

# **FPGA-конструктор советских микро-ЭВМ**

ДВК-1

ДВК-2

ДВК-3

Электроника-60

## **Описание проекта**

# Содержание

1. Введение.....	3
2. Общая структура проекта.....	3
3. Используемые FPGA.....	5
4. Структура дерева файлов проекта.....	5
5. Настройка конфигурации создаваемой ЭВМ.....	6
6. Внешние органы управления и индикации.....	6
7. Интерфейсный модуль.....	7
7.1. Контроллер SDRAM.....	8
7.2. Тактовый генератор PLL.....	8
7.3. Схема управления цветами VGA DAC.....	9
8. Соединительный модуль верхнего уровня.....	9
8.1. ПЗУ пользователя.....	10
8.2. Генератор сбросов.....	10
8.3. Контроллер прерываний.....	10
8.4. Арбитр DMA.....	10
8.5. Арбитр SD-карты.....	11
8.6. Формирователь сигналов управления видео-ЦАП VGA.....	11
8.7. Коммутатор линий последовательных портов.....	11
9. Процессорные платы.....	12
9.1. Плата MC1201.01.....	12
9.2. Плата MC1201.02.....	13
9.3. Плата MC1260.....	13
9.4. Плата MC1280.....	14
10. Дисковая подсистема.....	14
10.1. Распределение адресов на SD-карте.....	14
10.2. SDSPI.....	15
10.3. Модули дисковых контроллеров.....	16
10.3.1. Контроллер RK11/RK05.....	17
10.3.2. Контроллер RD50C.....	17
10.3.3. Контроллер RX11/RX01.....	18
10.3.4. Контроллер НГМД MY.....	18
10.4. Ручная загрузка операционной системы с дисков.....	19
10.4.1. Загрузка с RK.....	19
10.4.2. Загрузка с диска MY.....	19
10.4.3. Загрузка с диска DX.....	19
11. Контроллер ИРПС.....	20
12. Контроллер ИРПР.....	20
13. Терминальная подсистема.....	20
13.1. Контроллер КСМ.....	21
13.1.1. Внутреннее устройство контроллера.....	21
13.1.2. Поддерживаемые режимы работы.....	21
13.1.3. Кодировка символов.....	22
13.1.4. Формат экрана.....	22
13.1.5. Клавиатура.....	24
13.1.6. Шрифт.....	24
13.2. Контроллер КГД.....	26
13.3. Аппаратные особенности терминальной подсистемы.....	26

# 1. Введение

Данная разработка является FPGA-конструктором, позволяющим из набора готовых модулей собрать FPGA-вариант старых советских pdp11-совместимых микро-ЭВМ. Проект охватывает микро-ЭВМ серий ДВК-1, ДВК-2, ДВК-3 и Электроника-60, являющихся архитектурными подобиями соответствующих ЭВМ серии pdp-11 фирмы DEC.

Изначально я был вынужден заняться этой разработкой для замены окончательно вышедшей из строя процессорной платы, стоящей в контроллере испытательного стенда в нашей лаборатории. Тут очень кстати великий VSLAV подарил миру FPGA-версию процессора 1801BM2, чем и спас наше оборудование. В процессе разработки оказалось проще сделать на FPGA весь контроллер, чем эмулировать асинхронную шину МПИ, в результате чего получилась ДВК-совместимая машина, управляющая нашим стендом. Эта разработка уже полгода работает у нас в лаборатории в достаточно напряженном режиме, и все явные баги более-менее выявлены и устранены.

Проект основан на всех процессорных ядрах, для которых VSLAV произвел реверс и создал рабочее HDL-описание. Возможно, когда-нибудь список поддерживаемых процессоров будет расширен.

## 2. Общая структура проекта

Все поддерживаемые проектом советские микро-ЭВМ были модульными, и представляли собой соединительную корзину, в которую устанавливались процессорная плата и несколько плат периферийных устройств. Данный проект имеет такую же структуру, содержит ряд готовых процессорных и периферийных плат, и позволяет добавлять также и пользовательские устройства. В этом проекте в качестве общей шины, вместо асинхронной МПИ, используется синхронная шина Wishbone.

Имеются 4 процессорные платы, основанные на 4 имеющихся процессорных ядрах:

- K1801BM1
- K1801BM2
- M2 (LSI-11)
- M4 (LSI-11M)

Особенности этих плат рассматриваются в соответствующем разделе.

Также в проект входят модули периферийных устройств, из которых набирается конфигурация ЭВМ:

- Последовательный порт ИРПС, используемый для подключения консольного терминала (ТТ:), а также дополнительных терминалов, последовательных принтеров (LS:) и магистралей связи с другими ЭВМ (XL:). В конфигурацию по умолчанию входят два таких модуля.
- Параллельный порт ИРПР (LP:) для подключения параллельного принтера. Внешний интерфейс модуля использует протокол Centronics, поскольку настоящих ИРПР-принтеров уже не осталось в природе.
- Дисковый контроллер RK11 (RK:) с подключенными к нему 8 дисками RK05.
- Дисковый контроллер RX11 (DX:) с подключенными к нему двумя дисками RX01 (он же ГМД-70)
- Дисковый контроллер КМД (MY:) с подключенными к нему двумя дисковыми НГМД-6121 (4 логических диска).

- Контроллер жесткого диска RD50C (DW:) с подключенным к нему HDD на 64М. Контроллер выполнен в стандарте ДВК, а не оригинальном дековском, и для работы требует драйвер DW именно от ДВК (штатный драйвер DW из поставки rt-11 работать не будет).
- Текстовый терминал КСМ (контроллер символьного монитора) с VGA-выходом, подключенный к консольному ИРПС. Скорости терминала и ИРПС аппаратно согласованы.
- Графический контроллер КГД (контроллер графического дисплея).
- Динамическая память SDRAM размером 64К
- Пользовательское ПЗУ размером 8К

Любой ненужный модуль можно легко выкинуть из конфигурации для экономии ресурсов FPGA. Некоторые модули могут работать только совместно (КГД не работает без КСМ). Кроме того, к общей шине можно подключать и другие самодельные модули, создавая нужную пользователю конфигурацию ЭВМ. Например, у меня в конфигурацию входят модули ЦАП, АЦП и дискретного ввода-вывода для подключения к стенду. Рекомендации по написанию таких модулей находятся в отдельном документе build-own-modules.

В полном составе плата является одноплатной версией машины ДВК или Электроника-60 со встроенным терминалом. Вот пример экрана дисплея КСМ с загруженной RT-11:

```

LINE      CAPS      LAT      38400      00:00:28
@B
#RK■
DW Handler V6.5, (c) D.S.C., 1992-2016

?MY-I-MY/Dz-emulator handler. Un_Soft 1991. V3.04

RT-11FB (S) V05.04 D

.SET IT QUIET
?21
.DIR DX:

RT11SJ.SYS      84   03-Jan-99      SWAP .SYS      27   17-Nov-87
IT      .SYS      2   17-Nov-87      DX      .SYS      4   03-Jan-99
RK      .SYS      3   03-Jan-99      PIP     .SAV     30   17-Nov-87
DUP     .SAV     49   17-Nov-87      DUMP    .SAV      9   17-Nov-87
RESORC.SAV     26   17-Nov-87      DIR     .SAV     19   17-Nov-87
DATIME.SAV      3   01-Aug-90      STARTS.COM      1   18-Nov-99
SL      .SYS     14   20-Dec-85      MY      .SYS      3   02-Jan-99
DW      .SYS     15   02-Jan-99      K52     .SAV     54   17-Nov-87

 16 Files, 343 Blocks
143 Free blocks

.-

```

Поскольку быстродействие полученной ЭВМ гораздо выше прототипа, в схеме имеется возможность снизить тактовую частоту до 4.5 Мгц с помощью переключателя sw\_cpulow для работы с игрушками и другими интерактивными программами.

### 3. Используемые FPGA

Разработка сделана на основе FPGA серии Cyclone 4 фирмы Altera (увы, теперь это уже intel). В вышеуказанном составе схема занимает чуть меньше 9000 ячеек и влезает в микросхему EP4CE10. Если выкинуть часть дисковых контроллеров и КГД, то схема поместится в самую младшую микросхему этой серии – EP4CE6.

Из аппаратных особенностей Cyclone в схеме используется только статическая память (altsyncram) и генератор тактовых сигналов PLL. Поэтому схема легко может быть портирована на Cyclone других серий и вообще куда угодно, лишь бы хватило ресурсов.

В проекте используются следующие модули статической памяти altsyncram:

- rom000 — ROM размером 4096\*16 бит, предзагруженный ПЗУ 000 (rom/000.mif) — теневой монитор платы MC1201.01.
- rom055 — ROM размером 4096\*16 бит, предзагруженный ПЗУ 055 (rom/055.mif) или 279 (rom/279.mif) — теневой монитор платы MC1201.02.
- fontrom — ROM размером 32768\*1 бит, предзагруженный файлом экранного шрифта ksm-firmware/font/font-main.mif или ksm-firmware/font/font-ksm.mif..
- vtmem — RAM размером 2048\*16, предзагруженный терминальной микропрограммой ksm-firmware/terminal.mif
- kgdram — двухпортовая RAM размером 16384\*8, являющаяся видеопамью графического контроллера КГД. Второй порт этого модуля имеет 1-битную шину данных.
- userrom — ROM размером 4096\*16 бит для хранения образа ПЗУ пользователя.

Если вы используете FPGA серии, отличный от Altera Cyclone, то описатели PLL и модулей памяти придется переделать или создать заново.

Для удобства все файлы, относящиеся к мегафункциям, помещены в каталог ip-components, отдельный для каждой FPGA-платы. При портировании проекта на другую плату эти мегафункции будут зависеть от конкретной примененной FPGA.

### 4. Структура дерева файлов проекта

В дереве проекта имеются следующие каталоги:

- boards/ - хранилище проектов для каждой поддерживаемой FPGA-платы.
- doc/ - документация проекта
- disk/ - утилиты для чтения-записи образов дисков и тестовый дисковый банк
- hdl/ - verilog-файлы основного проекта
- hdl/ksm/ - verilog-файлы модуля KCM
- hdl/sdram\_ip - контроллер SDRAM
- rom/ - файлы образов ПЗУ
- ksm-firmware/ - файлы, относящиеся к микропрограмме KCM и экранные шрифты.
- screenshots/ - примеры текстовых и графических экранов

Главным модулем проекта является интерфейсный модуль, уникальный для каждой FPGA-платы. Этот модуль адаптирует ядро проекта к конкретной плате — определяет назначение переключателей и индикаторов, а также содержит в себе контроллер SDRAM, тактовый генератор PLL и схему управления VGA DAC.

Ниже интерфейсного модуля находится модуль topboard, лежащий в файле hdl/topboard.v. Этот модуль является ядром проекта — соединительной корзиной, к которой подключается

модуль процессора и набор периферийных устройств. Этот модуль представляет унифицированный внешний интерфейс, не зависящий от использованной платы FPGA. К его внешним модулям torboard портам подключаются:

- Тактовая частота - **clk\_p**, **clk\_n**, **sdclk** 12.5 МГц, **clk50** 50 МГц.
- 4 кнопки управления
- 4 переключателя
- 5 индикаторных светодиодов
- Интерфейс для доступа к ОЗУ (SDRAM)
- SD-карта
- сигналы VGA-монитора
- PS/2 клавиатура
- BUZZER (пищалка)
- Линии дополнительного UART
- Линии Centronics для принтера

Конечно, далеко не все из этих портов обязательно куда-то подключать. Например, принтер вряд ли кто-то будет реально использовать. Конкретное подключение этих портов определяется интерфейсным модулем и зависит от примененной платы FPGA.

## 5. Настройка конфигурации создаваемой ЭВМ

В корне проекта лежит файл config.v, с помощью которого настраивается состав проекта. В конфигурацию включается одна из процессорных плат, а также набор необходимых периферийных устройств. Кроме того, с помощью файла конфигурации можно настроить параметры некоторых компонентов — начальные скорости последовательных портов, тип экранного шрифта, параметры видеосигнала и т.д. Более подробная информация содержится в описании соответствующих модулей, а также в комментариях внутри самого файла config.v. Кроме глобальных настроек, можно также определять настройки конкретной процессорной платы — ее тактовую частоту, начальное состояние таймера, коэффициент замедления процессора, имя файла с образом ПЗУ, и т.д.

## 6. Внешние органы управления и индикации

К внешним портам схемы подключаются 4 кнопки, 3 переключателя и 5 светодиодов. В интерфейсе модуля torboard каждый из этих элементов имеет фиксированное символическое имя. Сопоставление их со светодиодами, кнопками и переключателями конкретной FPGA-платы производится на уровне интерфейсного модуля.

Назначение кнопок:

- **bt\_reset** – кнопка общего сброса. Инициализирует все подсистемы, кроме терминального модуля.
- **bt\_halt** – кнопка вызова пультового прерывания. При ее нажатии работа текущей программы прерывается, и ЭВМ переходит в пультовый режим. Вместо этой кнопки

можно установить двухпозиционный переключатель (программа-пульт), как это сделано в оригинальных ЭВМ. Тогда появится возможность пошагового прохождения программ.

- `bt_terminal_rst` – кнопка сброса терминального модуля, помогает при его зависании или просто для очистки экрана.
- `bt_timer` – включение-выключение таймерного прерывания. Каждое нажатие кнопки переключает режим таймера (вкл-выкл).

Назначение переключателей:

- `sw_diskbank` – эти два переключателя выбирают дисковый банк на SD-карте (описан далее в разделе “дисковая подсистема”).
- `sw_console` – переназначение консольного порта. Если этот переключатель включен, то к консольному порту ИРПС 1 подключаются внешние линии (`irps_txd`, `irps_rxd`), а аппаратный терминал переключается на дополнительный порт ИРПС 2. Это может быть использовано, например, если нужно в качестве консоли использовать компьютер с запущенной на нем терминальной программой.
- `sw_cpulow` — включение режима замедления процессора. При выключенном переключателе процессор работает на полной тактовой частоте. Если переключатель включен, то тактовая частота снижается до 4.5 МГц, что примерно соответствует тактовой частоте оригинальной платы. Этот режим необходим для игрушек и других слишком быстро работающих интерактивных программ. Плата MC1280 этот режим не поддерживает.

Назначение индикаторов:

- `rk_led` – обращение к диску RK
- `dw_led` – обращение к диску DW
- `dx_led` – обращение к диску DX
- `my_led` – обращение к диску MY
- `timer_led` – индикация состояния таймера (светится – таймер вкл).

В данном случае подразумевается, что:

- Кнопка в нажатом состоянии выдает 1, в отжатом 0
- Переключатель в отключенном положении выдает 0, во включенном – 1
- Светодиод светится при установке его порта в 0.

## 7. Интерфейсный модуль

Интерфейсный модуль производит адаптацию проекта к конкретной FPGA-плате. Он является посредником между соединительным модулем `topboard` и физическими портами FPGA. Кроме того, модуль содержит в себе PLL, контроллер SDRAM и схему управления цветовыми DAC интерфейса VGA.

Интерфейсный модуль разрабатывается индивидуально под каждую FPGA-плату. В дереве проекта уже имеются готовые модули для некоторых плат, их можно использовать как основу для создания собственного модуля при портировании на свою плату.

## 7.1. Контроллер SDRAM

Этот контроллер формирует блок оперативной памяти ЭВМ. В данный момент используется 16-битный адрес, позволяющий адресовать только 64К ОЗУ, однако в проект заложена возможность расширения адреса до 22 бит. Таким образом, в полном варианте модуль должен обеспечивать 4 мегабайта ОЗУ, но для полной реализации процессорных плат без диспетчера памяти (а они сейчас все такие) достаточно 64К.

Реализация контроллера зависит от конкретной микросхемы SDRAM, стоящей на FPGA-плате. В проект входит один из вариантов контроллера, позволяющий работать с 16-битными микросхемами SDRAM емкостью до 16 МБ, организованными как 4 банка по 4Мб. Основной временной параметр — Cas Latency, установлен равным 2. Если SDRAM поддерживает только CL3, то следует исправить значение `cas_latency` в файле `config.v`. Но тогда на один цикл обращения к памяти будет уходить 12 тактов.

Если у вас на плате установлена микросхема SDRAM другого типа, например DDR2, то контроллер придется делать самостоятельно.

Модуль `torboard` обеспечивает унифицированный интерфейс для работы с памятью. Сигналы этого интерфейса в основном соответствуют стандартному набору сигналов `wishbone`-шины:

- **`sdram_reset`** — сигнал сброса. По этому сигналу контроллер производит реинициализацию микросхемы SDRAM
- **`sdram_stb`** — строб начала обмена с памятью
- **`sdram_we`** — высокий уровень этого сигнала сообщает контроллеру, что идет операция записи.
- **`sdram_sel[1:0]`** — маска, определяющая, какие именно байты (старший, младший или оба) будут записываться. При чтении эта маска игнорируется, читаются всегда оба байта.
- **`sdram_ack`** — ответ от контроллера (`reply`), подтверждающий окончание транзакции.
- **`sdram_adr[21:0]`** — шина адреса
- **`sdram_out[15:0]`** — входная шина данных (из хоста в память)
- **`sdram_dat[15:0]`** — выходная шина данных (из памяти к хосту)
- **`sdram_ready`** — выходной сигнал, формируемый контроллером по окончании процесса инициализации памяти.

## 7.2. Тактовый генератор PLL

PLL должна выдавать 3 тактовых сигнала:

1. `clk_p` прямая фаза (0°) - основной тактовый сигнал
2. `clk_n` инверсная фаза (180°) - дополнительный тактовый сигнал процессорного модуля.
3. `sdclk` 12.5 Мгц прямая фаза (0°) - тактовый сигнал SD-контроллера и карты памяти.

Частота основного тактового сигнала зависит от процессорной платы и примененной FPGA, и задается в файле конфигурации. Соответственно, модуль PLL принимает 2 макроса — `PLL_MUL` и `PLL_DIV`, устанавливающих делитель и множитель основной тактовой частоты. Для Cyclone 4 это делается так:

```
altpll_component.clk0_divide_by = `PLL_DIV,  
altpll_component.clk0_duty_cycle = 50,  
altpll_component.clk0_multiply_by = `PLL_MUL,
```



```

altpll_component.clk0_phase_shift = "0",
altpll_component.clk1_divide_by = `PLL_DIV,
altpll_component.clk1_duty_cycle = 50,
altpll_component.clk1_multiply_by = `PLL_MUL,
altpll_component.clk1_phase_shift = "5000",
altpll_component.clk2_divide_by = 4,
altpll_component.clk2_duty_cycle = 50,
altpll_component.clk2_multiply_by = 1,
altpll_component.clk2_phase_shift = "0"

```

Файл описания pll — ip-components/pll.v

Собственный генератор платы должен обеспечивать частоту 50 МГц, которая используется PLL для синтеза вышеуказанных трех сигналов, а также напрямую терминальным модулем КСМ для формирования видеосигнала VGA. Если ваша плата имеет генератор другой частоты, то 50 МГц также придется синтезировать с помощью PLL.

### 7.3. Схема управления цветами VGA DAC

Модуль torboard обеспечивает 3 однобитных сигнала, включающих соответствующий цвет: vgared, vgreen, vblue.

В зависимости от подключения разъема VGA на конкретной FPGA-плате требуется создание схемы, преобразующей эти однобитные цвета в многобитную шину, поступающую на вход каждого из видео-ЦАП, например:

```

// выбор яркости каждого цвета
assign vgag = (vgagreen == 1'b1) ? 6'b111111 : 6'b000000 ;
assign vgab = (vgablue == 1'b1) ? 5'b11111 : 5'b00000 ;
assign vgar = (vgared == 1'b1) ? 5'b11111 : 5'b00000 ;

```

Если на вашей плате разрядность ЦАП другая, эту схему необходимо поправить под ваши нужды. Если ЦАП вообще отсутствует и линии R, G, B разъема VGA подключены непосредственно к портам FPGA, то преобразователь вообще не требуется.

## 8. Соединительный модуль верхнего уровня.

Модуль torboard — это, по сути, соединительная панель (корзина), к которой подключаются остальные модули. Именно так были устроены и оригинальные ЭВМ — процессор и вся периферия находится на отдельных платах, а корзина обеспечивает их питанием и соединяет в общую шину МПИ.

В данном проекте соединительная плата torboard обеспечивает общую шину Wishbone, а также содержит ряд вспомогательных компонентов:

- Генератор системных сбросов (DCLO и ACLO)
- Схема формирования сигналов выбора каждого периферийного устройства (xxx\_stb)
- Контроллер динамической памяти SDRAM
- Контроллер прерываний
- Арбитр DMA и коммутатор источника сигналов шины Wishbone.
- Арбитр доступа к SD-карте
- Формирователь сигналов управления видео-ЦАП VGA

- Коммутатор линий последовательных портов.
- ПЗУ с резидентной программой пользователя (отключаемое)

Состав включаемых в соединительную плату устройств определяется файлом конфигурации config.v. В конфигурацию обязательно включается одна процессорная плата, контроллер SDRAM, один последовательный порт (системная консоль). Все остальные компоненты являются необязательными и подключаются в схему при необходимости.

Соединительная плата обеспечивает унифицированный интерфейс, не зависящий от примененных FPGA. Согласованием с конкретной платой занимается интерфейсный модуль.

## 8.1. ПЗУ пользователя.

Как и в оригинальных микро-ЭВМ, в этом проекте предусмотрено подключение ПЗУ пользователя в адресное пространство 140000-157777 вместо соответствующего банка ОЗУ. В ЭВМ ДБК-1 в этом ПЗУ штатно размещался интерпретатор бейсика или фокала, что позволяло использовать ЭВМ в полностью бездисковой конфигурации. В остальных машинах на процессорной плате предусмотрена пустая колодка для установки ПЗУ пользователя.

В этом проекте для формирования этого ПЗУ предусмотрена мегафункция altsyncram с именем userrom. Для включения ПЗУ в конфигурацию в файле config.v надо раскомментировать строчку ``define userrom`, и указать имя mif-файла с загружаемым образом ПЗУ. В каталоге rom/ лежат готовые к использованию образы ПЗУ 013 (basic) и 058 (focal).

## 8.2 Генератор сбросов

Этот модуль вырабатывает согласованную пару сигналов, предназначенных для сброса процессорного ядра — ACLO и DCLO. Входным сигналом, приводящим к сбросу, является кнопка сброса sw[0], а также сигнал готовности PLL и SDRAM — процессор запускается только после выхода PLL и динамической памяти в рабочий режим.

Кроме сигнала сброса процессора, модуль также формирует сигнал сброса динамической памяти. Этот сигнал формируется только на основе кнопки и готовности PLL.

## 8.3 Контроллер прерываний

Контроллер прерываний реализует механизм векторных прерываний процессора. По сигналу запроса IRQ, выданному одним из периферийных устройств, модуль выдает процессору сигнал векторного прерывания VM\_VIRQ, затем выставляет на шину соответствующий устройству вектор прерывания, и передает прерывающему периферийному устройству сигнал подтверждения прерывания IACK.

Все адреса векторов передаются через входную шину IVEC, представляющую собой массив 16-разрядных чисел. В этот список можно дописывать и новые векторы, если в схему добавляются новые модули, требующие обслуживания прерывания. Параметр N модуля определяет количество обслуживаемых им векторов.

## 8.4. Арбитр DMA

Арбитр предназначен для обслуживания запросов прямого доступа к памяти (DMA), поступающих со стороны периферийных устройств.

Любое устройство на шине может взять управление шиной на себя, и производить обмен данными с памятью и другими периферийными устройствами без участия процессора. Для этого устройство выставляет запрос DMA\_REQ. В ответ на это запрос арбитр дожидается окончания текущей транзакции на шине, после чего снимает сигнал CPU\_GNT, запрещая тем самым процессору работать с шиной. Точнее говоря, этот сигнал отключает внутренний

таймер шины в процессоре и бесконечно растягивает следующую транзакцию. После этого арбитр выдает запрашивающему устройству ответ DMA\_ACK, по которому устройство берет на себя управление шиной. Шинные мультиплексоры отключают процессор от шины, и подключают вместо него запрашивающее устройство.

По окончании DMA-обмена устройство снимает сигнал DMA\_REQ, процессор вновь подключается к шине и продолжает работу.

## 8.5 Арбитр SD-карты.

Арбитр SD-карты позволяет всем дисковым контроллерам использовать одну и ту же SD-карту для хранения дисковых массивов. При этом каждому контроллеру выделяется отдельный участок в пространстве блоков SD-карты.

Дисковый контроллер, желающий получить доступ к карте, выставляет запрос SDREQ. Арбитр проверяет, не занята ли карта обменом с другим контроллером, и, если карта свободна, подключает ее к запрашивающему доступ контроллеру, и сигнализирует об этом ответом SDACK. По окончании работы с картой контроллер снимает запрос SDREQ, арбитр отключает карту от контроллера и переходит в состояние ожидания очередного запроса. Если на момент поступления запроса карта занята другим контроллером, запрос ставится в очередь и будет обслужен сразу после освобождения карты.

В схеме существует один дисковый контроллер, к которому карта подключается по умолчанию, если нет других активных запросов. Этот контроллер производит инициализацию карты. Таким контроллером по умолчанию может быть объявлен любой из дисковых контроллеров.

## 8.6. Формирователь сигналов управления видео-ЦАП VGA

Как правило, на отладочных FPGA-платах, имеющих VGA-разъем, установлен ЦАП для каждого из цветовых сигналов. Обычно такой ЦАП имеет 4-5 разрядов, и нужен для управления яркостью каждого из цветов. Поскольку в данном проекте видеосигналы не используют градации яркости, на видео-ЦАП подаются фиксированные значения — 0 для темной точки и некоторая величина (обычно — максимальная, типа 5'b11111) для светлой точки.

Схема формирователя зависит от разрядности ЦАП на конкретной плате и может потребовать коррекции при адаптации проекта. Вот пример формирователя для 6-битного зеленого и 5-битных красного и синего сигналов:

```
assign vgag = (vgagreen == 1'b1) ? 6'b111111 : 6'b000000 ;
assign vgab = (vgablue == 1'b1) ? 5'b11111 : 5'b00000 ;
assign vgar = (vgared == 1'b1) ? 5'b11110 : 5'b00000 ;
```

Шины vgar, vgag, vgab — это выходные порты модуля topboard, подключаемые к видео-ЦАП на плате.

## 8.7. Коммутатор линий последовательных портов.

Это устройство управляет подключением выходных линий модулей ИРПС1 и ИРПС2. По умолчанию, ИРПС1 подключен к терминальному модулю КСМ, а ИРПС2 — к выходным портам платы irps\_txd и irps\_rxd. С помощью переключателя sw[2] есть возможность поменять эти сигналы местами — тогда ИРПС2 подключается к терминалу, а ИРПС1 — к внешним портам. Это нужно, например, для вывода консольного терминала на последовательный порт РС, чтобы использовать в качестве консоли запущенную на РС терминальную программу.

Кроме этого, данное устройство управляет скоростью последовательных портов. Скорость ИРПС, подключенной к терминалу КСМ, определяется самим модулем КСМ (и может изменяться кнопками F5/F6 клавиатуры КСМ). Скорость другого порта ИРПС фиксирована и задается в файле конфигурации config.v.

## 9. Процессорные платы.

В состав проекта включены 4 процессорные платы:

Плата	Процессор	ЭВМ	Тактовая частота
МС1201.01	K1801BM1	ДВК-1,ДВК-2	100 МГц
МС1201.02	K1801BM2	ДВК-3	100 МГц
МС1260	M2 (LSI-11)	Электроника-60	100 МГц
МС1280	M4 (LSI-11M)		50 МГц

В схему может быть включена только одна плата из всех возможных.

Каждая процессорная плата, кроме собственно процессорного ядра, также содержит некоторые вспомогательные компоненты:

- Интервальный таймер, вызывающий прерывания процессора с частотой 50 Гц.
- Схему управления таймером.
- Схему замедления процессора, позволяющую снизить производительность процессора до уровня оригинальной платы, не изменяя тактовую частоту общей шины.
- Теневое ПЗУ с отладочным монитором и набором загрузчиков.

Интервальный таймер генерирует импульс шириной в один период тактовой частоты шины, повторяющийся с частотой 50 Гц. Этот сигнал подается на вход таймерного прерывания процессора и используется для организации системных часов. Схема управления таймером позволяет включать и выключать эти прерывания кнопкой sw[3] (как и в оригинальных микроЭВМ — переключатель или кнопка «таймер»).

Кроме таймерного прерывания, предусмотрено прерывание пультового режима, вызываемое по кнопке sw[1]. В оригинальных микро-ЭВМ эту функцию выполнял переключатель «программа-пульт». По этому прерыванию прекращается работа основной программы, и управление передается встроенному в ПЗУ или микрокод процессора монитору-отладчику.

### 9.1. Плата МС1201.01

Эта плата основана на процессоре КР1801ВМ1. Процессор не поддерживает команд FIS, а из EIS поддерживает только команду MUL, из-за чего RT-11 ошибочно полагает наличие полной поддержки EIS.

Плата содержит 64Кб ОЗУ, разбитых на 8 банков, причем банк 7 исключен из адресного пространства пользователя (на его месте находится страница ввода-вывода)

На плате имеется специальный регистр начального пуска, содержимое младших двух бит которого определяет режим запуска ЭВМ. Возможен запуск с выходом в монитор-отладчик, автозагрузка с гибкого диска, или запуск программы из ПЗУ пользователя. Другие биты этого регистра управляют картой адресного пространства, и определяют начальный адрес запуска (адрес расположения ПЗУ монитора в адресном пространстве).

По умолчанию, после включения питания, прерывания от таймера разрешены, и нажатие кнопки сброса не изменяет состояние таймера.

ПЗУ монитора содержит единственный загрузчик с гибких дисков DX (ГМД-70). Загрузка возможна с любого из дисков DX0 или DX1 и производится командами D0 и D1. Загрузиться с остальных типов дисков можно только из-под уже загруженной с DX операционной системы, или введя загрузчик в память командами монитора.

На этой плате основывались микроЭВМ ДВК-1 и ДВК-2. В качестве ПЗУ с монитором использовалось ПЗУ 1801PE2-000. ДВК-1 штатно комплектовались дополнительным пользовательским ПЗУ с Бейсиком (1801PE2-013) или Фокалом (1801PE2-057). Образы всех этих ПЗУ лежат в каталоге [g0m/](http://g0m/).

В отличие от ДВК-3, машины ДВК-1 или ДВК-2 комплектовались не контроллером КСМ, а аппаратным фрязинским терминалом 15ИЭ-00-013. Однако и КСМ, и фрязинский дисплей полностью совместимы с DEC VT52, и именно в этом режиме эксплуатировалось подавляющее большинство программного обеспечения. Те, кому нужен именно фрязинский дисплей, могут заменить контроллер КСМ на аппаратный терминал <https://github.com/forth32/vt52>. При этом потеряется возможность использовать графический модуль КГД.

## 9.2. Плата MC1201.02

Эта плата основана на процессоре KP1801BM2. Процессор полностью поддерживает набор команд EIS, а команды FIS эмулирует с помощью подпрограмм теневого ПЗУ.

Как и предыдущая, эта плата содержит 64Кб ОЗУ, из которых пользователю доступны 56К.

Также имеется регистр начального пуска, но режим запуска определяется уже тремя битами, и, соответственно, имеется 8 режимов запуска. Процессор 1801BM2 имеет полноценную аппаратную поддержку теневого режима, поэтому специальных регистров для управления картой памяти не требуется. Режим процессора определяется сигналом SEL, в wishbone-ядре представленным в виде дополнительной адресной линии `wb_adr[16]`.

По умолчанию, после включения питания, прерывания от таймера разрешены, и нажатие кнопки сброса не изменяет состояние таймера.

На плате штатно устанавливалось ПЗУ 1801PE2-055. Существует также обновленная версия этого ПЗУ — 279, имеющая расширенный набор команд и дополнительные загрузчики. Каждое из этих ПЗУ имеет в себе богатый набор загрузчиков и поддерживает загрузку с любого из имеющихся в проекте дисков (за исключением загрузчика МУ, имеющегося только в ПЗУ 279). Загрузка системы производится командой В, в ответ на приглашение «В» следует набрать имя и номер загружаемого устройства.

В обеих версиях ПЗУ содержится ошибка, в результате которой загрузка с диска RK становится невозможной. Суть этой ошибки в неправильной длине области, читаемой с диска при начальной загрузке. Когда вводится команда монитора В RK, монитор копирует код начальной загрузки в память по адресу 1000 и передает туда управление. Этот загрузочный код запускает чтение диска RK, начиная с сектора 0, но читает не один 512-байтовый блок, а два — в этом и заключается ошибка. В результате код загрузчика, начиная с адрес 1000, затирается в режиме DMA данными сектора 1, и по окончании обмена с диском управление получает мусор вместо загрузочного кода. Я внес исправления в обе версии ПЗУ, и в каталоге [g0m](http://g0m/) лежат уже исправленные версии.

На этой плате основана микро-ЭВМ ДВК-3.

## 9.3. Плата MC1260

Эта плата основана на секционном процессоре M2 (его прототип — DEC LSI-11). Процессор этот имеет внешнее ПЗУ микрокода, и позволяет подключать дополнительный пользовательский микрокод. Монитор-отладчик также реализован на уровне микрокода, и

дополнительные ПЗУ в адресном пространстве отсутствуют. Соответственно, нет ни теневого режима процессора, ни каких-либо переключений карты памяти. Процессор аппаратно поддерживает наборы команд EIS и FIS.

Как и на всех других платах, нижние 56Кб адресного пространства заняты модулем динамической памяти, а выше находится страница ввода-вывода.

Плата имеет 4 режима пуска, из которых практическую пользу имеет только один режим — запуск монитора-отладчика.

По умолчанию, после включения питания, прерывания от таймера запрещены, и нажатие кнопки сброса запрещает прерывания от таймера. Это сделано потому, что разрешенные таймерные прерывания вызывают сбой в загрузке операционной системы. Таймер можно включить после окончания загрузки ОС для работы системных часов.

Встроенный в микрокод монитор не имеет каких-либо загрузчиков, кроме перфоленточного, и загрузку с диска можно произвести только введя в память дампы загрузчика, используя команды монитора.

На этой плате основана микро-ЭВМ Электроника-60, которая, в основном, использовалась в различных управляющих системах. В качестве полноценной настольной ЭВМ она использовалась крайне редко, если вообще когда-либо использовалась.

## 9.4. Плата MC1280

Эта плата основана на секционном процессоре M4 (прототип — DEC LSI-11M). В отличие от всех предыдущих плат, это процессорное ядро работает на более низкой тактовой частоте — для Cyclone 4 используется 50 Мгц. В остальном эта плата схемотехнически аналогична MC1260.

Какие ЭВМ собирались на этой плате — мне неизвестно.

## 10. Дисковая подсистема

Все дисковые контроллеры хранят свои данные на SD-карте. Поддерживаются только SDHC-карты объемом до 64Гб. Карты SDSC для упрощения схемы не поддерживаются — таких карт теперь уже не делают и достать их проблематично. Следует учесть, что многие дешевые китайские SDHC-карты (продающиеся на алиэкспрессе за копейки) некорректно работают в режиме SPI. Некорректность выражается в полном зависании карты при первой же команде записи. Карта перестает реагировать на входящие команды, и вывести ее из этого состояния можно только снятием питания. При этом чтение с карты работает без всяких проблем. Фирменные карты — transcend, kingston, samsung итд, такого бага не имеют.

### 10.1. Распределение адресов на SD-карте

Карта распределения адресов блоков SD-карты под образы дисков, используемая в данном проекте, описана в файле card-layout.ods.

Все четыре дисковых контроллера разделяют одну и ту же карту. Набор из 8 дисков RK, одного диска DW, двух дисков DX и 4 дисков MY образует дисковый банк размером 128 Мб. Часть этого пространства для дисков не используется и пропадает ради упрощения формирователей адресов. Поскольку такой объем — мизер для современных карт, то дисковых банков на карте можно разместить несколько. В данной схеме поддерживаются 4 банка, переключаемые с помощью sw[1:0], но при желании минимальным изменением схемы этих банков можно наплотить сколько влезет на карту.

Каждый модуль дискового контроллера имеет входную шину «start\_offset», которая определяет, с какого абсолютного номера блока начинаются образы дисков данного типа.

Изменяя это значение, можно перемещать массив дисков в любое место SD-карты. Можно сделать переключаемые банки для каждого типа дисков отдельно, можно даже вынести любой тип диска на отдельную SD-карту. Все определяется вашими потребностями. Главное не забывать, что начальный адрес массива дисков должен быть кратен сумме размеров образов дисков всех устройств, подключенных к данному контроллеру. Например, образ диска MY занимает на карте 2048 блоков, следовательно размер 4 дисков будет равен 8192 блока. Начальный адрес дискового массива должен быть кратен 8192.

Для записи образов дисков на SD-карту используется программа sd-store, а для извлечения образов с карты – программа sd-extract. Обе программы используют контейнеры DSK (посекторные образы дисков), которые распознаются эмулятором simh. Исходные тексты этих программ лежат в каталоге disk/. Карта распределения адресов блоков хранится в файле devtable.h.

Также, для облегчения начального запуска проекта, в каталоге disk лежит полный образ дискового банка в файле initdisk.img. Этот образ содержит в себе:

- полный дистрибутив RT-11 v5.4 на диске DW0,
- загрузочный диск RT-11 v5.4 на RK0 и DX0,
- загрузочный образ XXDP 2.5 на диске RK1,
- загрузочный диск RT-11 v5.1 и набор игр и графических программ на MY0
- ФОДОС-ТМОС на диске MY1 и RK2.

В систему встроен драйвер DW, разбивающий HDD на 4 одинаковых диска по 16 Мб (можно при желании переразбить диск командой set dw part). На дисках RK0 и DW0 лежат исходные тексты драйвера DW.MAC и командный файл DWBLD.COM для сборки драйвера под конкретную систему.

Стартовый набор initdisk.img можно прямо записать на SD-карту с нулевого блока командой

```
dd if=initdisk.img of=/dev/sdx
```

где sdx – имя устройства SD-карты, например /dev/sdc. Установив записанную карту в плату, можно произвести загрузку системы командой монитора B с устройств DX0, RK0, RK1, MY0.

## 10.2. SDSPI

Для работы с картой памяти используется модуль SDSPI – контроллер SD-карты. За его основу взят одноименный модуль из проекта rdp2011, переписан с мерзкого VHDL на кошерный verilog и переделан под мои потребности. Модуль содержит в себе два 512-байтовых буфера – буфер чтения и буфер записи и умеет производить 2 простейших операции – прочитать блок с карты в буфер и записать блок из буфера на карту. Работа с буферами происходит на полной скорости общей шины (wb\_clk), а операции ввода-вывода на SD-карту вообще шину не занимают и выполняются асинхронно по отношению к процессору. Это позволяет совместить ввод-вывод и работу основной программы, как и в реальной ДВК.

Модуль может работать в двух режимах — ведущего и ведомого. Ведущий контроллер получает доступ к карте по умолчанию, сразу после запуска. Он производит первичную инициализацию SD-карты. Ведущий SDSPI подключается только к одному дисковому контроллеру из всех. К остальным дисковым контроллерам подключается SDSPI в режиме ведомого — в этом режиме инициализация карты не производится. Режим работы sdspi каждого дискового контроллера задается через порт sdmode (0 — ведомый, 1 — ведущий).

Любой из дисков RK, DX, MY или DW также можно вынести на отдельную SD-карту. Для этого порту sdmode контроллера следует присвоить значение 1, а порты sdcard\_cs, sdcard\_mosi, sdcard\_miso, sdcard\_sclk подключить к соответствующим выводам отдельной sd-карты. Линии sdreq (запрос доступа к карте) и sdack (подтверждение доступа) в этом случае надо просто замкнуть друг на друга.

### 10.3. Модули дисковых контроллеров.

Каждый дисковый контроллер представлен отдельным модулем, подключаемым к общей шине wishbone и диспетчеру SD-карты (или реальной карте). Имеются следующие дисковые модули:

- rk11.v — контроллер дисков RK11 (RK:)
- rx01.v — контроллер 8-дюймового дисководов RX01 (DX:)
- fdd-my.v — контроллер 5-дюймового дисководов (MY:)
- dw.v — контроллер жесткого диска RD50C (DW:)

Любой из этих контроллеров можно безболезненно выкинуть из схемы за ненадобностью.

### Сводная таблица дисковых контроллеров

	<b>RK</b>	<b>DX</b>	<b>MY</b>	<b>DW</b>
Емкость, блоков	4872	475	1600	131072
Размер на SD	6144	4096	2048	131072
Число приводов	8	2	4	1
Address	177400-177416	177170-177172	172140-172142	174000-174026
Vector	220	264	170	300
DMA	*		*	
CYL	312	76	80	1024
HD	2	1	2	8
SPT	10	25	10	16
Размер сектора	512	128	512	512

Пояснение к таблице.

- Емкость — это сколько логических блоков по 512 байт влезает на устройство
- Размер на SD — сколько виртуальный образ занимает места на SD-карте. Часть этого пространства может не использоваться и содержать мусор.
- Число приводов — сколько виртуальных приводов подключено к контроллеру.
- DMA — использует ли контроллер DMA для чтения-записи буферов
- CYL — число цилиндров эмулируемого привода
- HD — число головок
- SPT — количество секторов на одну дорожку
- Размер сектора — размер физического сектора, обрабатываемого данным контроллером. Параметр SPT отражает количество на дорожку секторов именно этого размера. Система RT-11 всегда использует логические блоки размером 512 байт, и именно в блоках такого размера вычисляется первый параметр, емкость диска.

Далее подробно описаны каждый из дисковых модулей.



### 10.3.1. Контроллер RK11/RK05

Этот модуль реализует контроллер DEC RK11 (RK: в rt-11), с подключенными к нему 8 дисками RK05 объемом 3 Мб. Советским аналогом данного устройства является СМ-5300. Данный дисковый контроллер является одним из самых быстрых – он использует DMA для чтения-записи дисковых буферов. Контроллер этот был реализован самым первым, и уже полгода трудится в нашей лаборатории в качестве рабочего диска.

Размер дискового блока RK05 – 512 байт, что прекрасно соответствует размеру блока SD-карты. Поэтому дисковые образы на SD-карте представляют собой посекторную копию реального диска – каждый логический номер блока LBN совпадает с физическим номером блока от начала образа. Эти образы можно напрямую использовать в simh, установив формат RAW (командой `set RKn format=raw`).

В реальной жизни диски СМ-5300 крайне редко подключали к ДБК и другим настольным компьютерам — чаще они встречались в составе больших ЭВМ, например СМ-4. Но этот диск — самый быстрый и удобный для реальной работы. Скорость обмена с этим диском такая же, как у дискеты MY (в данной реализации, конечно), а емкость — гораздо больше.

### 10.3.2. Контроллер RD50C

Контроллер RD50C (DW: в RT-11) обеспечивает интерфейс с жесткому диску объемом 64М – это максимальный объем, поддерживаемый драйвером DW. Оригинальный контроллер RD50C фирмы DEC был предназначен для работы в машинах DECpro-350, и использовал специфическую для этой машины систему прерываний. В варианте ДБК контроллер работает с обычной системой векторных прерываний, как все другие устройства, и именно в этом варианте реализован данный модуль. Поэтому фирменный драйвер DW.SYS из комплекта RT-11 не будет работать с этим контроллером – требуется драйвер, написанный именно под ДБК. Их много разных вариантов, можно использовать любой из них. Но следует учесть, что максимальный объем диска, поддерживаемого файловой системой RT-11 – 32Мб, поэтому все 64Мб виртуального жесткого диска можно использовать только если драйвер умеет разбивать диск на несколько разделов. Я использовал драйвер, который называет себя “DW Handler V6.5, (c) D.S.C” - он умеет разбивать HDD на 4 логических диска.

Обратите внимание, что с образом жесткого диска будет корректно работать только тот драйвер, под управлением которого жесткий диск был разбит на разделы. В отличие от мира IBM PC, здесь нет стандартных форматов разбивки диска, и таблица разделов чаще всего хранится не на диске, а в самом драйвере. При использовании чужого драйвера (или родного, но с другой настройкой разделов) результат непредсказуем — от потери доступа к данным и невозможности загрузки ОС до полного разрушения информации на диске. Поскольку на имеющихся в сети образах дискет часто попадаете драйвер DW.SYS, будьте внимательны при загрузке с таких дискет.

В отличие от RK11, этот контроллер не умеет работать через DMA – чтение/запись буферов производится программно, по одному слову. Поэтому контроллер DW существенно медленнее контроллера RK. В нашем стенде диск DW используется только как общая свалка файлов, а вся основная работа идет только с дисками RK.

Размер сектора HDD – 512 байт, как и у sd-карты. Как и в случае RK11, логический номер блока виртуального HDD соответствует абсолютному номеру блока от начала образа диска. Тем не менее, если диск разбит на несколько разделов, то работать в simh можно только с самым первым разделом диска – остальные будут недоступны, ибо ДБКшный драйвер DW.SYS не будет работать в simh. Под simh надо монтировать образ как MSCP-диск DU:, не забыв указать `format=raw`.

Этот модуль имеет одну проблему – с диска DW нельзя загрузиться аппаратно, из-под монитора командой “B DW”. Вылезает сообщение “?BOOT-U-I/O ERROR”. Почему так происходит, я не разобрался, но глубоко в эту проблему не вникал. ПЗУ 055, которые стояли на реальных ДВК, с которыми мне приходилось работать, вообще не умеет делать загрузку с DW. Пзу 279 имеет такой вариант загрузки, но он не работает. Более того, командой BOOT DW: из RT11 можно произвести загрузку только FB-монитора из-под FB-монитора. Загрузить SJ-монитор не получается, загрузить FB-монитор из-под SJ-монитора тоже не получается. Возможно, я что-то упустил в аппаратной реализации контроллера, но под управлением драйвера DW контроллер работает без всяких проблем, чтение-запись идут без ошибок. Этот вопрос придется изучить отдельно.

### **10.3.3. Контроллер RX11/RX01.**

Модуль реализует контроллер RX11 с подключенными к нему двумя дисковыми RX01. Нашим аналогом такого дискового является ГМД-70. Конечно, времена 8-дюймовых дискет прошли уже четверть века назад, и дискет таких уже, наверное, в природе почти не осталось. Но этот контроллер был встроен в плату MC1201.02, и для полноты картины я реализовал и его. Практической пользы от него совсем немного, в основном он нужен для работы с попадающимися в сети образцами DX-дискет.

Контроллер этот так же как и DW не умеет работать через DMA. Что еще хуже, он не поддерживает словный обмен по шине – передача данных в/из буфера происходит побайтно, что в два раза замедляет процесс загрузки-выгрузки буфера. Размер сектора этого диска всего 128 байт, и логический блок RT-11 соответствует 4 секторам дискеты. Общий размер диска – 251 Кб, но дорожка 0 не используется системой для хранения данных.

Для упрощения схемы и экономии ресурсов FPGA я не стал уплотнять данные, и образ диска DX на SD-карте занимает 4Мб. При этом данные хранятся только в младших байтах блока SD-карты, и используется только половина каждого блока (128 слов, каждое из которых хранит 1 байт данных и 1 байт мусора). Геометрия диска – 76 дорожек и 25 секторов на дорожку, причем сектора имеют номер 1-26, а сектора 0 не существует. Но поскольку под хранение номера цилиндра отводится 7 бит, а номера сектора – 5 бит, то образ, лежащий на SD-карте, содержит 128 цилиндров и 32 дорожки. Неиспользуемые поля данных могут содержать что угодно – контроллер эту информацию игнорирует.

Поскольку образ диска на SD-карте лежит в довольно извратном формате, для извлечения посекторных образов DX-дисков и записи их обратно на карту в утилиты sd-store и sd-extract введено преобразование форматов. Утилита sd-extract выдает на выходе образ, подходящий для монтирования в simh к устройству gx. Утилита sd-store записывает такой образ на карту в формате, пригодном для использования контроллером.

### **10.3.4. Контроллер НГМД МУ**

Это, наверное, самый сложный и развитый контроллер флоппи-дисков из всех, применявшихся в ДВК. Это чисто советская разработка, не имеющая DEC-аналога. В фирменной документации контроллер называется громоздкой аббревиатурой «ЩИЗ.057.136», но мне не припоминается, чтобы кто-нибудь в реальной жизни его так называл, все его называют просто МУ по имени обслуживающего его драйвера MY.SYS.

Контроллер рассчитан на работу с двумя дисковыми НГМД-6022 или 6121. Каждый дисковод — двухдисковый, 6022 поддерживает 40 дорожек (емкость дискеты 400 Кб), а 6121 — 80 дорожек (800 Кб). Дисковод имеет 2 головки. Для передачи данных контроллер использует DMA, что дает производительность не хуже чем у RK11.

Оригинальный контроллер может работать с секторами размером 256, 512 и 1024 байта. Формат конкретного сектора хранится в его заголовке, записанным на дорожку дискеты перед полем данных. Поскольку никакой служебной информации в DSK-контейнере не имеется, то данная реализация контроллера поддерживает только 512-байтные сектора (10

секторов на дорожку), чего более чем достаточно для работы с образами дискет от ДВК. Также имеются следующие функциональные отличия от реального контроллера:

- Команды «форматирование» и «чтение заголовков» не поддерживаются. Для виртуального диска на SD-карте эти операции не имеют смысла.
- Команды «чтение с меткой» и «запись с меткой» работают как обычные команды чтения-записи. Метки хранить негде, и вообще мне как-то не встречался софт, использующий работу с помеченными секторами.
- Контроллер эмулирует только 80-дорожечный дисковод. При этом образы 40-дорожечных дискет, записанные на SD-карту, будут корректно читаться и записываться, но создать пустой образ 40-дорожечной дискеты средствами контроллера нельзя.
- Контроллер в режиме DMA ограничен 16-битным адресным пространством. Оригинальный контроллер может работать с 22-битным адресом, но реализация этой возможности бессмысленна — в MC1201.02 шина адреса 16-битная. Впрочем, при желании контроллер легко доработать до полной 22-битной шины, если такая потребность когда-нибудь возникнет.

В остальном поведение контроллера более-менее соответствует его реальному прототипу. Проблем с чтением, записью файлов, созданием файловой системы и загрузкой ОС не выявлено.

## 10.4. Ручная загрузка операционной системы с дисков

Не все платы содержат в себе загрузчики со всех имеющихся дисковых устройств. Полный набор загрузчиков есть только в мониторе 279 платы MC1201.02. Монитор платы MC1201.01 умеет загружать систему только с дисков DX, а платы MC1260 и MC1280 вообще не содержат ни одного загрузчика.

Тем не менее можно, как в старые добрые времена, произвести загрузку с любого диска с помощью команд монитора, работающих с ячейками памяти. Для устройств, работающих в режиме DMA, достаточно ввести код команды в регистр CSR и параметры команды в некоторые другие регистры. Для устройств, использующих программный обмен, в ОЗУ вводится программа начальной загрузки и запускается на выполнение.

### 10.4.1 Загрузка с RK

Команды для загрузки с диска RK0:

```
@177406/000000 177400^  
177404/000200 5 <CR>  
@0G
```

Первая команда записывает счетчик слов. 177400 — это 1 сектор диска. Вторая команда запускает процесс чтения. Сектор читается с диска и размещается в ОЗУ с адреса 0. Третья команда передает туда управление.

### 10.4.2. Загрузка с диска MY

Дисковый контроллер MY имеет специализированную команду для загрузки ОС. Процесс загрузки:

```
@172140/000040 37 <LF>  
172142/000000 <CR>  
@0G
```

Первая команда переводит контроллер в режим загрузки, вторая — устанавливает номер загрузочного дисковода — 0. После выполнения этих двух команд в память с адреса 0 считывается загрузочный сектор. Третья команда передает туда управление.

### 10.4.3. Загрузка с диска DX

Этот контроллер не использует DMA, поэтому простым набором команд тут не обойтись. Для загрузки с этого диска придется ввести в память программу начальной загрузки. Программа довольно длинная, и приводить ее здесь я не буду. При вводе этой программы с консоли достаточно ошибиться в одной цифре — и никакой загрузки не получится. Кому это действительно надо — легко могут найти в сети дампы загрузчиков с RX01.

## 11. Контроллер ИРПС

Контроллеры ИРПС реализуют линию связи в последовательном коде со скоростью от 1200 до 115200 бит/с. В проект входят два контроллера:

ИРПС1, по адресу 177560, является консольным терминалом, через который монитор и операционная система общаются с пользователем. Этот порт обязательно должен присутствовать в схеме. В данном проекте порт состыкован с терминалом КСМ, но это не обязательно — можно вывести его сигналы наружу и пользоваться внешним терминалом. Скорость порта ИРПС1 управляется модулем КСМ (если он есть), начальная скорость задается параметром `TERMINAL_SPEED` в файле конфигурации.

ИРПС2, по адресу 176500 — это дополнительный последовательный интерфейс, который можно использовать для многих целей. Можно подключить к нему дополнительный терминал (RT-11 умеет работать в многотерминальной конфигурации). Можно подключить принтер с последовательным портом — это гораздо проще, чем делать интерфейсный адаптер для centronics. В этом случае принтер будет работать через устройство LS: в RT-11. Также можно его использовать для передачи файлов между ЭВМ. В данной схеме этот интерфейс использует нестандартный вектор прерывания — 330-334 вместо 300-304. Это связано с тем, что вектор 300 занят контроллером HDD DW:. При генерации системы следует указать правильный адрес вектора для устройств LS, XL и дополнительного терминала.

Скорость ИРПС2 фиксирована и определяется параметром `UART2SPEED` в файле конфигурации.

Есть возможность поменять местами оба порта с помощью переключателя `sw[2]`. Тогда ИРПС2 подключается к модулю КСМ, сигналы ИРПС1 выводятся наружу, а скорость ИРПС1 фиксируется.

## 12. Контроллер ИРПР

Этот модуль реализует половину интерфейса ИРПР — только передатчик. Оригинальный ИРПР, стоящий на плате MC1201, умеет еще и вводить данные по 16-битной шине, но я ни разу не видел, чтобы хоть кто-то этим пользовался. У всех этот порт использовался исключительно для подключения принтеров. В RT-11 порт обслуживается драйвером LP:.

Поскольку настоящие ИРПР-принтеры на сегодня вымерли как динозавры, внешний интерфейс модуля реализует протокол centronics, что в принципе позволяет подключить любой пишущий принтер с параллельным интерфейсом. Естественно, придется делать интерфейсную схему, преобразующую уровни FPGA 3.3v в стандартный для centronics 5V, и буферизующую выходные линии данных. Я глубоко сомневаюсь, что кто-то реально захочет подключить физический принтер, но модуль вполне работоспособен и пусть остается в проекте. Кому он не нужен — могут выкинуть его из схемы и сэкономить примерно 50 ячеек FPGA.

## 13. Терминальная подсистема

Набор видеоконтроллеров ЭВМ ДВК-3 представляет собой забавную смесь совсем древних и более современных технологий. Набор состоит из двух плат – текстового терминала КСМ и графического видеоконтроллера КГД. В данном проекте реализованы оба контроллера.

В оригинальной ЭВМ для отображения информации использовался монохромный видеомонитор МС6105, а для ввода – клавиатура МС7004. В данном проекте для отображения используется любой монитор с VGA-входом, для ввода – клавиатура PS/2. Контроллер использует видеорежим 800\*600\*75 Гц. Этот режим в обязательном порядке поддерживают все мониторы, и ЭЛТ, и современные LCD, даже 4K мониторы могут такое отобразить. Если у монитора нет VGA-входа, можно воспользоваться переходником VGA2HDMI – проверено, работает. Для мониторов, некорректно размещающих на экране картинку (сдвиг изображения за левый край) предусмотрена коррекция фазы строчного синхроимпульса, задаваемая в файле конфигурации.

### 13.1. Контроллер КСМ

Этот контроллер представляет собой полноценный алфавитно-цифровой терминал, примерно совместимый с DEC VT52. Контроллер устанавливается в корзину ДВК в виде полноразмерной платы, однако не использует шину МПИ – от корзины он получает только питание. Передача данных производится через последовательный порт, подключаемый перемычкой к разъему ИРПС платы МС1201. Такой способ подключения консоли оператора к ЭВМ использовался еще во времена телетайпов и пишущих машинок, и более характерен для больших ЭВМ, а не персональных. Плата КСМ поддерживает скорости обмена от 75 до 9600 бод. В данном проекте диапазон поддерживаемых скоростей изменен – поддерживаются скорости от 1200 до 115200 бод. Скорости ниже 1200 не имеют практического смысла, а скорости выше 9600 оригинальный контроллер, видимо, не мог поддерживать в силу низкой производительности.

Экран терминала состоит из 25 строк по 80 символов. Верхняя строка – служебная, туда выводится информация о настройках и режимах работы терминала. Остальные 24 строки содержат пользовательские данные. Это соответствует терминалу DEC VT52 – там тоже использовался экран 80\*24.

#### 13.1.1. Внутреннее устройство контроллера

Контроллер представляет собой полноценное микропроцессорное устройство. Оригинальный КСМ был построен на процессоре K580BM80A и нескольких БИС из серии 580. В контроллере имеется блок статической рабочей памяти емкостью 1К, ПЗУ с управляющей программой емкостью 2К, ПЗУ знакогенератора емкостью 2К и видеопамять емкостью 16К, выполненная на микросхемах DRAM K565PY6. Видеопамять используется довольно бездарно – половина ее предназначена только для хранения признака наличия курсора в данном знакоместе. В остальной части видеопамати организован буфер для хранения 2 страниц текста – 48 строк по 80 символов.

В данном проекте контроллер построен на процессорном ядре K1801BM2 от уважаемого VSLAV. Писать управляющую программу под систему команд PDP-11, да еще и с EIS, гораздо приятнее и проще, чем под убогий Intel 8080. Кроме процессорного ядра, в контроллер входит модуль RAM размером 4К, предназначенный для хранения управляющей программы и рабочих переменных, модуль ROM размером 4К для хранения шрифтов и двухпортовый модуль RAM размером 2К в качестве видеопамати. Таким образом, общий размер используемой памяти существенно сокращен по сравнению с оригиналом.

### 13.1.2. Поддерживаемые режимы работы

Управляющая программа полностью поддерживает систему команд оригинального контроллера, в том числе режим HOLD SCREEN и нестандартные ESC-последовательности “раздвижка строки” и “раздвижка страницы”.

Поддерживается псевдографика, причем она отличается от оригинальной псевдографики DEC VT52 как по форме знаков, так и по их кодам (100-137 вместо 137-177).

Реализован автономный режим работы (локальная петля), включающийся кнопкой F10. В этом режиме вводимые с клавиатуры символы и команды сразу попадают на экран, а обмен по интерфейсу не производится. Этот режим можно использовать для проверки и отладки клавиатуры и дисплейного модуля. Такой режим имеется и у оригинального КСМ.

Реализована поддержка звукового сигнала (пищалки), как во всех терминалах DEC, а также в 15ИЭ-00-013. На плате КСМ пищалки нет, но многие программы, и сама ОС RT-11, пользуются звуковым сигналом (передавая код 007), а реализовать эту возможность совсем нетрудно. В управляющую программу добавлена возможность отключения звука кнопкой F9. Ну а кому звук совсем не нужен – просто никуда не подключайте порт buzzer.

Полудуплексный режим (локальное эхо) не реализован в силу полной бесполезности. Не реализована и смена формата кадра последовательного интерфейса – передача всегда идет в режиме 8 бит без контроля четности.

Так же не реализован режим автоматического переноса курсора на следующую строку. Этого режима нет в DEC VT52, и практической пользы от него мало. Наоборот, в этом режиме нарушается работа полноэкранных текстовых редакторов и других подобных программ. Но, при желании, доделать этот режим в управляющую программу совсем не сложно.

### 13.1.3. Кодировка символов.

Терминал работает в 7-битном режиме. Формат кадра последовательного порта 8-битный (8-N-1), но старший бит байта игнорируется. Поддерживаются 2 набора кодировок символов — КОИ7 Н0/1, и КОИ7 Н2.

В режиме КОИ7 Н0 (режим по умолчанию) терминал работает со стандартной таблицей ASCII, в точности как оригинальный VT52. При этом отображаются малые и большие латинские буквы. Из кодировки Н0 (LAT) можно переключиться в кодировку Н1 (RUS), при этом терминал отображает малые русские буквы вместо больших латинских, и большие русские вместо малых латинских. На экране может отображаться смесь символов обеих кодировок. Переключение между кодировками производится управляющими кодами 016 и 017.

В режиме КОИ7 Н2 терминал отображает только большие латинские и русские буквы, при этом русские буквы занимают место малых латинских стандартной таблицы ASCII. Такой режим использовали терминалы, эксплуатировавшиеся совместно с СМ ЭВМ, например СМ7209 или ВТА-2000-3. Переключение в этот режим и обратно в режим Н0/Н1 производится клавишей F11.

В режиме Н2 эксплуатация оригинальных ОС от DEC невозможна, ибо все сообщения выводятся в искаженном виде (знаменитое ИНЖАЛИД ДЕЖИЦЕ). Но в этом режиме корректно будут работать адаптированные в СССР системы (РАФОС, например), а также многие написанные в те времена программы, использовавшие русский язык, к примеру, экранный редактор USED.

Неотображаемые символы из диапазона 00-37, не распознанные терминалом как управляющие команды, отображаются на экране в виде символов латинского алфавита с кодом 100-137, мигающих с частотой около 3 Гц. Эта особенность отображения имеется не только в оригинальном модуле КСМ, но и во многих терминалах, например 15ИЭ-00-013 и СМ7209. В оригинальном DEC VT52 этой особенности нет, и все нераспознанные управляющие коды игнорируются.

#### 13.1.4. Формат экрана

Дисплейный блок КСМ работает в стандартном для всех таких терминалов разрешении 24 пользовательских строки по 80 символов. Самая верхняя строка экрана терминала — служебная строка. В эту строку выводятся буквенные индикаторы режимов работы красного цвета:

<b>LINE</b>	режим связи с ЭВМ.
<b>LOCAL</b>	автономный режим (локальная петля)
<b>CAPS</b>	включен CAPS LOCK.
<b>ALT</b>	включен альтернативный режим дополнительной клавиатуры.
<b>KOI7</b>	включена кодировка KOI7 H2
<b>LAT</b>	включена кодировка H0
<b>RUS</b>	включена кодировка H1
<b>HOLD</b>	включен режим HOLD SCREEN
<b>WAIT</b>	терминал ждет команды Scroll для продолжения вывода
<b>MUTE</b>	звуковой сигнал отключен.

У правого края служебной строки отображается текущая скорость последовательного интерфейса, а также время, прошедшее с момента включения дисплея.

Строка состояния данного контроллера отличается внешним видом от соответствующей строки оригинального КСМ. В оригинале ключевые слова выводились русскими буквами, причем часть аббревиатур была малопонятна (типа ЗВЫВ или ДКЛ). Некоторые параметры отображались в виде еще более малопонятных битовых полей. Я считаю, что, например, HOLD выглядит более понятно, чем ЗВЫВ. Однако, если кому-то хочется поностальгировать именно со старым внешним видом строки, то исправить надписи в тексте управляющей программы совсем несложно.

Со строки 2 начинается собственно экран терминала. Пользовательские данные выводятся зеленым цветом. Выглядит все это так:

```
LINE      CAPS      LAT      HOLD      MUTE      38400      00:00:43

000000
@B
&RK■
DW Handler V6.5, (c) D.S.C., 1992-2016

?MY-I-MY/Dz-emulator handler. Un_Soft 1991. V3.04

RT-11FB (S) V05.04 D

.SEI IT QUIET
?21
.SEI IT HOLD

.-
```

Модуль поддерживает 2 типа курсора — подчеркивание и блок размером со знакоместо. Форма курсора переключается клавишей F8. В случае блочного курсора символ, под которым он стоит, отображается инвертированным. Также курсор может быть мигающим или немигающим, этот режим переключается клавишей F7. Курсор отображается желтым цветом.

### 13.1.5. Клавиатура.

Алфавитно-цифровое поле клавиатуры используется для ввода букв, цифр и символов. Раскладка букв и символов во всех режимах примерно соответствует стандартной раскладке АТ-клавиатуры (qwerty/йцукен) с небольшими отличиями, которые легко выявить опытным путем (лень рисовать картинку с раскладкой).

Поле цифровой дополнительной клавиатуры используется в точности так же, как на клавиатуре MC7004 — в стандартном режиме вводятся цифры, в альтернативном — управляющие коды. Клавиши - и + соответствуют стрелкам вверх-вниз. Клавиши Numlock, /, \* соответствуют кодам PF1, PF2 и PF3 стандартной клавиатуры VT52 (и расположены в тех же местах). Отсутствующие клавиши влево-вправо перенесены на поле управления курсором, там же продублированы стрелки вверх-вниз.

Управляющие клавиши АТ-клавиатуры имеют следующие значения:

F5	Уменьшение на одну ступень скорости последовательного интерфейса
F6	Увеличение на одну ступень скорости последовательного интерфейса
F7	Переключение формы курсора - блок/подчеркивание



F8	Включение-отключение мигания курсора
F9	Отключение-включение звукового сигнала.
F10	Переключение режимов Line (связь с ЭВМ)/Local (автономный режим)
F11	Включение/отключение кодировки КОИ7 Н2 (все большие буквы)
L-ALT	Переключение кодировки RUS (H1) / LAT (H0)
R-ALT	Ввод кода LF (ПС) — переход на новую строку
Scrollock	Вывод следующей строки в режиме HoldScreen (с shift - вывод страницы)
PgDn	Раздвижка экрана вниз
PgUp	Раздвижка экрана вверх
Home	Установка курсора в начало экрана
Ins	Раздвижка строки вправо

### 13.1.6. Шрифт.

Оригинальный контроллер КСМ отображает знакоместа формата 8\*8. В данной разработке знакоместа имеют размер 8\*12, что позволяет использовать более четкий и разборчивый шрифт.

В схему входит ROM размером 4К (созданный в статической памяти FPGA) с образом шрифта знакогенератора. Исходные образы шрифта лежат в ksm-firmware/font/font-\*.bin. Раскладка кодов символов внутри файла шрифта:

00-1f (000-037) большие латинские буквы (для индикации управляющих кодов)  
20-3f (040-077) цифры и обычные символы  
40-5f (100-137) большие латинские буквы  
60-7f (140-177) малые латинские буквы  
80-9f (200-237) псевдографика  
a0-bf (240-277) не используется  
c0-df (300-337) малые русские буквы  
e0-ff (340-377) большие русские буквы

К проекту приложены 2 шрифтовых файла.

**font-main.bin** — основной шрифт 8\*12, в основном выданный из какого-то древнего досовского русификатора. Большинство линий символов имеет толщину в 2 пикселя, и шрифт смотрится вполне достойно на современных TFT-мониторах.

**font-ksm.bin** — шрифт 8\*8, выданный из знакогенератора платы КСМ (терминал 15ИЭ использует такой же шрифт). При использовании этого шрифта экран выглядит в точности таким же, как экран КСМ. Этот шрифт я приложил для ностальгирующих по старым временам, а также для наглядного сравнения корявых древних шрифтов с современными.

Вот пример изображения, формируемого шрифтом КСМ:

```
LINE      CAPS      LAT      38400  00:02:46
CHESS0.SAV  23  30-May-88    REVERS.SAV  12  17-May-85
 54 Files, 1207 Blocks
 383 Free blocks

.B00 RK2:

?040C ? B03.00
?KMON-F-Invalid command

.R FTMON

***?040C-TMOC*** МОНИТОР FTMON APR-87
ЗАПУСК: R FTMON  ПОВ.ПУСК: 30124  ВЫХОД: CУ/С

КОМАНДЫ МОНИТОРА
L ИМЯ<ВК> ЗАГРУЖАЕТ ПРОГРАММУ
R ИМЯ<ВК> ЗАГРУЖАЕТ И ЗАПУСКАЕТ ПРОГРАММУ
O ИМЯ<ВК> ЗАГРУЖАЕТ И ВЫПОЛНЯЕТ ПРОГРАММУ ОДИН РАЗ
C ИМЯ<ВК> ВЫПОЛНЯЕТ ЦЕПОЧНЫЙ ФАЙЛ
D<ВК> ВЫВОДИТ СПРАВОЧНЫЙ ФАЙЛ НА ТЕРМИНАЛ
H<ВК> ВЫВОДИТ ЭТО СООБЩЕНИЕ

,
```

Используемый шрифт подключается к мегафункции fontrom. Выбор шрифта производится в файле конфигурации config.v.

Также в каталоге ksm-firmware/font лежат некоторые утилиты для обработки шрифтовых файлов:

**font2mif** — преобразователь файл шрифта (bin) в формат mif для загрузки в FPGA.

**fontlist** — выводит на экран изображения всех шрифтов указанного bin-файла

**fontextract** — извлекает шрифт указанного или всех символов из bin-файла

**fontreplace** — заменяет указанный символ в bin-файле.

Извлеченный из двоичного файла образ символа представляет собой текстовый прямоугольник размеров 8\*12 знаков, в котором пикселя, формирующие изображение, обозначаются символом O, остальные — точкой:

```
..000...
.00.00..
00...00.
00...00.
00...00.
0000000.
00...00.
00...00.
00...00.
.....
```

.....

Такая форма удобна для редактирования изображения символа. 2 нижние строки лучше не использовать для размещения изображения, поскольку на этих строках размещается курсор в режиме подчеркивания.

## 13.2. Контроллер КГД

КГД (Контроллер Графического Дисплея) — это монохромный графический контроллер, выводящий на экран изображение размером 400\*286. Контроллер подключается прямо к шине МПИ и представлен на ней четырьмя регистрами. Такой подход является более современным по сравнению с подключением через последовательный порт.

КГД не является полноценным видеоконтроллером — он не умеет формировать полный набор видеосигналов, и работает только в паре с модулем КСМ. Поскольку КСМ формирует изображение 800\*600, то каждый графический пиксель занимает 2 точки по горизонтали и вертикали. При этом в верхней части экрана остается темное поле высотой в 28 точек. Видеоданные, формируемые текстовым и графическим контроллером, могут выводиться на экран одновременно, накладываясь друг на друга. Также, используя регистры КГД, можно запретить вывод текстовой или графической информации.

Контроллер содержит в себе двухпортовую видеопамять размером 16К, которая доступна через регистры адреса и данных. Видеопамять формируется как мегафункция 2-port altsyncram из внутренней памяти FPGA. Это накладывает серьезные ограничения на тип применяемой FPGA — младшая из серии Cyclone 4 микросхема EP4CE6 не содержит достаточного количества памяти.

## 13.3. Аппаратные особенности терминальной подсистемы

Модуль КСМ умеет управлять звуковым сигналом по коду 007 — этой возможности не было в оригинальном контроллере. Сигнал передается через порт buzzer и принимает значение 1, когда надо включить звук. Его можно прямо подавать на пьезопищалку со встроенным генератором звука. Либо создать таймер/делитель на частоту около 400 Гц и использовать его для генерации звука. Или вообще обойтись без звука, как в настоящем КСМ.

На экране изображение имеет следующие цвета:

- Строка состояния — **красный цвет**
- Часы — **фиолетовый цвет**
- Данные пользователя — **зеленый цвет**
- Графика КГД — **белый цвет**
- Курсор — **желтый цвет**.

Если вы хотите получить полностью монохромное изображение, как на древних мониторах, то сигналы vga<sub>red</sub>, vga<sub>green</sub>, vga<sub>blue</sub>, выходящие из модуля КСМ, а также сигнал vga<sub>video</sub>, выходящий из модуля КГД, надо объединить по схеме ИЛИ и подать на все 3 видео-ЦАП, настроив эти ЦАП на формирование нужного оттенка изображения. Так можно получить черно-зеленое, черно-желтое, черно-белое и любое другое монохромное изображение.

Линии клавиатуры PS/2 Clock и Data должны быть подтянуты к питанию 3.3v. Для этого можно включить внутренние подтягивающие резисторы в FPGA (опция weak pull-up resistor в assignments editor), или подпаять внешние резисторы 2-5к между каждой линией и 3.3v, если их нет на вашей плате.

Некоторые мониторы и устройства видеозахвата некорректно отображают картинку — она частично уходит за левый край экрана. В этом случае следует произвести коррекцию фазы строчного синхроимпульса. В файле `config.v` надо раскомментировать строчку

```
`define hsync_shift 11'd27
```

и указать, на сколько пикселей следует сдвинуть картинку (в этом примере — 27).