

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ
Математико-механический факультет
Кафедра системного программирования

Курсовая работа:

Реализация модулей ввода/вывода ПВП в связке с ядром DSP48E в рамках проекта МПВ

Выполнил студент 361гр:
Солдатов Дмитрий
Научный руководитель:
Кривошеин Б.Н.

2012 г.

Оглавление

Оглавление.....	2
Введение	3
Цель и задачи.....	3
Модуль Input.....	4
Описание портов, регистров и атрибутов	4
Внутренние регистры	4
Контроль портов DSP	5
Параллельное программирование ядер.....	5
Модуль DSP_short.....	6
Описание портов	6
Модуль Output.....	6
Заключение	8
Список литературы	9

Введение

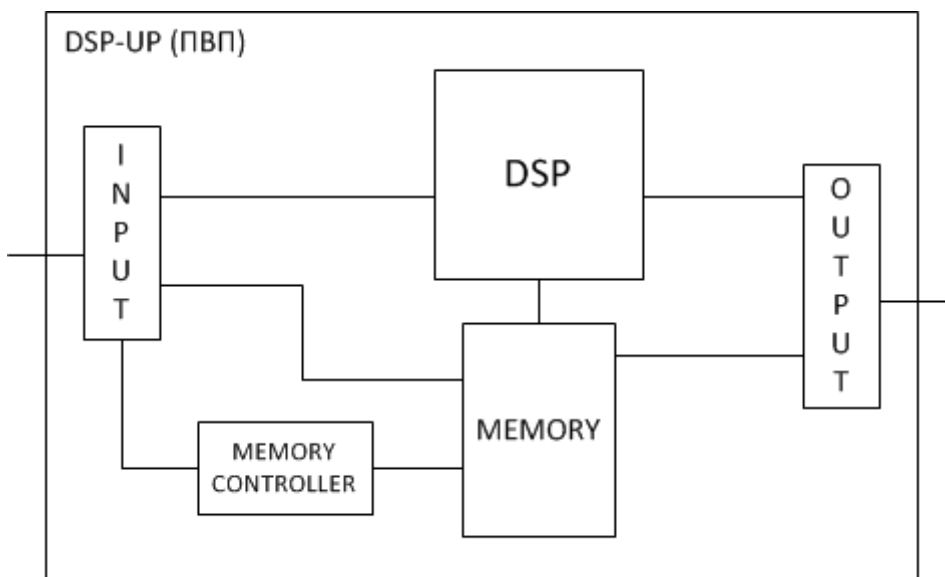
Моя курсовая проходила в рамках студенческого проекта «Разработка многоядерного потокового вычислителя», целью которого была разработка архитектуры вычислителя на базе управляющего процессора и массива простых вычислительных ядер, способных обрабатывать параллельно потоки данных. Реализация устройства проходила на основе платформы FPGA Xilinx Virtex-6 – это электронный компонент, позволяющий реализовывать интегральную схему программно [2]. В качестве языка реализации использовался VHDL.

Цель и задачи

Целью моей курсовой было создание *простого вычислительного процессора (ПВП)*.

ПВП предназначен для выполнения неких простых арифметических операций над числами. Он обладает внутренней памятью, что позволяет хранить в нем целый набор микрокоманд [4]. Таким образом, на каждом такте он может производить различные операции, не получая данные команд из внешней памяти, что существенно сократит количество обращений к ней.

Предполагается, что несколько ПВП общаются между собой, посредством сети, в которую они объединены [5]. В итоге мы имеем вычислительную сеть, позволяющую обрабатывать данные параллельно.



На рисунке изображена текущая архитектура ПВП. Как видно, здесь есть модуль памяти, и ее контроллер, о которых было упомянуто ранее. С внешней средой ПВП взаимодействует посредством модулей input и output. За вычислительные действия отвечает модуль DSP_short. Я занимался последними тремя модулями.

Таким образом, в мои задачи входило, конечно, изучить язык программирования устройств VHDL, спроектировать и реализовать взаимодействие ПВП с внешней средой, и разобраться с готовой компонентой платформы Virtex-6 dsp48e.

Модуль Input

После некоторых рассуждений было принято решение централизовать ввод ПВП, то есть один модуль отвечает за взаимодействие с внешней средой, а уж через этот модуль данные передаются в другие.

Описание портов, регистров и атрибутов

Name	Direction	Bit width	Description
NUM	Attribute	Integer	Уникальный номер ПВП. Задается при описании сети ПВП
CLK	In	1	Сигнал тактовой частоты
I1, I2	In	48	Внешние данные
P_IN	In	48	Внутренние данные, полученные с выхода этого же ПВП
IC	In	8	Управляющая команда ПВП
ICPU	In	48	Маска команд IC \ данные внутренней микрокоманды ПВП
R1, R2, R3	Register		Внутренние регистры ПВП
CTRL_R1, CTRL_R2, CTRL_R3	In	2	Контроллеры внутренних регистров R1, R2, R3
CTRL_A, CTRL_B, CTRL_C, CTRL_D	In	3	Контроллеры портов A_OUT, B_OUT, C_OUT, D_OUT
RUN	Out	1	Сигнал запуска работы ПВП
WME	Out	1	Сигнал начала записи во внутреннюю память
MEM_DATA	Out	48	Данные внутренней микрокоманды ПВП
A_OUT, B_OUT, C_OUT, D_OUT	Out	30, 18, 48, 25	Данные для модуля DSP

Внутренние регистры

В модуле Input находится 3 регистра для хранения данных (48 бит *(макс. размерность входного порта DSP)), поскольку непосредственно отсюда данные передаются на вход в DSP. Внутренние регистры могут быть нужны для констант, необходимых во время вычислений. В эти регистры можно записать данные.

Записать данные в них можно посредством портов ic и icpu, о чем будет подробнее рассказано в пункте “Параллельное программирование ядер”.

Так же, данные в регистр могут быть записаны на каждом такте посредством микрокоманды из внутренней памяти. Ниже предоставлена таблица команд управления регистрами (аналогично для всех регистров).

Код команды	Источник данных
01	P_IN
10	I1
11	I2

Контроль портов DSP

Здесь же реализован контроль портов, отвечающих за входные данные модуля DSP.

Какими способами мы можем передать данные в эти порты?

- Внешние порты (I1, I2), через которые идет поток данных из внешней памяти,
- Порт P_IN
- Внутренние регистры (R1, R2, R3).

Связь между всем этим задается посредством мультиплексоров внутри модуля Input, которые в свою очередь, на каждом такте определяют свой выбор с помощью команд из внутренней памяти. Ниже предоставлена таблица команд управления портами.

Код команды	Источник данных
101	P_IN
110	I1
111	I2
001	R1
010	R2
011	R3

Параллельное программирование ядер

Предполагается, что несколько ПВП (они же - ядра сети) подключены к одной шине, и каждое ядро имеет свой номер (num). Для того, чтобы производить какие-то вычисления, прежде всего, нужно запрограммировать каждое ядро. Это делается посредством портов ic и icpu.

Какие команды существуют, можно видеть ниже.

Код IC-команды	Наименование IC-команды	Описание IC-команды	Значение сигнала ICPU
0000001	wr1	Запись в регистр R1	Маска
0000010	wr2	Запись в регистр R2	Маска
0000011	wr3	Запись в регистр R3	Маска
0000100	wm	Запись в память	Маска
0000101	start	Начать обход внутренней памяти	Маска
0000110	stop	Остановить обход внутренней памяти	Маска

0000111	clr	Очистить внутреннюю память	Маска
0001000	Td	Идет поток данных	Данные

В первых семи случаях, показанных на таблице icpu выступает в роли маски, то есть, если маска соответствует номеру ядра, то ядро воспринимает эту команду. Такой способ организации позволяет нам параллельно программировать сразу несколько ядер.

Подробнее о последней команде td. Дело в том, что после принятия команды записать в регистр или во внутреннюю память, ядро переходит в режим принятия, собственно, тех данных, которые нужно записать. Понимание, что данные все еще идут задается командой td через порт IC, а через порт ICPU те самые данные и идут.

Модуль DSP_short

Библиотечная компонента dsp48e для virtex6 имеет порядка 50 портов и внутренних регистров, поддающихся управлению. На стадии создания прототипа устройства было решено выделить основные из них, которые мы будем использовать непосредственно, а остальные зафиксировать или же отключить. Таким образом, модуль DSP_short является оболочкой на библиотечной компонентой dsp48e.

Описание портов

Name	Direction	Bit width	Description
CLK	In	1	Сигнал тактовой частоты
A, B, C, D	In	30, 18, 48, 25	Входные данные
SPO	In	48	Код операции над числами
P	Out	48	Результат вычислений
OVERFLOW	Out	1	Индикатор переполнения сигнала P
PATTERNDETECT	Out	1	Индикатор совпадения с шаблоном указанным в атрибуте PATTERN
CARRYOUT	Out	4	
UNDERFLOW	Out	1	Индикатор переполнения сигнала P

Подробнее о dsp48e можно прочесть в оригинальной документации Xilinx [2].

Модуль Output

На данный момент модуль не содержит особой функциональности, просто перебрасывает результаты вычислений наружу. Единственное, что хочется сказать здесь, что результаты так же передаются на вход этому же ПВП (порт P_IN в модуле Input), это может быть полезным, когда над результатом нужно произвести повторно такие же вычисления.

В дальнейшем в этом модуле предполагается реализация операции деления, как операции обратной умножению.

Заключение

В процессе работы были достигнуты следующие результаты:

- Изучены основы VHDL
- Реализованы модули вводы и вывода с упором на параллельное программирование ядер
- Адаптирован существующий модуль DSP48e
- Протестирована работоспособность модулей в рамках симуляции

Список литературы

- [1] Р. Грушвицкий “Проектирование систем на микросхемах программируемой логики”
- [2] Xilinx “Virtex-6 FPGA DSP48E1 Slice User Guide” (ug369)
- [3] <http://xilinx.com>
- [4] Курсовая работа, мат-мех, 2012, гр.361, Е. Тодорук “Разработка модуля памяти для многоядерного потокового вычислителя”
- [5] Курсовая работа, мат-мех, 2012, гр.361, Д. Забранский “Разработка сети простых вычислительных процессоров и фильтра в рамках студенческого проекта МПВ”