

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)
- 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
- ROM с монитором 055 или 279

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 полном составе плата является одноплатной версии машины ДВК-3 с подключенными к ней модулями КСМ и КГД. Вот пример экрана дисплея с загруженной RT-11:

```
LINE      CAPS      LAT      38400  00:00:28
@B
xRK■
DH 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
DH      .SYS     15   02-Jan-99      K52     .SAV     54   17-Nov-87

 16 Files, 343 Blocks
 143 Free blocks

.-
```

В данной схеме процессор работает на частоте 100 МГц. Имеется возможность снизить эту частоту до 4.5 МГц с помощью переключателя sw[3] для работы с игрушками и другими интерактивными программами.

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. PLL должна выдавать 3 тактовых сигнала:

1. 100 МГц прямая фаза (0°) - основной тактовый сигнал
2. 100 МГц инверсная фаза (180°) - дополнительный тактовый сигнал процессорного модуля.
3. 12.5 МГц прямая фаза (0°) - тактовый сигнал SD-контроллера и карты памяти.

Собственный генератор платы должен обеспечивать частоту 50 МГц, которая используется PLL для синтеза вышеуказанных трех сигналов, а также напрямую терминальным модулем

для формирования видеосигнала VGA. Если ваша плата имеет генератор другой частоты, то придется синтезировать 50 МГц с помощью PLL.

В проекте используются следующие модули статической памяти altsyncram:

- rom055 — ROM размером 4096*16 бит, предзагруженный ПЗУ монитора 055 (rom/055.mif) или 279 (rom/279.mif).
- fontrom — ROM размером 32768*1 бит, предзагруженный файлом экранного шрифта vt52/font/font.mif.
- vtmem — RAM размером 2048*16, предзагруженный терминальной микропрограммой vt52/terminal.mif
- kgdram — двухпортовая RAM размером 16384*8, являющаяся видеопамятью графического контроллера КГД. Второй порт этого модуля имеет 1-битную шину данных.

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 альтеровским мегафункциям, помещены в отдельный каталог ip-components/.

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/ksm/ - verilog-файлы модуля KCM
- hdl/sdram_ip - контроллер SDRAM
- ip-components — файлы описания мегафункций (ip компоненты)
- rom/ - файлы образов ПЗУ
- ksm-firmware/ - файлы, относящиеся к микропрограмме KCM и экранные шрифты.
- screenshots/ - примеры текстовых и графических экранов

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 SDRAM
- 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.

Настройка конфигурации проекта

В корне проекта лежит файл `config.v`, с помощью которого настраивается состав проекта. Для каждого входящего в проект модуля в файле имеется отдельная строка:

```
`define KSM_module      // текстовый контроллер КСМ
`define KGD_module      // графический контроллер КГД
`define IRPS2_module    // второй последовательный порт ИРПС
`define IRPR_module     // параллельный порт ИРПР
`define RK_module       // диск RK-11/RK05
`define DW_module       // жесткий диск DW
`define DX_module       // гибкий диск RX01
`define MY_module       // гибкий диск двойной плотности MY
```

Любую из этих строк можно закомментировать, и тогда соответствующий модуль исключается из проекта — это позволит сэкономить ресурсы FPGA. Обратите внимание, что при исключении модуля КСМ автоматически исключается и модуль КГД, поскольку графический контроллер может работать только в паре с текстовым. Кроме того, при исключении модуля КСМ консольный последовательный интерфейс автоматически выводится на внешние порты `irps_txd/irps_rxd`.

Кроме исключения модулей, с помощью файла конфигурации можно настроить начальную скорость терминала и второго последовательного порта.

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.
- sw[3] — включение режима замедления процессора. При выключенном переключателе процессор работает на полной тактовой частоте 100 МГц. Если переключатель включен, то тактовая частота снижается до 4.5 МГц, что примерно соответствует тактовой частоте платы MC1201.02. Этот режим необходим для игрушек и других слишком быстро работающих интерактивных программ.

Назначение индикаторов:

- led[0] – обращение к диску RK
- led[1] – обращение к диску DW
- led[2] – обращение к диску DX или MY
- 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 выдает 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.

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.

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 ДБК.

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 подключить к соответствующим выводам отдельной sd-карты. Тактовый сигнал CLK для карты можно взять из линии sdclock. Линии sdreq (запрос доступа к карте) и sdack (подтверждение доступа) в этом случае надо просто замкнуть друг на друга.

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:)
- gx01.v — контроллер 8-дюймового дисковода RX01 (DX:)
- fdd-my.v — контроллер 5-дюймового дисковода (MY:)
- dw.v — контроллер жесткого диска RD50C (DW:)

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

Сводная таблица дисковых контроллеров

	RK	DX	MY	DW
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.

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 МҮ (в данной реализации, конечно), а емкость — гораздо больше.

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 сд-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.

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 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 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.

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.

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 `mc1201_02.v`.

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 `mc1201_02.v`.

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.

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 буферизирующую 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 SDRAM

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 opencores 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 SDRAM 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 SDRAM 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 gom/. Для смены ПЗУ нужно отредактировать имя файла инициализации памяти ip-компонента gom055.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 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 gom лежат уже исправленные версии.

Дисплейные модули КСМ и КГД

Эти модули подробно рассмотрены в отдельном документе ksm-kgd.odt.