

# FPGA-реплика платы MC1201.02 и ДВК-3

## Введение

Данная разработка является попыткой сделать современный вариант платы MC1201.02 на основе FPGA, а также одноплатную ЭВМ, аналогичную ДВК-3, которая и была построена на основе этой платы.

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.

## Общее устройство платы

Плата выполнена на основе wishbone-варианта процессора 1801BM2. Все внутренние периферийные устройства также работают с этой шиной. Сейчас на плате имеются следующие устройства:

- Последовательный порт ИРПС консольного терминала (ТТ:)
- Дополнительный последовательный порт (XL:, LS:), может использоваться для подключения последовательного принтера или обмена данными с другой ЭВМ.
- 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)
- 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).
- Динамическая память DRAM размером 64К
- ROM с монитором 055 или 279
- Терминал VT52 с VGA-выходом, подключенный к консольному ИРПС. Скорости терминала и ИРПС аппаратно согласованы. Можно считать этот терминал аналогом платы КСМ настоящей ДВК.

Любой ненужный модуль можно легко выкинуть из конфигурации для экономии ресурсов FPGA. 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.

В предлагаемом составе плата является одноплатной версией машины ДВК-3 с подключенным к ней модулем КСМ или фоязинским дисплеем. Вот пример экрана дисплея с загруженной RT-11:

```
VT52  LINE      CAPS      L24    LAT      38400  00:02:03

000000
@B
8RK
DH Handler V6.5, (c) D.S.C., 1992-2016

RT-11SJ (S) V05.04 D

.SET TT QUIET
?21
.DIR DX:
03-Jan-99
RT11SJ .SYS      84   03-Jan-99      SHAP  .SYS      27   17-Nov-87
TT      .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
DATEIME.SAV       3   01-Aug-90      STARTS.COM  1   18-Nov-99
SL      .SYS      14   20-Dec-85      K52   .SAV      54   17-Nov-87
14 Files, 325 Blocks
161 Free blocks

--
```

## Используемые FPGA

Разработка сделана на основе FPGA серии Cyclone 4 фирмы Altera (увы, теперь это уже intel). В вышеуказанном составе схема занимает чуть больше 7000 ячеек и влезает в микросхему EP4CE10. Если выкинуть часть дисковых контроллеров, 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.

## Структура проекта

В дереве проекта имеются следующие каталоги:

doc/ - документация проекта

disk/ - утилиты для чтения-записи образов дисков и тестовый дисковый банк

hdl/ - verilog-файлы основного проекта

hdl/vt52/ - verilog-файлы терминала vt52

hdl/sdram\_ip - контроллер DRAM

rom/ - файлы образов ПЗУ

vt52/ - файлы, относящиеся к микропрограмме терминала и экранные шрифты.

Главным модулем проекта является модуль mc1201\_02, лежащий в файле hdl/mc1201-02.v. К его внешним портам подключаются:

- Тактовая частота 50 МГц

- 4 кнопки
- 4 переключателя
- 4 индикаторных светодиода
- Микросхема динамической памяти DRAM
- SD-карта
- VGA монитор
- 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.

## Внешние органы управления и индикации

К внешним портам схемы подключаются 4 кнопки, 3 переключателя и 4 светодиода.

Назначение кнопок:

- `button[0]` – кнопка общего сброса. Инициализирует все подсистемы, кроме терминального модуля.
- `button[1]` – 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 оригинальной ДВК. Тогда появится возможность пошагового прохождения программ.
- `button[2]` – кнопка сброса терминального модуля, помогает при его зависании или просто для очистки экрана.
- `button[3]` – включение-выключение таймерного прерывания. Каждое нажатие кнопки переключает режим таймера (вкл-выкл). По умолчанию при включении питания таймер включен. Общий сброс не изменяет состояние выключателя таймера. При желании вместо кнопки можно установить двухпозиционный переключатель “таймер”, как в настоящей ДВК. Переключатель следует подключить к линии `timer_on`, а логику обработки данной кнопки вообще выкинуть из схемы.

Назначение переключателей:

- `sw[1:0]` – эти два переключателя выбирают дисковый банк на SD-карте (описан далее в разделе “дисковая подсистема”).
- `sw[2]` – 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.

Назначение индикаторов:

- led[0] – обращение к диску RK
- led[1] – обращение к диску DW
- led[2] – обращение к диску DX
- led[3] – индикация состояния таймера (светится – таймер вкл).

В данном случае подразумевается, что:

Кнопка в нажатом состоянии выдает 0, в отжатом 1

Переключатель в отключенном положении дает 0, во включенном – 1

Светодиод светится при установке его порта в 0.

## 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.

Все три дисковых контроллера разделяют одну и ту же карту. Набор из 8 дисков RK, одного диска DW и двух дисков DX 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. Можно сделать переключаемые банки для каждого типа дисков отдельно, можно вынести любой тип диска на отдельную SD-карту. Все определяется вашими потребностями.

Карта распределения памяти под образы дисков описана в файле card-layout.ods. Для записи образов дисков на SD-карту используется программа sd-store, а для извлечения образов с карты – программа sd-extract. Исходные тексты этих программ лежат в каталоге disk/.

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, и ФОДОС-ТМОС на диске 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`.

## SDSPI

Для работы с картой памяти используется модуль `SDSPI` – контроллер SD-карты. За его основу взят одноименный модуль из проекта `rdr2011`, 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.

Имеется два варианта модуля – `spspi-master` и `sdspi-slave`. Вариант `slave` не содержит в себе кода инициализации карты и используется для всех дисковых контроллеров кроме `RK` – эти контроллеры получают доступ к уже инициализированной карте. Вариант `master` – это полноценный контроллер, включает в себя код инициализации карты и используется только в контроллере `RK` – этот контроллер получает доступ к карте по умолчанию, сразу после сброса.

Любой из дисков `DX` или `DW` также можно вынести на отдельную SD-карту. Для этого в контроллере надо заменить модуль `sdspi_slave` на `sdspi`, и подключить его к физическим портам отдельной SD-карты. Линии `sdreq` (запрос доступа к карте) и `sdack` (подтверждение доступа) в этом случае надо просто замкнуть друг на друга.

## Контроллер RK11/RK05

Этот модуль реализует контроллер `RK11` (`RK`: в `rt-11`), с подключенными к нему 8 дисками `RK05` объемом 3 Мб. Данный дисковый контроллер является самым быстрым из всех – он единственный 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`).

Контроллер `RK05` является первичным дисковым контроллером – именно его модуль `sdspi` производит инициализацию SD-карты. Если вы хотите убрать этот контроллер из конфигурации, то первичным надо назначить какой-либо из оставшихся – `DX` или `DW`, заменив в нем модуль `sdspi_slave` на `sdspi`.

## 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.

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 сд-карты. Как и в случае RK11, логический номер блока виртуального HDD соответствует абсолютному номер 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 ДБК, 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.

## Контроллер RX11/RX01.

Модуль реализует контроллер RX11 с подключенными к нему двумя дисководы RX01. Нашим аналогом такого дисковода является ГМД-7012. 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 образами 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 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.

## 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.

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.

## Контроллер DRAM

Это, пожалуй, самое узкое место данного проекта. Контроллер этот я взял из комплекта демонстрационных модулей, идущих в комплекте с моей китайской FPGA-платой.

Контроллер этот далеко не оптимален – он тратит по 11 тактов на одну операцию чтения (от фронта stb до фронта ask), что, на мой взгляд, многовато. Было бы неплохо заменить его на что-нибудь более приличное, но практически все контроллеры с orpcores рассчитаны на 32-битную шину данных, а в других местах я и не искал. Но, видимо, рано или поздно придется это сделать, ибо данный контроллер существенно снижает производительность платы.

Физически к контроллеру можно подключить практически любую 16-битную микросхему SDRAM. Основной временной параметр — Cas Latency, установлен равным 2. Если DRAM поддерживает только CL3, то следует исправить значение cas\_latency в файле sdram\_para.v. Но тогда на один цикл обращения к памяти будет уходить 12 тактов.

Если в вашей FPGA есть лишних 512 Мбит внутренней памяти, то контроллер DRAM лучше вообще выкинуть, а вместо него ввести в адресное пространство 64К статической памяти. Производительность вырастет в разы.

## Системное ПЗУ

На платах MC1201.02 штатно устанавливалось ПЗУ 1801pe2 с прошивкой 055, содержащее в себе теневого монитора и набор загрузчиков. Существует также доработанная версия этого ПЗУ — 279, поддерживающая расширенный набор отладочных команд и несколько новых загрузчиков. В данной схеме можно использовать любое из этих ПЗУ, их образы в исходном виде и в формате MIF лежат в каталоге rom/. Для смены ПЗУ нужно отредактировать имя файла инициализации памяти ip-компонента rom055.qip.

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 2 — в этом и заключается ошибка. В результате код загрузчика, начиная с адреса 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 rom лежат уже исправленные версии.

## Аппаратный терминал

Этот терминал я выкладывал в виде отдельного проекта на гитхабе — <https://github.com/forth32/vt52/>

Терминал полностью совместим с протоколом DEC VT52, а также фрязинским терминалом 15ИЭ-00-013. Поддерживаются все кодировки букв, использовавшихся на ДВК и СМ ЭВМ — КОИ7 Н0, Н1, Н2. Для вывода информации используется VGA-монитор, для ввода — PS/2 клавиатура. Подробное описание работы терминала и раскладки клавиатуры можно прочитать в документации, лежащей в вышеуказанном репозитории в каталоге doc/.

Терминал содержит в своем составе собственный микропроцессор 1801BM2 и собственную статическую память, адресные пространства терминала и основной системы никак не связаны. Обмен данными между терминалом и консольным портом системы происходит через пару связанных друг с другом модулей ИРПС. Скорость последовательного порта самого терминала и сопряженного с ним интерфейса ИРПС аппаратно согласованы — при изменении скорости терминала кнопками F5/F6 также меняется и скорость ИРПС. Таким образом, скорость вывода данных на экран можно динамически менять, не нарушая работу системы.

Терминал может быть подключен к любому из портов ИРПС1 или ИРПС2 в зависимости от положения переключателя sw[2]. Переключение можно делать прямо на ходу без сброса системы. Если терминал вам вообще не нужен — его можно спокойно убрать из схемы, а в качестве консоли использовать, например, компьютер с запущенной на нем терминальной программой, подключив его последовательный интерфейс (или переходник usb-serial) к портам ИРПС1 uart1\_txd и uart1\_rxd.

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.

В зависимости от подключения разъема VGA может потребоваться переделка схемы управления видео-цап, лежащей в головном модуле mc1201-02.v. В данном проекте использовалась плата, на которой установлен 6-битный ЦАП зеленого цвета и 5-битные ЦАП синего и красного цветов, поэтому схема выглядит так:

```
// выбор яркости каждого цвета
assign vgag = (vgagreen == 1'b1) ? 6'b101111 : 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 или ЦАП вообще отсутствует, этот фрагмент необходимо поправить под ваши нужды.