

Санкт-Петербургский государственный университет

Математико-механический факультет

Кафедра системного программирования

Курсовая работа:

Разработка сети простых вычислительных процессоров и фильтра в  
рамках студенческого проекта МПВ

Автор: Забранский Дмитрий, 361 гр.

Научный руководитель: Кривошеин Б. Н.

24 мая 2012 г.

# Содержание

<b>1</b>	<b>ВВЕДЕНИЕ</b>	<b>1</b>
1.1	МНОГОЯДЕРНЫЙ ПОТОКОВЫЙ ВЫЧИСЛИТЕЛЬ	1
1.2	ПРОСТОЙ ВЫЧИСЛИТЕЛЬНЫЙ ПРОЦЕССОР	2
<b>2</b>	<b>ПОСТАНОВКА ЗАДАЧИ</b>	<b>2</b>
2.1	ЗНАКОМСТВО С ПРЕДМЕТНОЙ ОБЛАСТЬЮ	2
2.2	РАЗРАБОТКА СЛОЯ И СЕТИ ПВП	2
2.3	РАЗРАБОТКА ФИЛЬТРА	2
<b>3</b>	<b>РЕАЛИЗАЦИЯ</b>	<b>3</b>
3.1	Слой ПВП	3
3.2	СЕТЬ ПВП	4
3.3	ФИЛЬТР ЛАПЛАСА	5
3.3.1	<i>Описание фильтра.....</i>	<i>5</i>
3.3.2	<i>Реализация.....</i>	<i>6</i>
<b>4</b>	<b>ТЕХНОЛОГИИ</b>	<b>8</b>
<b>5</b>	<b>ЗАКЛЮЧЕНИЕ</b>	<b>8</b>
<b>6</b>	<b>СПИСОК ЛИТЕРАТУРЫ</b>	<b>9</b>

# 1 Введение

## 1.1 Многоядерный потоковый вычислитель

Данная курсовая работа была проделана в рамках студенческого проекта: “Разработка многоядерного потокового вычислителя (МПВ)”. Задачей проекта является разработка архитектуры вычислителя на базе управляющего процессора и массива простых вычислительных ядер, способных обрабатывать параллельно потоки данных.

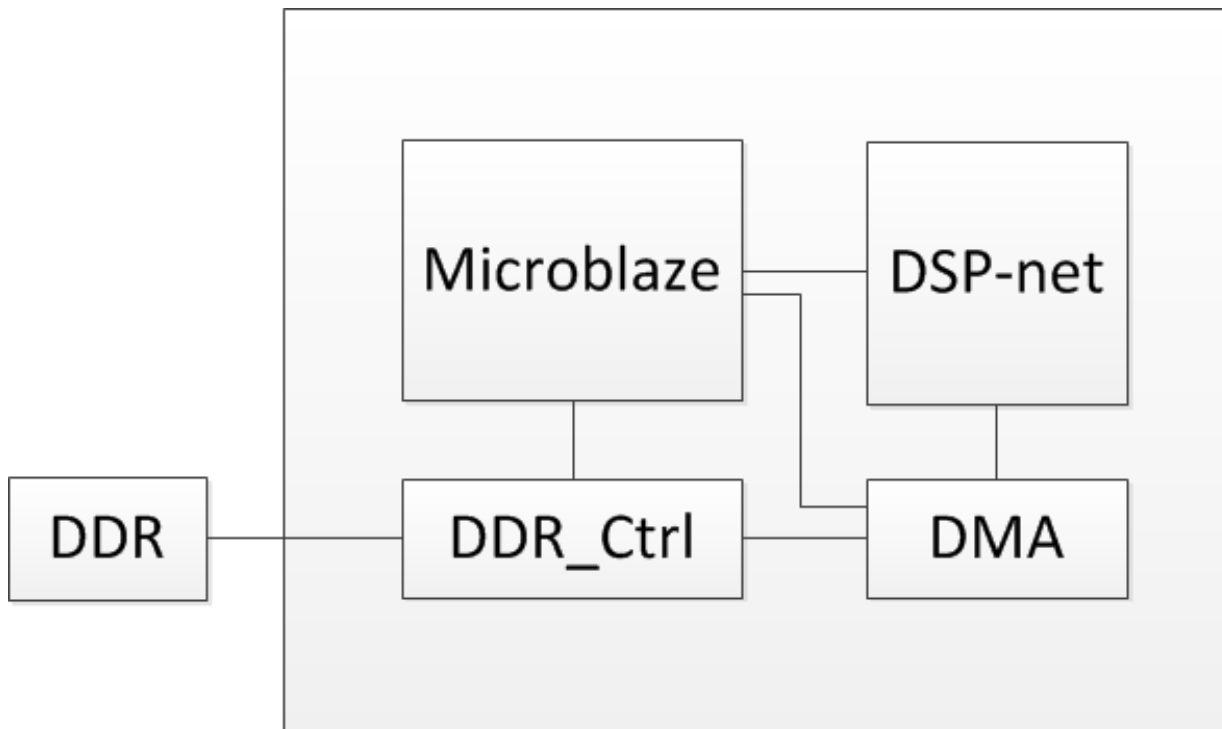


Рис. 1: Структура МПВ

В прошедшее полугодие участниками студ. проекта была выполнена следующая мини-задача: разработка прототипа на ПЛИС с минимальным количеством потоковых сопроцессоров и масштабируемой архитектурой.

Которая в свою очередь была разбита на подэтапы:

- 1) Изучение языка VHDL и основ программирования ПЛИС;
- 2) Проектирование архитектуры простого вычислительного процессора (ПВП, DSP-group);
- 3) Создание компонент DSP-group: DSP-ядро, внутренняя память, контроллер памяти, контроллеры входных и выходных данных;
- 4) Создание слоя и сети DSP-group;
- 5) Тестирование отдельных компонент и всей модели.

Я занимался проектированием слоя и сети ПВП, начал тестировать данную часть, то есть реализовывать фильтр Лапласа на нашем вычислителе.

## **1.2 Простой вычислительный процессор**

Данный процессор может выполнять простые операции, такие как сложение, вычитание, умножение чисел. Обладает внутренней памятью команд, позволяющей запрограммировать вычисление ряда последовательных операций.

ПВП был разработан нашим студ. проектом, а также послужил элементом построения слоя и сети.

## **2 Постановка задачи**

### **2.1 Знакомство с предметной областью**

Первой целью работы было изучение языка описания аппаратуры интегральных схем VHDL и основ программирования FPGA. Данный этап был очень важен для понимания сути и структуры нашего проекта.

### **2.2 Разработка слоя и сети ПВП**

Следующей целью было спроектировать особые блоки ПВП, названные слоями, в которых ПВП не были бы соединены между собой, также стояла задача объединить ПВП так, чтобы полученная архитектура была масштабируема и соединение между слоями ПВП задавали статическим образом.

### **2.3 Начало разработки фильтра**

Сеть и слой нужно протестировать, был выбран фильтр Лапласа, как актуальный для нашего вычислителя, так как отсутствие деления мешало реализации более сложных тестов. Разработка включает в себя несколько этапов, часть из которых было сделано.

# 3 Реализация

## 3.1 Слой ПВП

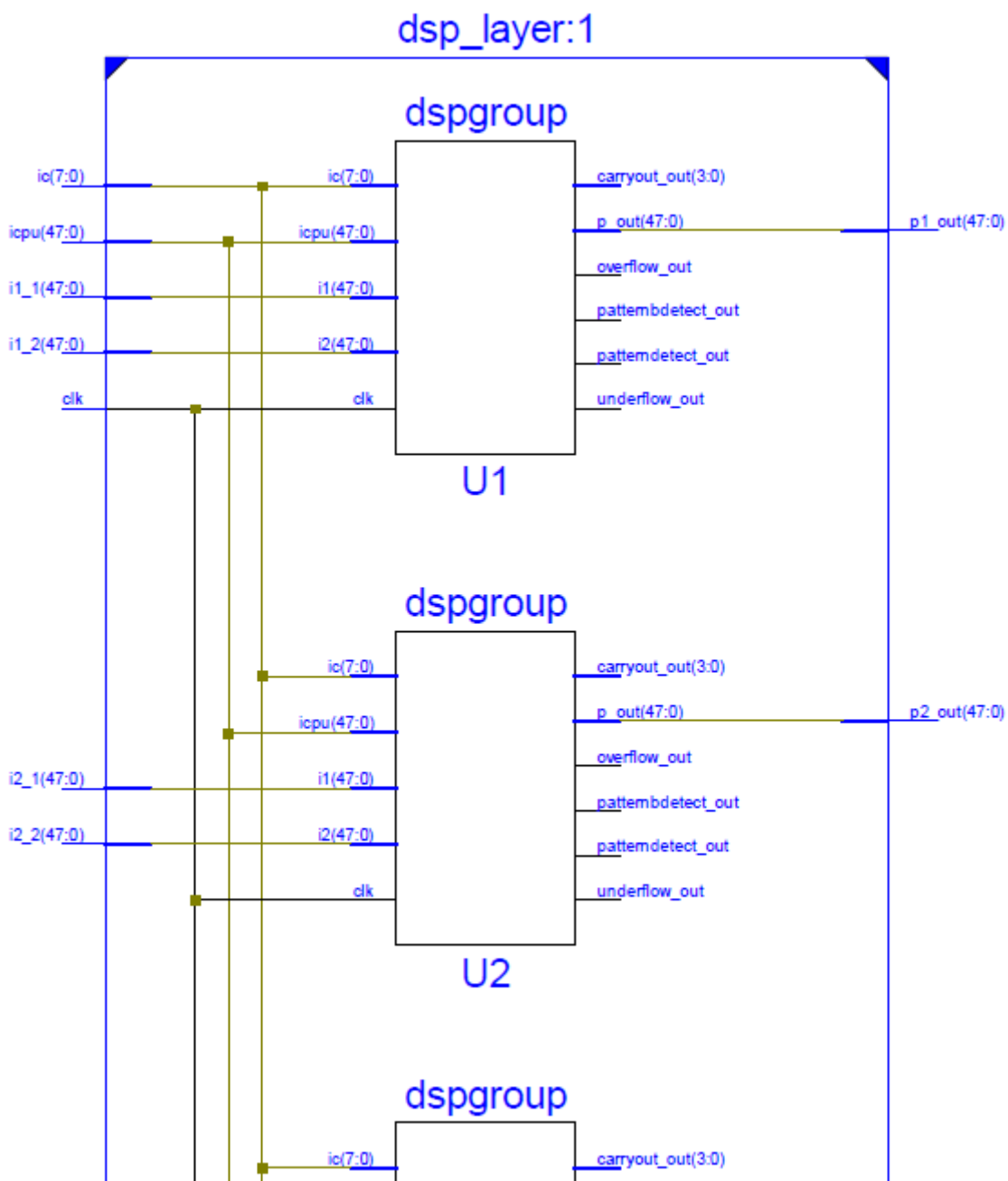


Рис. 2: Структура слоя

Слой был сделан таким образом, чтобы вычисления можно было вести параллельно, и каждый процессор был независим от других процессоров в слое. На данном рисунке показана общая схема слоя ПВП. Количество процессоров в нем можно настраивать вручную. Для каждого такого процессора рассчитано два порта ввода и один вывода.

## 3.2 Сеть ПВП

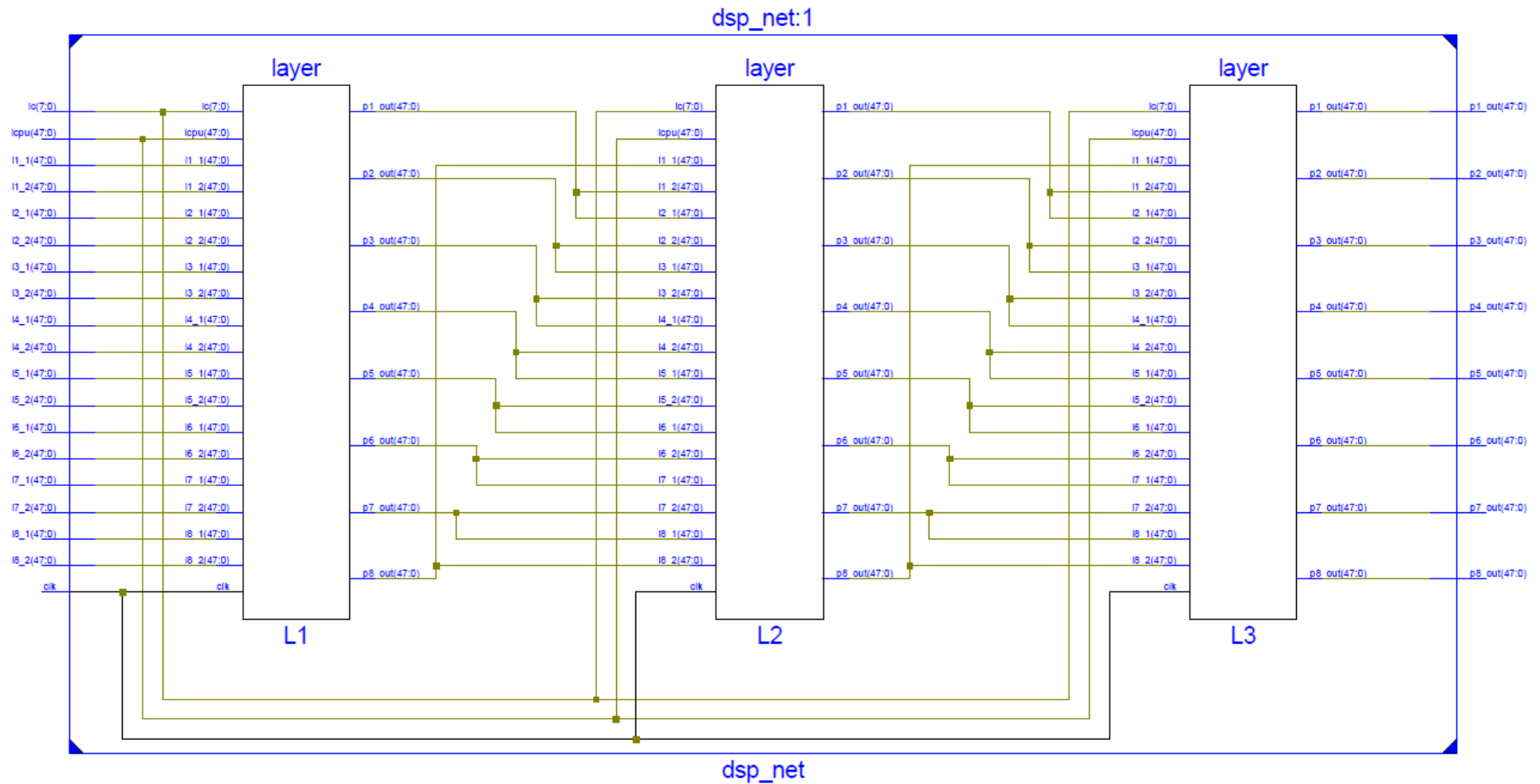


Рис. 3: Структура сети ПВП

Данная структура сети обладает масштабируемой архитектурой, то есть мы можем вручную задавать количество слоев и элементов в слое. В нашем случае есть три слоя по восемь ПВП в каждом. Слои соединены между собой статическими связями, то есть по ходу работы изменить их нельзя. На данную структуру хорошо ложатся цифровые фильтры и некоторые алгоритмы.

### 3.3 Фильтр Лапласа

#### 3.3.1 Описание фильтра

После проектирования и реализации сети ПВП нужно провести тестирование данного модуля. Была выбрана задача выделения границ. Одним из подходов к выделению границ является применение к изображению фильтра Лапласа, который ищет границы между разными цветами, таким образом, находя контуры объектов.

$$L_{ker} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Этот фильтр выделяет края по методу Лапласа, который производит края шириной в один пиксель. Стоит заметить, что выбор данного фильтра обусловлен отсутствием деления у нашего вычислителя в данный момент времени.

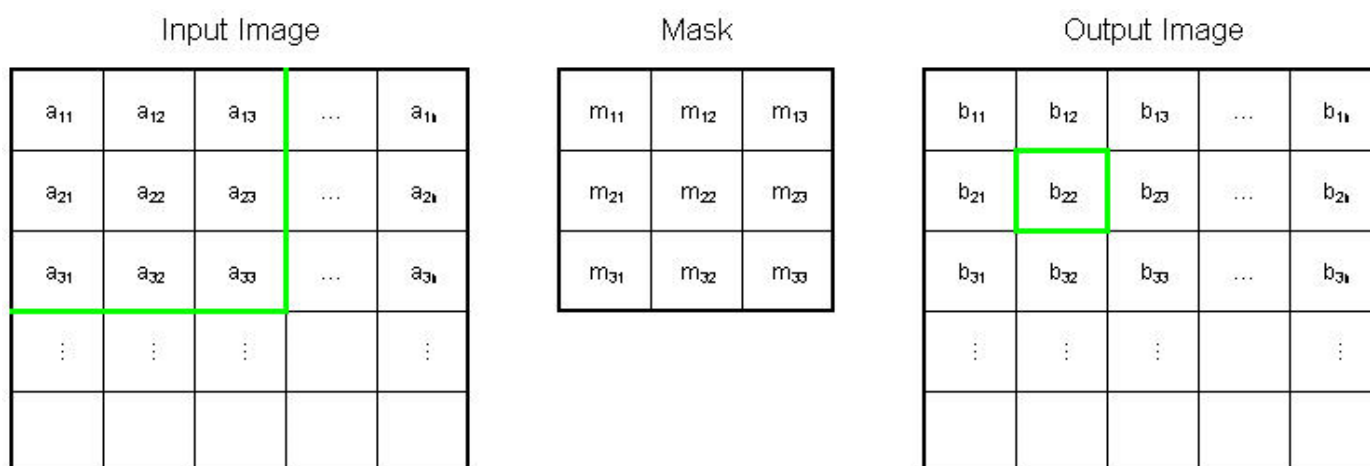


Рис. 4: Вычисление нового значения пикселя

Значение пикселя вычисляется следующим образом:

$$b_{22} = \sum_{i=1}^3 \sum_{j=1}^3 a_{ij} \times m_{ij}$$

Пиксели на краях матрицы принимаем равными нулю.

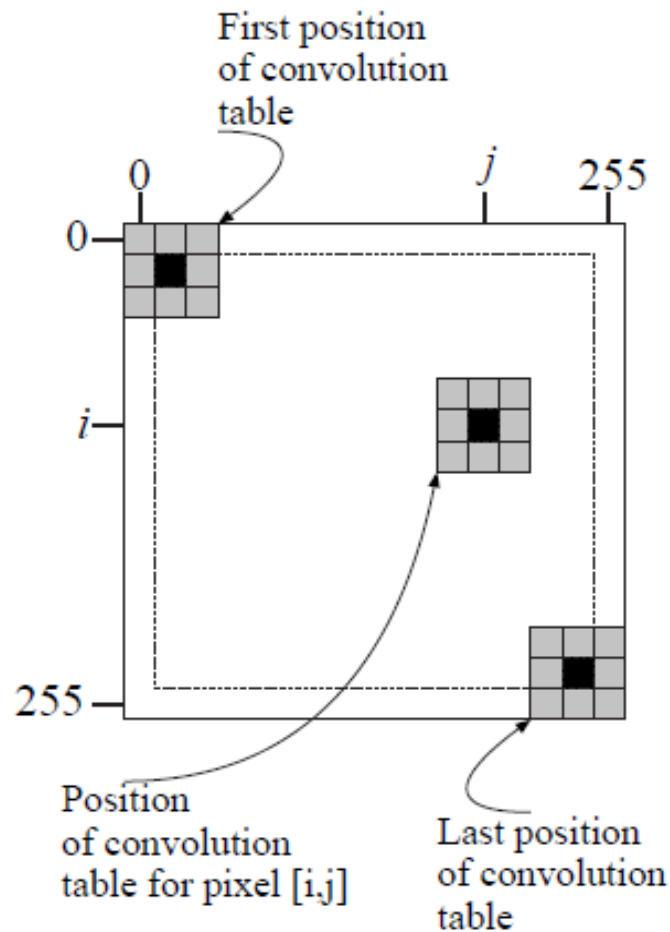


Рис. 5: Обход изображения при вычислении свёртки

### 3.3.2 Начало реализации

Первое, что нужно сделать, это перевести изображение в матрицу с кодами соответствующих цветов. Для этого я использовал небольшую программу на Matlab'е.

```
function [ ] = img2bin( file_in, file_out )
    a = rgb2gray(imread(file_in));
    fid = fopen(file_out, 'w');
    fwrite(fid, size(a), 'uint16');
    fwrite(fid, a, 'uint8');
    fclose(fid);
end
```

Данная программа принимает на вход изображение, которое переводится в серые тона, на выходе получается файл, где хранится в побитовом представлении длина и ширина изображения, коды цветов изображения.

Вторым этапом нужно произвести считывание данной информации в наш вычислитель. Это было реализовано с помощью VHDL, был получен поток данных, который мы будем подавать на вычислитель.



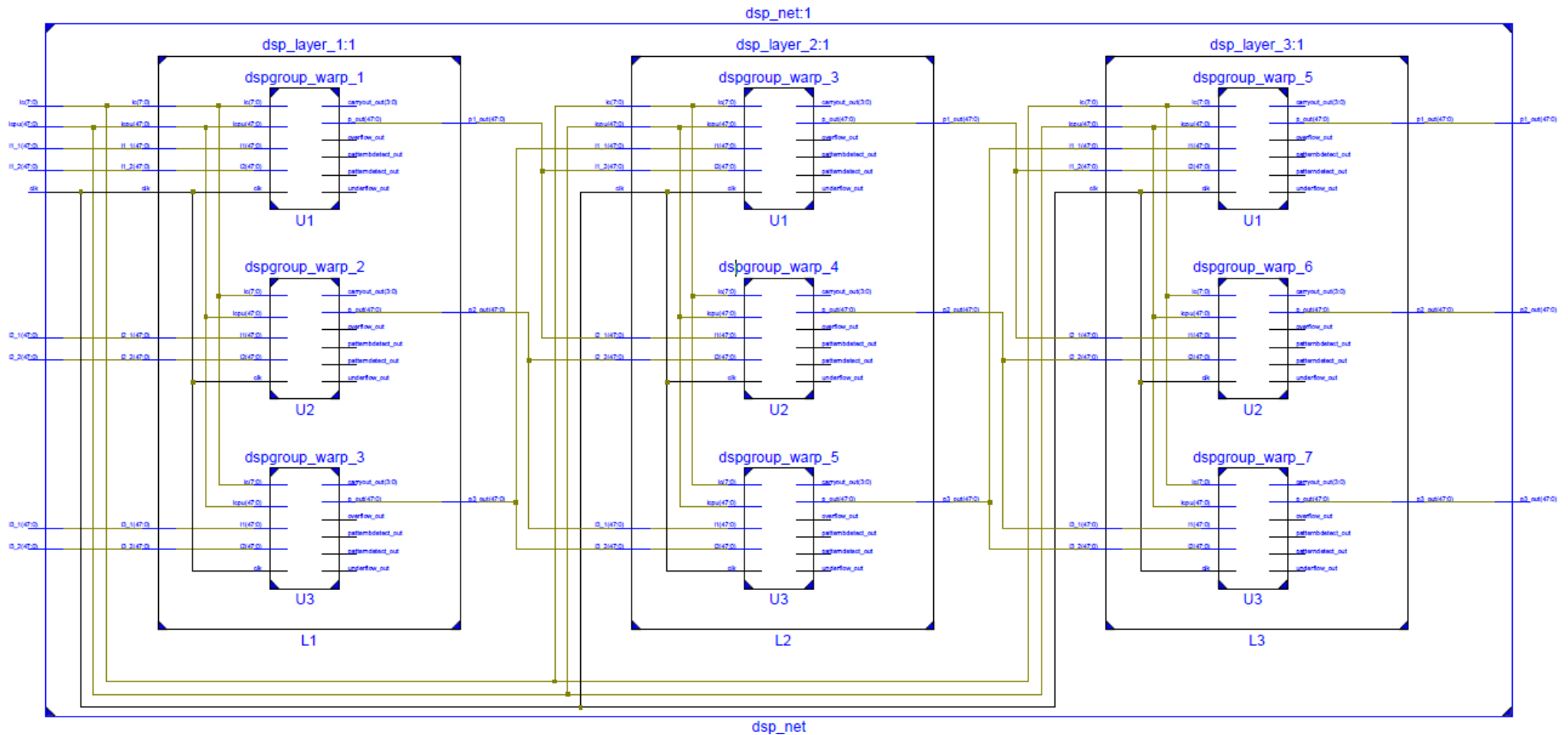


Рис. 6: Структура сети ПВП для фильтра Лапласа

Далее была создана структура сети под нашу задачу, три слоя по три ПВП в каждом. Для вычисления нового значения пикселя следует проводить четыре операции сложения и одну операцию умножения, из-за этого три процессора будут не использованы, два процессора будут работать вхолостую (через них будет просто передаваться значение кода цвета пикселя). Фильтр Лапласа ложится на эту структуру, что показывает универсальность нашей сети.

Дальнейший этап в разработке фильтра Лапласа: запрограммировать каждый ПВП под свою задачу, то есть написать программу для вычислителя.

## 4 Технологии

При написании курсовой работы использовались следующие технологии:

- VHDL (VHSIC (Very high speed integrated circuits) Hardware Description Language) - язык описания аппаратуры интегральных схем. Использовался для написания модуля контроллера памяти и тестов.
- Xilinx ISE - программное средство для синтеза и анализа HDL дизайна. Использовался для синтеза готовых модулей.

## 5 Заключение

В процессе работы были достигнуты следующие результаты:

- Изучены основы VHDL,
- Изучен процесс проектирования ПЛИС с помощью Xilinx ISE,
- Реализованы слой и сеть ПВП,
- Частично реализован фильтр Лапласа.

## 6 Список литературы

- [1] Грушвицкий Р. И., Мурсаев А. Х., Угрюмов Е. П. Проектирование систем на микросхемах программируемой логики. БХВ -Петербург, 2002.
- [2] Тодорук Е. А. Разработка модуля памяти для многоядерного потокового вычислителя. Курсовая работа, СПбГУ Математико-Механический факультет, 2012.
- [3] Солдатов Д. В. Реализация модулей ввода/вывода ПВП в связке с ядром dsp48e в рамках проекта МПВ. Курсовая работа, СПбГУ Математико-Механический факультет, 2012.
- [4] Xilinx. Virtex-6 FPGA DSP48E1 Slice User Guide, ug369 edition, 2011.
- [5] Миллер Р., Боксер Л. Последовательные и параллельные алгоритмы