительные функции (0x00..0x0D);
- Поддерживается запись ячеек, управляющих временем:
  1. 0x00 – регистр секунд;
  2. 0x02 – регистр минут;
  3. 0x04 – регистр часов;
  4. 0x06 – регистр дня недели;
  5. 0x07 – регистр дня месяца;
  6. 0x08 – регистр месяца;
  7. 0x09 – регистр года.
- Статусные ячейки возвращают следующие значения:
  1. 0x0A – возвращает значение 0x00;



2. 0x0B – возвращает значение 0x02 (доступен на запись — установка бита data mode);
  3. 0x0C – возвращает значение 0x00 (поддерживается состояние update flag);
  4. 0x0D - возвращает значение 0x80.
- Статусная ячейка 0x0C имеет дополнительный функционал. При записи в эту ячейку доступны следующие функции.
    1. 0-ой бит — управляет сбросом буфера кодов PS/2 клавиатуры. В случае если бит установлен в 1, происходит сброс буфера кодов PS/2 клавиатуры (см. п9.8);
    2. 1-ый бит — управляет состоянием Caps Lock Led на PS/2 клавиатуре (значение сохраняется). В случае если бит установлен в 0 — Caps Lock Led не активен, если установлен в 1 - Caps Lock Led активен.
  - Операция чтения значения ячейки является вайтовой и занимает небольшое кол-во тактов (чтение происходит из внутренних регистров ATMEGA128, дублирующих значения в PCF8583);
  - Операции записи значения ячейки является вайтовой и занимает достаточно большое кол-во тактов (цикл записи в ячейку включает этап записи значения в PCF8583).

## **9.7 Чтение версии базовой конфигурации/бутлоадера.**

Прикладное ПО имеет возможность считывать версию/название базовой конфигурации и бутлоадера ZX Evolution. Обращение к этому функционалу происходит через порты управления часами и NVRAM (см. п9.6) так, как будто это расширение функционала эмулируемой м/сх 12877.

Записью значения в любую из ячеек с адресами 0xF0..0xFF можно выбрать, что будет считываться. Допустимы следующие значения:

- 0 — версия базовой конфигурации;
- 1 — версия бутлоадера.

Длина данных версии составляет 16 байт (соответственно ячейки 0xF0..0xFF) и имеет следующий формат:

- 0xF0..0xFB – 12 байт, название версии в кодировке ASCII (если название короче 12 байт, то дополняется нулями);
- 0xFC..0xFD – 2 байт (little endian), дата и бит релиза (если бит не установлен, то считается, что версия тестовая beta);
- 0xFE..0xFF – 2 байт (little endian), CRC значение прошивки. CRC считается для файла прошивки AVR (zxevo\_fw.bin) и не имеет практического смысла для Z80.

Формат упаковки даты и бита релиза:

- 7 бит 0xFD – 0: тестовая версия; 1: официальный релиз;
- 6..1 биты 0xFD – год выпуска релиза, значения 0..63 соответствуют 2000..2063 годам;
- 0 бит 0xFD и 7..5 биты 0xFC – месяц выпуска релиза (допустимые значения: 1..12);
- 4..0 биты 0xFC – день выпуска релиза (допустимые значения: 1..31).

Пример:

50 65 6E 74 31 6D 00 00 00 00 00 00 00 7B 14 3C B1

50 65 6E 74 31 6D 00 00 00 00 00 00 00 — строка «Pent1m»;

7B 14 – число #147B или %0001\_0100\_0111\_1011: beta, 10 год, 03 месяц, 27 день (27.03.2010);

3C B1 — число #B13C, CRC.

## 9.8 Доступ к PS/2 клавиатуре.

Прикладное ПО имеет возможность считывать коды PS/2 клавиатуры напрямую. Обращение к этому функционалу происходит через порты управления часами и NVRAM (см. п9.6) так, как будто это расширение функционала эмулируемой м/сх 12877.

Записью значения 2 в любую из ячеек с адресами 0xF0..0xFF включает режим доступа к буферу кодов PS/2 клавиатуры.

Путем чтения с адресами 0xF0..0xFF можно получить значения из буфера кодов PS/2 клавиатуры, при этом считанное значение удаляется из буфера. Если буфер кодов PS/2 клавиатуры пуст, то возвращается значение 0. Если буфер кодов PS/2 клавиатуры переполнен, то возвращается значение 0xFF. В случае переполнения буфера рекомендуется его сбросить (путем записи значения 1 в ячейку с адресом 0x0C).

**Внимание!** В буфер кодов PS/2 клавиатуры записываются коды клавиш не длиннее двух байт (трех байт, если с префиксом отпускания клавиш).

## 9.9 Доступ к SD-карте.

Происходит аналогично доступу в Z-контроллере от КОЕ. В обычном режиме:

Порт	Описание		
#xx77 RW noshad	Запись: управление сигналом CS на SD-карту:		
	Биты 7..2: устанавливайте в 0 для совместимости	Бит 1: сигнал CS, 1 после сброса, устанавливать в 0 для выбора SD-карты	Бит 0: устанавливайте в 1 для совместимости с Z-контроллером
	Чтение: всегда 0 (карта всегда вставлена и в режиме R/W — в соответствии с интерпретацией в Z-контроллере). Реальное наличие карты следует проверять попыткой её инициализации с тайм аутom.		
#xx57 RW noshad	Запись: отсылка байта в SD-карту по SPI, одновременно принятый байт можно в дальнейшем считать из этого же порта.		
	Чтение: считать ранее принятый байт, отослать #FF в карту. Вновь принятый байт доступен при повторном чтении.		

Примечание: в цикле обмена по SPI, инициируемом *записью или чтением* порта #xx57, происходит *одновременно* отсылка байта в SD-карту и приём байта от неё. Отсылаемый байт тот же, что записан в этот порт (если цикл обмена инициирован записью), или #FF, если цикл обмена инициирован чтением порта.

Принятый байт запоминается во внутреннем буфере и доступен для последующего чтения из этого же порта. Данное чтение вновь инициирует цикл обмена и т. д.

Допускается читать/писать порт #xx57 командами **INIR** и **OTIR**. Пример чтения сектора:

```
LD    C, #57
LD    B, 0
INIR
INIR
```

SD-карта также доступна в shadow-режиме. В этом режиме порт #xx77 используется для других целей, потому остаётся только порт #xx57 и адресация меняется следующим образом.

Порт	Описание
#xx57 RW shadow	<p>Запись: если A15=0, то отсылка байта в SD-карту по SPI, одновременно принятый байт можно в дальнейшем считать из этого же порта — аналогично записи в #xx57 в noshad-режиме. Если A15=1 — управление сигналом CS, аналогично записи в #xx77 в noshad-режиме.</p> <p>Чтение: считать ранее принятый байт, отослать #FF в карту. Вновь принятый байт доступен при повторном чтении. Чтение полностью аналогично noshad-режиму.</p>

Таким образом полноценный доступ к SD-карте сохраняется и в shadow-режиме. Запись командой **OTIR** 512 байт выглядит так (**OTIR** работает с предекрементом **B**):

```
LD    C, #57
```

```
LD    B, #80
OTIR
LD    B, #80
OTIR
LD    B, #80
OTIR
LD    B, #80
OTIR
```

### 9.10 Доступ к IDE-устройствам.

IDE-интерфейс сделан по стандарту nemo-ide с некоторыми расширениями. В отличие от оригинала (и Z-контроллера), IDE-порты доступны и в shadow-режиме. Однако, в отличие от оригинала, поддерживаются лишь порты #10, #11, #30, #50, #70, #90, #B0, #D0, #F0 и #C8, при этом осуществляется полная дешифрация (по 8 младшим битам адреса). Расширенный режим передачи данных по сравнению с оригиналом заключается в возможности читать и писать сектора полностью через порт #10 командами INIR и OTIR, причём в последнем случае нет надобности переставлять старший и младший байты. Расширенный режим не требует какого-либо специального включения, т. е. работает одновременно со старым (nemo-ide).

*Базовая конфигурация для ZX Evolution*

<i>Порт</i>	<i>Описание</i>
<b>#xx10</b> <b>RW</b> <b>always</b>	Порт для записи или чтения данных, читается/пишется как младшая часть 16-битного слова (режим совместимости с nemo-ide), так и старшая тоже (расширенный режим).
<b>#xx11</b> <b>RW</b> <b>always</b>	Чтение и запись старшей части 16-битного слова (режим совместимости с nemo-ide). Чтение происходит после чтения <b>#10</b> , запись сюда — до записи в <b>#10</b> . Обращение к этому порту не приводит к физическому обращению в IDE-устройства.
<b>#xx30</b> <b>RW</b> <b>always</b>	Порт ошибок/свойств. Этот и дальнейшие порты см. в документации на протокол ATA. Биты ША Z80 7..5 маются на биты адреса шины ATA 2..0, порты <b>#10</b> , <b>#30</b> , ..., <b>#F0</b> маются на CS0 шины ATA, порт <b>#C8</b> — на CS1.
<b>#xx50</b> <b>RW</b> <b>always</b>	Счётчик секторов
<b>#xx70</b> <b>RW</b> <b>always</b>	Номер сектора (CHS) или биты 7..0 LBA-адреса
<b>#xx90</b> <b>RW</b> <b>always</b>	Биты номера цилиндра 7..0 (CHS) или биты 15..8 LBA-адреса
<b>#xxB0</b> <b>RW</b> <b>always</b>	Биты номера цилиндра 15..8 (CHS) или биты 23..16 LBA-адреса
<b>#xxD0</b> <b>RW</b> <b>always</b>	Номер головки (CHS) или биты 27..24 LBA-адреса, выбор устройства (master/slave), выбор режима (LBA/CHS)
<b>#xxF0</b> <b>RW</b> <b>always</b>	Регистр состояния/регистр команд
<b>#xxC8</b> <b>RW</b> <b>always</b>	Регистр состояния/управления

Чтение данных в режиме nemo-ide выглядит так:

```

IN    A, (#10)    ;читаем младший байт, одновременно старший
                    ;защёлкивается в #11

LD     (HL), A
INC    HL
IN     A, (#11)    ;читаем ранее защёлкнутый старший байт
LD     (HL), A
INC    HL

```

Запись данных в режиме nemo-ide:

## Базовая конфигурация для ZX Evolution

```
LD    D, (HL)
INC   HL
LD    A, (HL)
INC   HL
OUT   (#11), A    ;пишем в защёлку старший байт
LD    A, D
OUT   (#10), A    ;пишем младший байт одновременно со старшим
                     ;из защёлки
```

Запись и чтение в расширенном режиме:

```
LD    C, #10
LD    B, 0
INIR  ;Читаем сектор из #10 (2 раза по 256 байт)
INIR
...
LD    C, #10
LD    B, 0
OTIR  ;пишем сектор в #10 (2 раза по 256 байт)
OTIR
```

**Внимание!** В расширенном режиме синхронизация выдачи младшего или старшего байтов (а также записи сначала младшего, а потом старшего байта) в начале чтения или записи сектора осуществляется в момент доступа в порты #30, #50 и т.д. (любые IDE-порты кроме #10 и #11).

### 9.11 Порты контроллера дисководов

Порт	Описание
#xx1F RW shadow	Регистр команд/состояния ВГ93
#xx3F RW shadow	Регистр трека ВГ93
#xx5F RW shadow	Регистр сектора ВГ93
#xx7F RW shadow	Регистр данных ВГ93
#xxFF RW shadow	«Системный» регистр. Чтение: бит 7 — состояние сигнала INTRQ от ВГ93 бит 6 — состояние сигнала DRQ от ВГ93 биты 5..0 — единичные

	<p>Запись:</p> <p>Биты 1..0: выбор одного из 4 дисководов</p> <p>бит 2: сигнал RESET на ВГ93, 0 после сброса</p> <p>бит 3: сигнал HRDY на ВГ93</p> <p>бит 4: выбор стороны дискеты</p>
<p>#xx2F, #xx4F, #xx6F, #xx8F RW shadow</p>	<p>Дополнительные порты, использующиеся для эмуляции ВГ93 при работе с рамдискком. Представляют из себя обычные ячейки памяти: считывается последнее записанное состояние, после сброса не определены.</p>

## 9.12 Порт RS232 (коммуникационный интерфейс)

Работа с коммуникационный интерфейс RS232C на ZX Evolution реализована по стандарту предложенным Кондратьевым без использования прерываний. В этом стандарте использовался стандартный PC совместимый ISA-модем, базировавшийся на микросхеме 16550 или совместимой. Соответственно внутренние регистры интерфейса соответствуют регистрам вышеуказанной микросхемы. В связи с тем, что не весь функционал 16550 используется в нашей реализации, часть регистров или битов регистров не используется. Поэтому рекомендуется перед программированием ознакомиться с ограничениями реализации (в документации описаны только используемые регистры и биты).

Особенности реализации:

- отсутствует поддержка прерываний;
- поддерживаются только следующие дополнительные сигналы коммуникационного интерфейса: CTS, RTS;
- FIFO всегда используется.

Порт	Описание
#F8EF RW	<LCR&0x80==0> <b>DAT</b> – регистр данных <LCR&0x80==1> <b>DLL</b> – младший регистр делителя
#F9EF RW	<LCR&0x80==0> <b>IER</b> – регистр разрешения прерываний (не используется) <LCR&0x80==1> <b>DLM</b> – старший регистр делителя
#FAEF RW	Запись: <b>FCR</b> – регистр управления FIFO Чтение: <b>ISR</b> – регистр статуса прерываний (не используется)
#FBEF RW	<b>LCR</b> – регистр управления линией
#FCEF RW	<b>MCR</b> – регистр управления модемом
#FDEF RO	<b>LSR</b> – регистр статуса линии
#FEEF RO	<b>MSR</b> – регистр статуса модема
#FFEF RW	<b>SPR</b> – пользовательский регистр

### 9.12.1 Регистр данных (DAT)

Регистр используется для передачи/приема данных.

### 9.12.2 Регистры делителя (DLL,DLM)

Регистр используется для установки скорости передачи порта.

Существует два режима работы делителя:

- стандартный режим (старший бит **DLM** установлен в 0);
- нативный режим (старший бит **DLM** установлен в 1);

Для стандартного режима скорость рассчитывается, как для микросхемы 16550, по формуле:  
 $\langle \text{скорость в бодах} \rangle = 115200 / ((\mathbf{DLM}) * 256) + \mathbf{DLL}$

Для нативного режима значения **DLL** и **DLM** (без учета старшего бита) напрямую записываются в регистры **UBRRL** и **UBRRH** микросхемы ATMEGA128. В этом случае значения **DLL** и **DLM** рассчитывается по формуле:

$(\text{uint16}) ((\mathbf{DLM} \& 0x7F) * 256 + \mathbf{DLL}) = ((\mathbf{F\_CPU} / 16) / \langle \text{скорость в бодах} \rangle) - 1$   
 где **F\_CPU** = 11059200 (для ZXEvO)

**Пример:** Рассчитаем значения **DLL** и **DLM** для скорости 31250 бод (скорость MIDI интерфейса) при использовании нативного режима.

1. Рассчитаем значение делителя  $((11059200/16)/31250)-1 = 21$  [0x15];



2. Устанавливаем значения **DLL** = 0x15, **DLM** = 0x80 (старший бит включает нативный режим).

**Внимание!** ZX Evolution поддерживает дополнительную скорость передачи 256000, для этого нужно установить **DLM** и **DLL** в значение 0.

### 9.12.3 Регистр управления FIFO (FCR)

Регистр используется для управления FIFO.

- 7..3 биты: не используются;
- 2 бит (**XMIT FIFO reset**): сброс FIFO отправки, если установлена 1;
- 1 бит (**RCVR FIFO reset**): сброс FIFO приема, если установлена 1;
- 0 бит (**enable FIFO**): включает FIFO, если установлена 1. Так как FIFO всегда включено, то этот бит должен быть установлен в случае подачи команд на сброс FIFO. В случае 0, команды сброса FIFO игнорируются.

### 9.12.4 Регистр управления линией (LCR)

Регистр используется для установки формата передачи данных по коммуникационному интерфейсу.

- 7 бит (**DL enable**): запись 1 устанавливает видимым по портам #F8EF, #F9EF соответствующие части регистра делителя **DLL**, **DLM**;
- 6 бит: не используется;
- 5..3 бит: задает алгоритм расчета бита четности данных. Поддерживаются только следующие значения:  
001 — odd parity;  
011 — even parity;  
все остальные значения — четность не используется.
- 2 бит (**stop bits**): кол-во стопбит. Кол-во “стоп-битов”: при 0 = 1, при 1 = 1,5 в режиме 5 бит данных, иначе = 2.
- 1,0 биты (**word length**): длина слова данных. Возможны следующие значения:  
00 — 5 бит;  
01 — 6 бит;  
10 — 7 бит;  
11 — 8 бит.

### 9.12.5 Регистр управления модемом (MCR)

Регистр используется для установки режима интерфейса и управления дополнительными сигналами.

- 7..2 биты: не используются;
- 1 бит (**RTS**): управление сигналом RTS интерфейса;
- 0 бит: не используется.

### 9.12.6 Регистр статуса линии (LSR)

Регистр для получения состояния линии и FIFO.

- 7 бит: не используется;
- 6 бит (**transmit empty**): если установлена **1**, то передатчик и FIFO отправки пустые;
- 5 бит (**THR empty**): если установлена **1**, то FIFO отправки данных пустое;
- 4 бит: не используется;
- 3 бит (**framing error**): если установлена **1**, то произошла ошибка кадра;
- 2 бит (**parity error**): если установлена **1**, то произошла ошибка четности;
- 1 бит (**overrun error**): если установлена **1**, то FIFO приема данных переполнено;
- 0 бит (**data received**): если установлена **1**, то FIFO приема данных не пустое;

### 9.12.7 Регистр статуса модема (MSR)

Регистр для получения состояния дополнительных сигналов интерфейса.

- 7 бит (**CD status**): всегда **1**;
- 6 бит (**RI status**): всегда **0**;
- 5 бит (**DSR status**): всегда **1**;
- 4 бит (**CTS status**): отображает состояние линии CTS интерфейса;
- 1..3 биты: не используется, всегда **0**;
- 0 бит (**CTS change**): устанавливается в **1**, если произошло изменение состояния сигнала CTS (сбрасывается в **0** после чтения регистра).

### 9.12.8 Пользовательский регистр (SPR)

Регистр служит для хранения пользовательских данных. Сбрасывается в случае выключения питания или полного сброса компьютера.

## **10 История версий**

### **10.1 Версия 29.09.2011**

- Добавлен ФАПЧ для ВГ93;
- Скорректирована обработка линии RTS коммуникационного порта;
- Добавлен выбор BCD/HEX в эмуляции работы GLUK часов;
- Изменен алгоритм обработки NMI.

### **10.2 Версия 11.06.2011**

- Добавлен обработчик NMI;
- Добавлена работа с текстовым режимом и поддержка шрифтов;
- Скорректирована процедура инициализации PS/2 мыши;
- Добавлено управление NMI с внешних контактов (для плат revision A и B - замыканием контактов “Turbo”).

### **10.3 Версия 11.05.2011**

- Добавлена поддержка нативного режим для делителя скорости RS232;
- Учитывается управление умножителем частоты ICS501M (для плат revision C).

### **10.4 Версия 26.04.2011**

- Добавлена поддержка коммуникационного интерфейса (RS232);
- Повышена надежность загрузки FPGA;
- Повышена надежность работы с RTC (исправлена ошибка, проявляющаяся на некоторых платах).

### **10.5 Версия 30.03.2011**

- Поправлено описание портов кемпстон-мышки.
- Добавлено описание ковокса.
- Поправлены рекомендуемые значения старшего байта адреса для порта #xx77.

- Добавлено описание поддерживаемых видеорежимов.
- Добавлен доступ к дополнительным ячейкам NVRAM (0x40..0xEF).

#### **10.6      *Версия 07.12.2010***

- Добавлена регулировка разрешения PS/2 мыши.
- Маппер клавиатуры PS/2 в ZX реализован без использования RAM микроконтроллера.

#### **10.7      *Версия 29.11.2010***

- Реализован выбор («Num Lock») и индикация (Num Lock led) режима аудиовыхода (beeper/tapeout).
- Поддержан маппер клавиатуры PS/2 в ZX для всех дополнительных кодов клавиатуры (E0 xx).
- Скорректирован менеджер памяти и видимость портов в различных режимах.

#### **10.8      *Версия 17.10.2010***

- Поддержка загрузки с магнитофонного входа.
- Скорректировано управление ресетами с PS/2 клавиатуры («F12» и «CTRL-ALT-DEL»).
- Реализован менеджер для всей памяти (ROM и RAM).
- Оптимизирована работа с PS/2 устройствами.
- Поддержан маппер PS/2 в ZX клавиатуру (пока без возможности редактирования).
- Реализованы расширения nemo-ide.

#### **10.9      *Версия 30.03.2010***

- Исправлена ошибка загрузки FPGA.

#### **10.10     *Версия 28.03.2010***

- Исправлена обработка ошибок PS/2 мыши. Реинициализация мыши в случае ошибок обмена.
- Поддержка получения версии базовой конфигурации и бутлоадера.

- Исправлена обработка ошибок тайм аута обмена с PS/2 устройствами.

### **10.11    Версия 24.03.2010**

- Исправлено поведение светодиода питания (Pwr Led), теперь выключается если основное питание отключено.
- Управление сигналом NMI на ZX с помощью кнопки «Print Screen» PS2 клавиатуры.
- Мягкий сброс компьютера (без перезагрузки fpga) при замыкании контактов «soft reset» на плате.

### **10.12    Версия 10.03.2010**

- Добавлена индикация режима TV/VGA на индикаторе «Scroll Lock» PS2 клавиатуры (горящий светодиод — VGA, не горящий TV).
- Исправлена ошибка отображения NVRAM, используемой в GLUK, в NVRAM PCF8583.
- Исправлена ошибка в эмуляции часов GLUK, реализована эмуляция Update flag в регистре C.
- Реализовано сохранение режима компьютера в NVRAM.
- Добавлена поддержка механической ZX клавиатуры.
- Добавлена поддержка Kempston джойстика.

### **10.13    Версия 04.02.2010**

Базовая версия (1.00 в SVN).