

Санкт-Петербургский государственный университет
Математико-механический факультет
Кафедра системного программирования

Курсовая работа:

**РАЗРАБОТКА МОДУЛЯ ПАМЯТИ ДЛЯ МНОГОЯДЕРНОГО ПОТОКОВОГО
ВЫЧИСЛИТЕЛЯ**

Студент: Тодорук Е.А.
Научный руководитель: Кривошеин Б.Н.
Группа: 361

Санкт-Петербург
2012

Содержание

1 Введение	2
2 Цель работы	3
3 Описание работы	4
3.1 Модуль памяти	4
3.2 Модуль контроллера памяти	5
4 Итоги работы	7
5 Используемые технологии и средства	8
Список литературы	9

1 Введение

Применение параллельных вычислительных систем является стратегическим направлением развития вычислительной техники. Это обстоятельство вызвано не только принципиальным ограничением максимально возможного быстродействия обычных процессоров, но и практически постоянным существованием вычислительных задач, для решения которых возможностей существующих средств вычислительной техники всегда оказывается недостаточно. В связи с постоянным ростом объёма данных одной из самых актуальных задач на сегодняшний день является обработка больших потоков данных.

Использование больших параллельных вычислительных систем становится невозможным, когда речь идёт о персональных компьютерах и мобильных устройствах. Для персональных компьютеров существуют графические процессоры (GPU), которые уже позволяют в некоторых случаях производить вычисления для нужд пользователей, но для общего использования они не подходят.

Осенью 2011 года стартовал студенческий проект, задачей которого стояла разработка многоядерного потокового вычислителя (МПВ), ориентированного на потоковую обработку данных для встроенных применений. Архитектура параллельного вычислителя создавалась на базе управляющего процессора и массива простых вычислительных ядер на базе FPGA. На текущий момент реализованы сеть простых вычислительных процессоров (ПВП) и транслятор микрокода для этого вычислителя.

2 Цель работы

Простой вычислительный процессор представляет из себя совокупность модулей: ввод данных, вывод данных, простой вычислитель DSP48E1, память и контроллер памяти (Рис. 1). Такие простые вычислительные процессоры объединяются в слои, а слои в вычислительную сеть — основную составляющую МПВ. Подробно о слоях и сетях ПВП можно прочитать в курсовой работе Забранского Дмитрия [1].

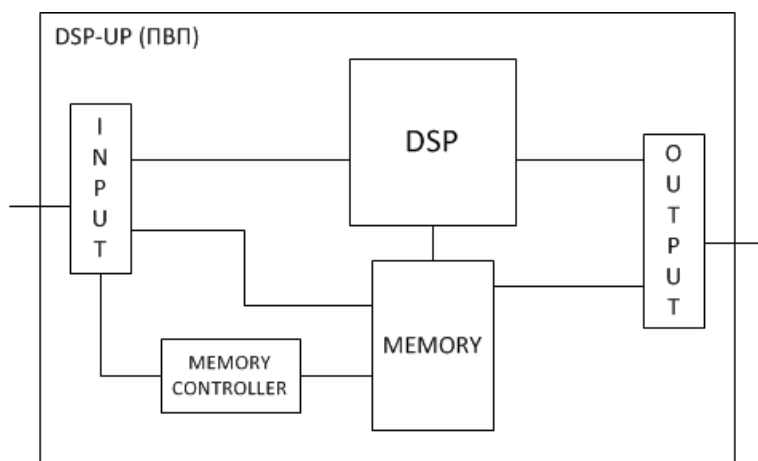


Рис. 1: Простая схема ПВП

Память в ПВП является основным управляющим элементом. Она хранит не только микрокоманды для DSP, а также управляющие сигналы для всех остальных модулей ПВП. Кроме основного требования к памяти (хранение и выдача команд) было добавлено ещё одно — расширяемость (возможность задавать глубину).

Там где есть память, есть и контроллер памяти, на который ложится ещё более ответственная задача — управление памятью.

Моя часть работы заключалась в разработке модулей памяти и контроллера памяти. Об остальных модулях ПВП можно прочитать в курсовой работе Солдатов Дмитрия. [2].

3 Описание работы

3.1 Модуль памяти

Для создания модуля памяти использовался Xilinx CORE Generator (LogiCORE) — утилита для генерирования параметризованных логических ядер для FPGA. При генерации задавались такие параметры для памяти как глубина памяти и ширина данных. На выходе генератора был получен готовый модуль на VHDL (Рис. 2).



Рис. 2: Модуль памяти ПВП

Входные порты

- **a** — Адрес, 6 бит. По этому порту памяти приходит адрес ячейки, по которому необходимо записать или прочитать данные. Так как заданная глубина памяти — 64 ячейки, то для задания адреса необходимо ровно 6 бит. Стоит отметить, что глубина и ширина памяти — это константы, которые можно менять, если потребуется больший объём памяти.
- **d** — Данные, 48 бит. В этот порт приходят данные для записи их в память.
- **clk** — Вход тактовых импульсов, 1 бит. Задаёт тактовую частоту работы модуля.
- **we** — Флаг записи, 1 бит. Порт включения/отключения записи в память. Если подаём на него 1, то память начинает записывать данные по адресу **a**, если подать 0, то память будет выдавать данные по адресу **a** на выходной порт **spo**

Выходной порт (spo)

На порт **spo** подаются данные при вычитывании их из памяти. С него идёт сигнал на все модули ПВП. В одной ячейке памяти хранится одна микрокоманда ПВП, структура команды изображена на Рис. 3.

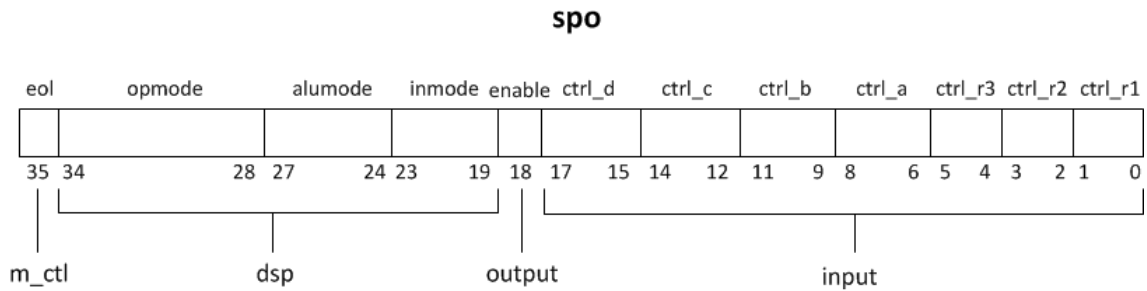


Рис. 3: Структура микрокоманды ПВП

- `eol` — Флаг конца программы, 1 бит. Сигнал идёт на контроллер памяти, оповещая его, что программа выполнена.
- `opmode`, `alumode`, `inmode` — Параметры для модуля DSP, 16 бит. Этими командами задаются простые вычислительные операции.
- `enable` — Включение вывода для модуля `output`, 1 бит. Этот сигнал подаётся, когда ПВП готов выдовать результаты вычислений.
- `ctrl_d`, `ctrl_c`, `ctrl_b`, `ctrl_a`, `ctrl_r1`, `ctrl_r2`, `ctrl_r3` — Параметры для модуля `input`, 18 бит. Используются для конфигурирования мультиплексоров.

Для упрощения программирования памяти ПВП был написан транслятор микрокода Улитиным Александром. О нём можно прочитать в его курсовой работе [3].

3.2 Модуль контроллера памяти

При реализации контроллера памяти было решено не протаскивать через него данные, посылаемые памяти. Данные, а также флаг записи идут в память напрямую из модуля ввода. Единственная задача контроллера — выдавать адреса ячеек памяти. В начальном состоянии контроллер посылает адрес `0x00`. Как только он получает сигнал записи в память или сигнал старта программы, он начинает инкрементировать адрес до тех пор, пока не пропадёт сигнал записи или не появится сигнал конца программы. Схема контроллера памяти изображена на Рис. 4.

Входные порты

- `clk` — Вход тактовых импульсов, 1 бит. Задаёт тактовую частоту работы модуля.
- `run` — Флаг старта выполнения программы, 1 бит. При появлении этого сигнала память начнёт выдавать команды для ПВП.

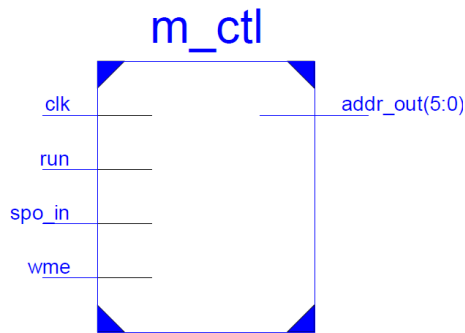


Рис. 4: Модуль контроллера памяти ПВП

- **spo_in** — Флаг конца программы, 1 бит. Этот сигнал говорит контроллеру памяти о том, что программа была выполнена. При его появлении контроллер начнёт выдавать адреса с начала (с адреса 0x00), то есть программа в памяти зацикливается.
- **wme** — Флаг записи программы в память, 1 бит. При появлении этого сигнала контроллер будет выдавать адреса для записи в память начиная с адреса 0x00 и инкрементировать адрес на каждом такте.

Выходной порт

- **addr_out** — Адрес ячейки памяти, 6 бит. Адрес, который ждёт память от контроллера.

Стоит отметить, что контроллер не разбирает конфликт одновременной записи и чтения памяти. Эту задачу было решено вывести на более высокий уровень.

4 Итоги работы

В ходе работы были изучены основы языка описания аппаратуры интегральных схем VHDL и получены базовые знания о программировании ПЛИС.

Из поставленных задач были реализованы модули памяти и контроллера памяти. Модули были проверены на симуляторе при помощи тестов. Также модули были встроены в состав ПВП и их работоспособность была проверена внутри системы.

Модули памяти (`dsp_mem_unit`) и контроллера памяти (`m_ctl_unit`) изображены в полной схеме ПВП на Рис. 5.

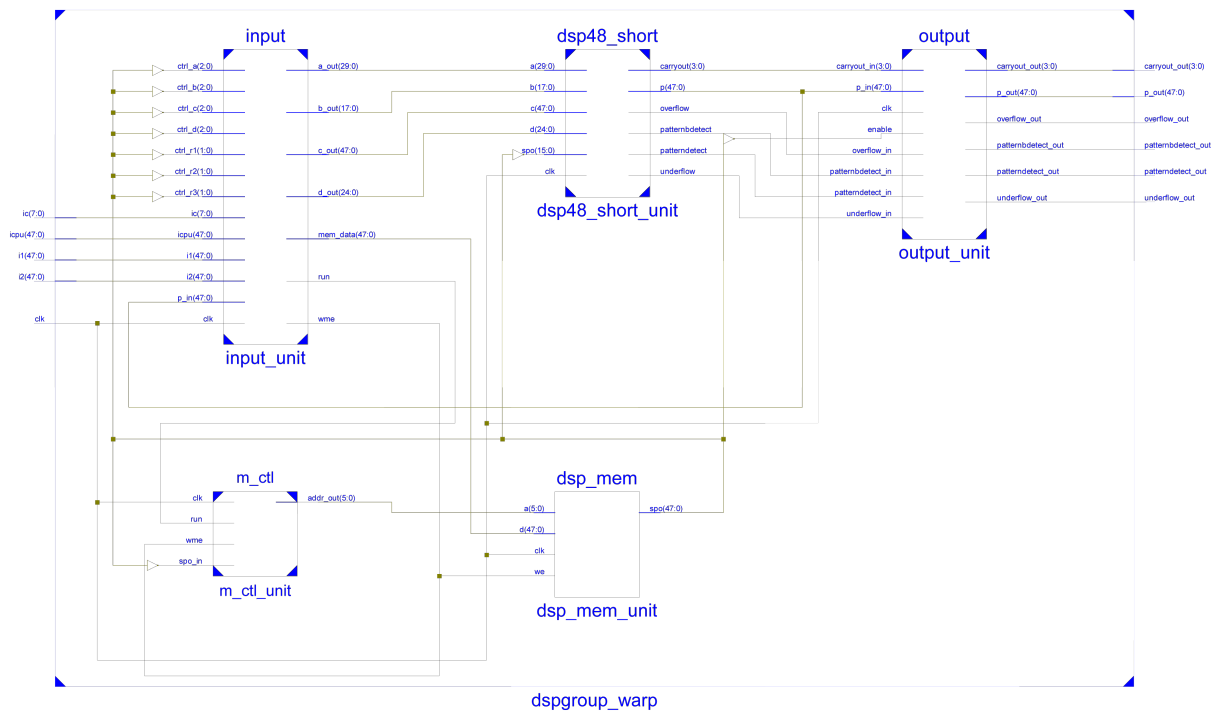


Рис. 5: Полная схема ПВП

5 Используемые технологии и средства

1. VHDL (VHSIC (Very high speed integrated circuits) Hardware Description Language) — язык описания аппаратуры интегральных схем. Использовался для написания модуля контроллера памяти и тестов.
2. Xilinx ISE — программное средство для синтеза и анализа HDL дизайна. Использовался для синтеза готовых модулей.
3. XILINX ISim — HDL-симулятор, который встроен в среду Xilinx ISE. Он позволяет нам произвести функциональное и временное моделирование для языков VHDL, Verilog и смешанных конструкций этих языков. Использовался для симуляции готовых модулей.
4. Xilinx CORE Generator (LogiCORE) — утилита для генерирования параметризованных логических ядер для FPGA. Использовалась для создания модуля памяти.

Список литературы

1. Д. Ю. Забранский. Разработка сети простых вычислительных процессоров и фильтра в рамках студенческого проекта МПВ. курсовая работа, СПбГУ Математико-Механический факультет, 2012.
2. Д. В. Солдатов. Реализация модулей ввода/вывода ПВП в связке с ядром dsp48e в рамках проекта МПВ. курсовая работа, СПбГУ Математико-Механический факультет, 2012.
3. А. А. Улитин. Транслятор микрокода для многоядерного потокового вычислителя. курсовая работа, СПбГУ Математико-Механический факультет, 2012.
4. П. Н. Бибило. *Основы языка VHDL*. Солон-Р, 2002.
5. Угрюмов Е. П. Грушвицкий Р. И., Мурсаев А. Х. *Проектирование систем на микросхемах программируемой логики*. БХВ -Петербург, 2002.
6. А. К. Поляков. *Языки VHDL и VERILOG в проектировании цифровой аппаратуры*. СОЛОН-пресс, 2003.
7. Xilinx. *Virtex-6 FPGA DSP48E1 Slice User Guide*, ug369 edition, 2011.
8. Xilinx. *Distributed Memory v7.1 Product Specification*, ds230 edition, 2005.