SYNOPSIS

ON

REDUCING COMPUTATIONAL COMPLEXITY OF MATHEMATICALFUNCTIONS USING FPGA

To be submitted by

NEHA GOUR, ABMTV16030

Under the supervision of

Prof. ARUP BANERJEE
RAJA RAMANNA CENTRE FOR ADVANCED TECHNOLOGY,
INDORE-452013

For the award of degree

Of

M.TECH
VLSI DESIGN

DEPARTMENT OF ELECTRONICS,
BANASTHALI VIDYAPITH
BANASTHALI – 304022
MAY-2018

TABLE OF CONTENT

List of Figures

1. INTRODUCTION

1.1 Background

Gone are the days when huge computers made of vacuum tubes in entire dedicated rooms and could do about 300 multiplications of 10 digit numbers in a second. Now, modern day's computers are getting smaller, faster, and cheaper and more power efficient every progressing second. The main key to attain these features is a choice of suitable system architecture since it determines to a high degree of integration. Because of ever increasing integration level of device becoming complex day by day. The complex designs demands larger computational capacity. By increasing the number of logic gates on a single chip computational complexity increases day by day. In this project, we will reduce the computational complexity using Field Programmable Gate Arrays (FPGAs).

1.2 Motivation

By the early, 1980's Large Scale Integrated circuit (LSI) formed the backbone of the logic circuits in major systems. Microprocessor, timer, etc. were implemented using integrated circuit fabrication technology. Interconnects were still require to help connect large ICs. Initial attempt to solve this problem led to development of **Custom ICs** (microcontroller) which were replace large amount of interconnections.

Field Programmable Gate Arrays (FPGA) were introduced as an alternative for custom ICs for implementing entire system on one chip. Field Programmable Gate Arrays (FPGA) is a most popular logic circuit component that allows to users to change their design after the end product has been manufactured. Basically FPGA eliminates the cost associated with redesigning. FPGA devices deliver the flexibility of re-programmability. The flexibility offered by FPGAs reduces design time and possibilities to implement a complex logic in a real time are among them. Another advantage of FPGA over microcontroller are having programmable Arithmetic Logic Unit (ALU), and FPGA works on high frequency range unlike microcontroller works on lower frequency range compare to FPGA. This is very much significant compared to any of microcontroller [1].

1.3 Project Goal

In this project, we propose a scheme for the designing a device for reducing computational complexity of mathematical function by FPGA. Computationally intensive applications are usually run on high-end servers which contain general purpose processors (similar to microcontrollers). These general purpose processors run all the computations in sequence. In this project we propose performing these operations in a parallel manner by using FPGAs.

In mathematics large-range of mathematical function developed, but here we will consider matrix multiplication as a mathematical example for reducing the computational complexity using FPGA. Matrix multiplication is commonly used in most signal processing applications. This matrix multiplication is a core operation in a wide variety of applications such as machine learning, robotics, signal processing, graphics, etc. Matrix multiplication involves a large number of multipliers involving longer delay, consumes large power. Registers, which are required to store the intermediate product values, are also major power intensive component [1]. These components pose a major challenge for designing VLSI structures for large-order matrix multipliers with optimized speed. This technique has been designed implementation on Artix-7 FPGA. We synthesized the design on existing tool VIVADO (2017.2) by using Very high speed integrated circuit Hardware Description Language (VHDL).

The project involved here is to explore using FPGA for performing highly computationally intensive calculations. The project can be divided into three phases.

1. Data transfer from computer to the FPGA board.
2. Performing mathematical operations in FPGA and pipelining the code for parallelization.
3. Resultant data transfer from FPGA board to computer.

We have proposed design for implementing the matrix multiplication operation in hardware keeping the goal of speed efficient architecture. In chapter 2, literature survey has been discussed. In Chapter 3, work plan has been discussed. In Chapter 4, conclusions and future scope of the design has been discussed.

2. LITERACTURE SURVEY

2.1 Field Programmable Gate Array (FPGA)

2.1.1 Overview of an FPGA

Field Programmable Gate Arrays (FPGA) is a Programmable Logic Device (PLD) that contains a matrix of reconfigurable gate array logic circuitry. The word Field in the name refers to the ability of the gate arrays to be programmed by the user instead of by the manufacturer of the device. The word Array is used to indicate a series of columns and rows of gates that can programmed by the end user [2]. When FPGA is configured, the internal circuitry is connected in a way that creates a hardware implementation of software application. FPGA are parallel in nature unlike the microprocessor or microcontroller, so different processing operation does not have to compete for same resources. As a result, the performance of one part of application is not affected when additional processing is added. FPGA devices deliver the performance and reliability of dedicated hardware circuitry.

2.1.2 Architecture of an FPGA

FPGA is defined as a matrix of configurable logic blocks (CLBs), linked to each other by interconnection network, which is entirely reprogrammable.

The FPGA architecture consists of three types of configurable elements-

I. IOBs – a perimeter of input/output blocks.
II. CLBs- a core array of configurable logic blocks.
III. Resources for interconnection.

The IOBs provide a programmable interface between the internal array of logic blocks (CLBs) and the device's external package pins. CLBs perform user-specified logic functions, and the interconnect resources carry signals among the blocks [2].

2.1.3 Artix-7 FPGA

Artix-7 devices for low power and low cost. The Xilinx Artix-7 family of FPGA offers the fastest line rates for cost sensitive applications.Artix-7 FPGA optimized for high performance logic, and offers more capacity, higher and more resources than earlier designs. Artix-7 family delivers maximal bandwidth with its sixteen 6.6 Gb/s transceivers that have been optimized for low power [5]. Due to the programmable nature, FPGAs are an ideal fit for many different markets. As the industry leaders, Xilinx provides comprehensive solution made up of FPGA devices, advanced software, configurable and read to use IP cores for market and applications [4].

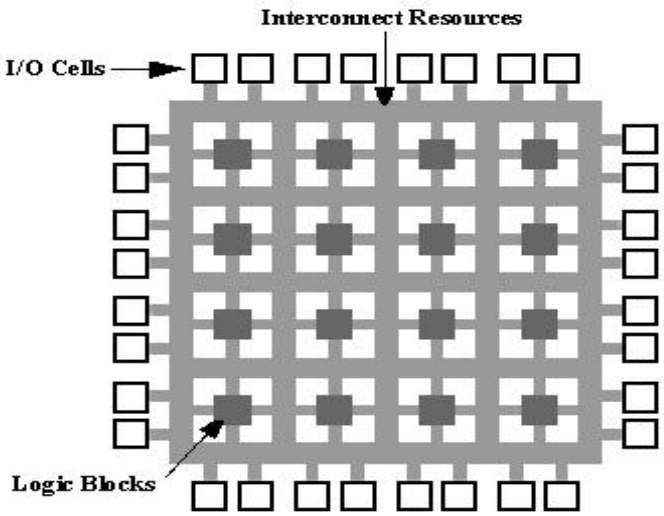

Figure 2.1.3.1: Internal structure of an FPGA

2.2 High Level Synthesis (HLS)

2.2.1 Introduction

High Level Synthesis (HLS) transform a C specification into a Register-Transfer Level (RTL) implementation that synthesize into Xilinx Field Programmable Gate Array (FPGA).High Level Synthesis (HLS) bridges the software and hardware domain.

- HLS allow hardware designer who implement design in an FPGA to take an advantage of productivity benefits of working at higher level of abstraction, while creating high performance hardware.

➢ Provides software developers with an easy way to accelerate computationally intensive part of their algorithm on a new compilation target.

The primary benefit of a HLS design methodology to improved the productivity for hardware designers and improved system performance for software engineers [3].

2.2.2 Algorithm of HLS: Scheduling and Binding

Many implementations are possible from same source description such as smaller design, faster design, optimal design, and enable design exploration with directives. Understand the priority of directives:

- Meet performance (clock & throughput).
- Then minimize latency.
- Then minimize the area.

With the help of VHLS control and data flow maps into a hardware design through scheduling and binding process. Scheduling and binding are the process at the heart of High Level Synthesis.

- Scheduling determines in which clock cycle operation will occur.
- Binding determines which library cell is used for each operation [8].

3. PLANNING OF WORK

In this project, we develop the algorithm at C level and transform into Register Transfer level (RTL) by using High Level Synthesis (HLS) and we will check the performance of Hardware. We are using different directives such as unrolling, pipelining etc. and measure the performance of hardware (Artix-7) created by high level synthesis. Firstly Next, we shall move to mathematical function (matrix multiplication) and take firstly 3*3 matrix and next we shall apply the same algorithm to larger matrix and see how they reduce the clock cycle and increase the hardware using the FPGA. Before moving to the matrix multiplication, we will apply these optimization techniques on basic operation such as addition of fixed point numbers and floating point numbers and compare the result with conventional processors. This project is based on many Digital signal processing application and programming them into hardware platform.

The Xilinx Artix-7 family of FPGAs has redefined cost-sensitive solutions by cutting power consumption in half from the previous generation while providing advanced functionality applications. The newest generation of 7 series devices are built on advanced 28nm process technology to produce the lowest-cost, lowest power FPGA for products like portable medical equipment, military radios etc.

4. CONCLUSION AND FUTURE SCOPE

This research work is aimed at proposing a simple architecture of design of reduction of computational complexity of Mathematical function using FPGA technology. Most of the digital signal processing (DSP) algorithms is formulated as matrix-matrix multiplication. The size of matrix multiplication is usually large for various practical applications. Most of these algorithms are currently implemented in hardware to meet the requirement of real-time applications. When large size matrix multiplication is implemented in hardware, consumes large amount of chip area and power. With such a vast application domain, new designs are required to the constraints of chip area and power and high speed. In this context, we have proposed design for high speed of matrix-matrix multiplication using FPGA (Artix-7).We have compared the proposed designs with the existing similar design and found that, the proposed designs offer higher throughput rate at relatively lower hardware cost.

As observed through performance comparison, proposed parallel design consumes significantly less energy than the other existing design. This is mainly due to the large number of resources of the design.

REFERENCES

[1] *"Field programmable gate arrays and Applications"*, Version 2 EE IIT, Kharagpur.

[2] www.srmuniv.ac.in/ramapuram/sites/ramapuram/files/part2.pdf

[3] *"Vivado design user guide high level synthesis"* UG902 MAY,30 2014

[4] Deepak Raj, Dr. Poonam singhal *"Reducing the Computational Complexity Of A 2d Gaussian Filter for Image Processing (An Overview)"* International Journal of Advance Research, Ideas and Innovations in Technology, Volume3, Issue3.

[5] Ehab Mohsen *"Reducing system power and cost with Artix-7 FPGAs"* WP423 (v2.1) February 7, 2013

[6] Volnei A. Pedroni *"Circuit Design with VHDL"*.

[7] M. Morris Mano, Michael D. Ciletti *"DIGITAL DESIGN with an Introduction to the VHDL"* fifth edition.

[8] *"Introduction to High Level Synthesis with VIVADO HLS"* Vivado HLS Version 2013.3.

S