



РАЗРАБОТКА ЭМУЛЯТОРА ЖЕСТКОГО ДИСКА НА БАЗЕ КАРТЫ ПАМЯТИ ТИПА SD

Данная работа показывает способ замены устаревающего оборудования при помощи ПЛИС на примере жестких дисков. Рассматривается несколько видов тестирования ПО, облегчающего процесс разработки. Итоговая замена соответствует стандартам и может быть усовершенствована в дальнейшем.

ПЛИС, жесткий диск, устаревание оборудования, эмуляция.

Широкий спектр оборудования, включая компьютеры для промышленной автоматизации и станки с ЧПУ, выпущенные до начала 2000-х, использовали в качестве носителя 3,5-дюймовые дискеты или жесткие диски с интерфейсом PATA (Parallel Advanced Technology Attachment), известным также под названием IDE (Integrated Drive Electronics). Производители выпускали диски с подключением по интерфейсу PATA даже после стандартизации в 2000 г. последовательного порта Serial Advanced Technology Attachment (SATA). Последний полностью вытеснил PATA с рынка в 2008 г., поэтому возраст практически любого диска с этим интерфейсом составляет как минимум 10 лет. По этим причинам при эксплуатации оборудования стала актуальной замена устаревших дисков на аппаратные эмуляторы, использующие носители данных на флеш-памяти. Такие эмуляторы, обеспечивающие простое подключение современных USB-носителей, NAND-памяти и различных карт памяти, ориентированы, как правило, на промышленное применение. Они обеспечивают повышенную надежность эксплуатации и хранения информации, практически не подвержены воздействию магнитных полей, пыли, вибрации, температуры и других вредных факторов. Аппаратные эмуляторы поддерживают одновременное хранение на одном накопителе множество образов дисков, что устраняет необходимость в хранении и использовании большого количества различных дисков [2].

Контроллер диска (рис. 1) состоит из автомата для работы с шиной, автомата для обработки команд, таймера и «поставщика» данных. Последний компонент может быть представлен любым носителем данных или связкой носителя данных и кэширующего устройства.

Для написания программы контроллера диска выбран язык описания аппаратуры Verilog. Он независим от какой-либо платформы и ее типа (CPLD/FPGA), что позволяет использовать его практически на любой платформе, а также воспроизводить результаты работы в симуляции/эмуляции.

Создание программной модели и ее тестирование выполнялось на первом этапе проектирования. Модель проекта на C++, необходимая для тестирования, создавалась с помощью симулятора Verilator [3], выбранного из-за простоты конфигурации и подключения к существующему коду. Симулятор преобразует

код на Verilog в код на C++, который на выходе упаковывается в обычную библиотеку.

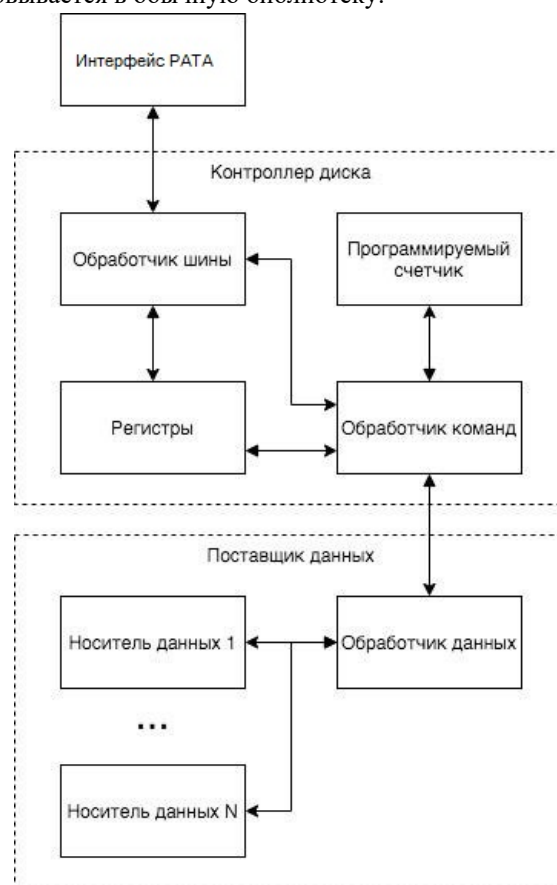


Рис. 1. Функциональная схема эмулятора

Проверка программы осуществлялась при помощи интерпретатора Icarus Verilog [1]. Он позволил просмотреть все внутренние регистры будущего устройства, а также временные диаграммы работы (рис. 2). На них можно убедиться, что алгоритмы обрабатывают сигналы в нужные моменты времени и предоставляют корректные значения.

В связке с симулятором использован эмулятор компьютера PCset, имеющий простой интерфейс для получения и отправки данных через порты ввода/вывода, большой выбор BIOS разных производителей. Это позволило проверить модель с различными реализациями управления жесткими дисками.

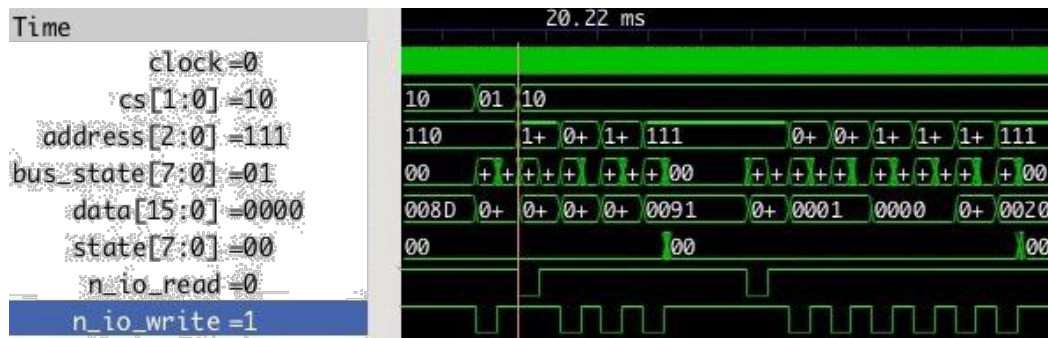


Рис. 2. Пример временной диаграммы

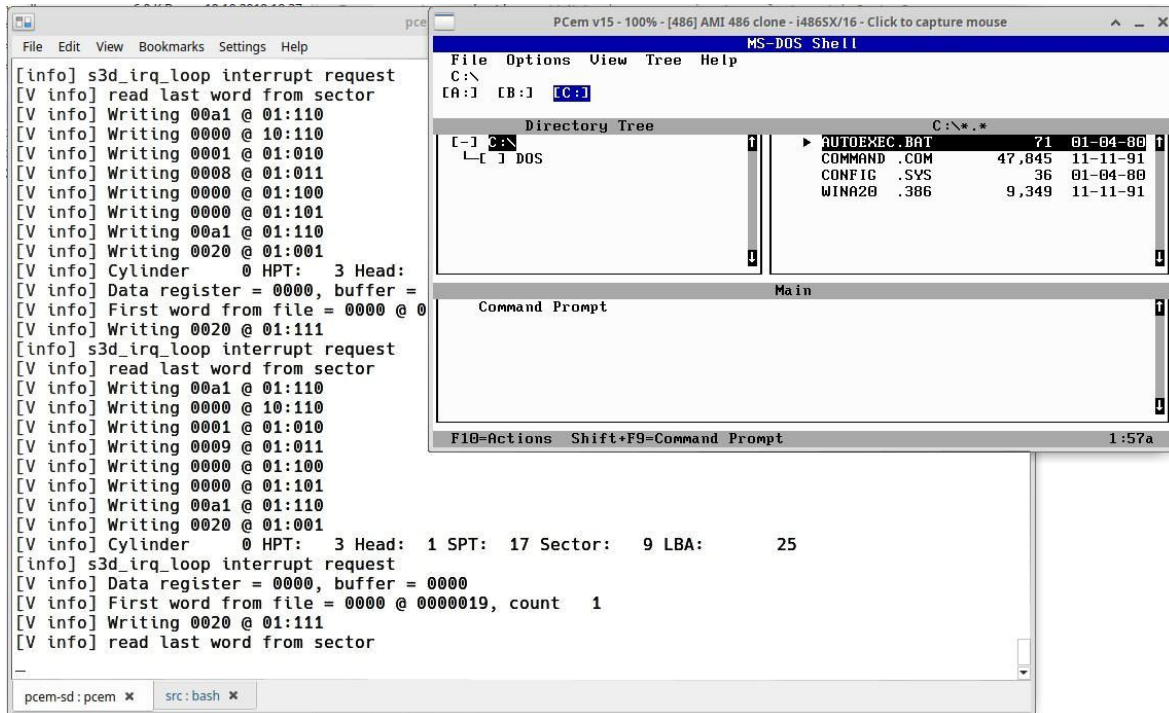


Рис. 3. Пример работы эмулятора и модели

Для проверки программной модели эмулятора использовано несколько типов тестирования:

- автоматическое;
- проверка шины, внутренних регистров, команд [2];
- проверка тактов на временной диаграмме;
- ручное на эмуляторе.

Проверялись команды из обязательного набора стандарта ATA-2, а также несколько команд, которые были добавлены в процессе тестирования:

- EXECUTE DEVICE DIAGNOSTIC (запуск диагностики диска);
- IDENTIFY DEVICE (вывести информацию о диске);
- INITIALIZE DEVICE PARAMETERS (записать параметры для работы с диском);
- SEEK (переместить позицию чтения/записи);
- READ SECTOR(S) (w/retry) (чтение данных с повтором при ошибке);
- READ SECTOR(S) (w/o retry) (чтение данных);
- READ VERIFY SECTOR(S) (w/retry) (проверка доступности данных с повтором при ошибке);

- READ VERIFY SECTOR(S) (w/o retry) (проверка доступности данных);
- WRITE SECTOR(S) (w/retry) (запись данных с повтором при ошибке);
- WRITE SECTOR(S) (w/o retry) (запись данных);
- RECALIBRATE (калибровка механизма).

При реализации команд в модели, компьютер запрашивает состояние диска через регистр статуса. Если диск не занят обработкой предыдущего запроса, компьютер посылает одну команду или их серию, чтобы задать различные параметры и запустить выполнение. После этого компьютер еще раз запрашивает статус, чтобы убедиться в том, что диск занят обработкой команды. Если команда имеет обратную связь через аппаратное прерывание, то компьютер будет ожидать его, иначе будет опрашивать регистр статуса через промежутки времени.

Тестирование показало, что модель способна проходить процесс инициализации и загружать простую операционную систему (рис. 3).

Однако, несмотря на компиляцию кода Verilog в C, текущая версия эмулятора персонального компьютера не позволяет корректно работать с отдельным

потоком диска и вовремя пересылать ему данные. Из-за этого основной поток часто ожидает выполнения кода со стороны диска и приостанавливает работу процессора.

С учетом результатов тестирования и проверки программой модели эмулятора диска для создания аппаратного прототипа выбрана отладочная плата **iCE40HX8K-EVB** от компании Lattice Semiconductor. Она имеет достаточно большое количество логических ячеек (прототип занимает 796 ячеек из 7680), работает на частоте до 100 МГц, относительно недорогая [4]. У данной платформы также имеются полностью бесплатные и открытые инструменты для создания прошивок.

Основные компоненты контроллера:

- FPGA iCE40HX-8K;
- конвертеры логического уровня;
- преобразователи питания;
- разъемы.

К FPGA, помимо нескольких напряжений (3.3 В для портов ввода/вывода, 1.2 В для основного питания) и тактового генератора на 100 МГц, подключена конфигурационная флеш-память.

FPGA и большая часть CPLD работают с логическими сигналами от 1.8 до 3.3 В, что не дает подключить их напрямую к интерфейсу IDE компьютера. Поэтому в схеме предусмотрены четыре двунаправленных конвертера логических уровней. Они автоматически переключают сторону входа/выхода в зависимости от напряжения на контактах той или иной стороны. Стандартный разъем питания жесткого диска предоставляет +5 В. Это напряжение +5 В преобразуется в +3.3 В и +1.2 В для FPGA и карты памяти.

На плате прототипа установлен разъем интерфейса PATA, два слота под карты памяти (для поддержки резервирования), разъем для программирования FPGA. Имеются два светодиода активности и группа контактов для установки режима устройства на шине – Master/Slave или выбор режима кабелем. Размер печатной платы совпадает с размером 3.5-дюймового жесткого диска для удобства установки в конструктив.

Для прототипа не понадобилось менять какие-либо критические части программы. Была добавлена инициализация регистров и состояний конечных автоматов при включении. После этого никаких изменений не требовалось, что является подтверждением

того, что тестирование на эмуляторе и автоматическое тестирование были корректными.

Прототип подтвердил корректную работу с шиной, передачу данных первого блока на 512 байт, передачу информации об устройстве. Тестирование прототипа выявило, что:

- для конечного устройства, необходимо увеличить частоту работы, т.к. период такта в 20 нс при частоте 50 МГц позволяет работать только в самом медленном режиме PIO 0 (600 нс на один цикл передачи);

- необходимо увеличить объем CPLD/FPGA, для того чтобы все блоки помещались в одну микросхему;

В результате выполнения работы разработан контроллер диска и программное обеспечение для него, а также система тестирования, позволяющая проверить его функционал.

Устройство имеет несколько отличий от большинства представленных на рынке:

- открытый исходный код;
- возможность добавления собственного функционала;
- возможность резервирования данных.

Проект показывает, что при помощи ПЛИС (FPGA) при сравнительно небольших затратах, можно восстановить поврежденные или даже неработоспособные устройства без потерь их совместимости с действующим оборудованием.

Литература

1. ASIC World: Verilog Basics – URL: <http://www.asicworld.com/systemverilog/basic.html> (дата обращения: 11.03.2020). – Текст : электронный.
2. CNews Analytics: Using FPGAs to avoid microprocessor obsolescence – URL: <https://www.edn.com/design/programmable-logic/4015159/Using-FPGAs-to-avoid-microprocessor-obsolescence> (дата обращения: 05.11.2019). – Текст : электронный.
3. Gisselquist Technology: Verilog, Formal Verification and Verilator Beginner's Tutorial – URL: <http://zipcpu.com/tutorial/> (дата обращения: 12.03.2020). – Текст : электронный.
4. Lattice Semiconductor: iCE40 LP/HX/LM – URL: <http://www.latticesemi.com/en/Products/FPGAandCPLD/iCE40> (дата обращения: 24.03.2020). – Текст : электронный.

А.М. Vodovozov, I.P. Korytov, D.Y. Kozin
Vologda State University

DEVELOPMENT OF HARD DISK EMULATOR BASED ON SD CARDS

This work shows a method for replacing aging equipment using FPGAs basing on the example of hard drives. Several types of software testing are considered to facilitate the development process. The final replacement is compliant with standards and may be further improved.

FPGA, hard disk drive, hardware obsolescence, emulation.