Chapter 1

Introduction

The demand for high speed processing day by day has been increasing as a result of expanding computer and signal processing applications. Higher throughput arithmetic operation is important to achieve the desired performance in many real-time signal and image processing applications. One of the key arithmetic operations in such application is multiplication and the development of fast multiplier circuit has been a subject of interest over decades. Reducing time delay and power consumption are very essential requirement for many application. In any processor the major unit is control unit, ALU and memory read write. ALU is an execution unit which not only performs the arithmetic operation but also logical operation and therefore ALU is called a heart of Microprocessor, Microcontroller, and CPUs. In every technology, the operations are either fully or partially performed by ALU.

Earlier multiplication comprises a succession of addition and subtraction followed by shifting the operands. By now array multiplication algorithm and booth multiplication algorithm are used in the digital hardware. The word "VEDIC" is a consequential of "VEDA" comprising accumulation of knowledge at single platform. Furthermore, it exhibits 16 Sutras that deals with several subdivisions including arithmetic, algebra, geometry and many more. Jagadguru Swami Sri Bharati Krishna Tirthaji in between 1884-1960, projected the concept of this ancient methodology that became very popular to achieve high speed processing of the data.

Vedic mathematics is used to solve the complex calculations involve in usual mathematics. Owing to its simple strategy, which mainly includes natural viewpoints of human being, it leads to a straightforward process. It consents to incorporate the arithmetic rules along with high speed and easy implementation, thereby viable for a range of applications based on computing. The Vedic multiplier is based on the algorithm named as "Urdhva Tiryakbhyam" sutra. Tradionally, this well known Sutra has been employed to multiply two given numbers in a decimal number system. It is a universal multiplication formula that can be used in for all multiplications. It literally means "Vertically and Crosswise". The algorithm is viable for multiplication of any two numbers exhibiting bit length equal to n. Besides this, it introduces a parallel execution of

partial products and sums. The clock frequency is not an obstacle in accounting the calculations. This technique has proven advantageous in combating with large delays and complexity of conventional multipliers.

Vedic mathematics is an ancient technique, which simplifies multiplication, divisibility, complex numbers, squaring, cubing, square and cube roots. In the Vedic system 'difficult' problems or huge sums can often be solved immediately by the vedic method. These are the part of a complete system of mathematics which is far more systematic than the modern system. Vedic mathematics can be used effectively for solving divisions, reciprocals, factorization, HCF, squares, square roots, cube, cube roots, algebraic equations, multiple simultaneous equations, quadratic equations, bi quadratic equations, cubic equations, higher degree equations, differential calculus, partial fractions, integrations, Pythagoras theorem, Apollonius theorem, analytical conics and so on.

Vedic Mathematics is the name given to the ancient system of Mathematics which was rediscovered from the Vedas between 1911 and1918 by Sri Bharati Krishna Tirthaji (1884-1960).According to his research all of Mathematics is based on 16 sutras or word formulae. These formulae describe the way the mind naturally works and are therefore a great help in directing to the appropriate method of solution.

Perhaps the most striking feature of the Vedic system is its coherence. The whole system is interrelated and unified: Vedic mathematics manifests the coherent and unified structure of mathematics and the methods are complementary, direct and easy. The simplicity of mathematics means that the calculations can be carried out mentally. There are many advantages in using a flexible, mental system.

The 16 sutras are the basis of Vedic mathematics, which attribute a set of qualities of a number or a group of numbers. The table 1 shows the 16 sutras. The sutras required for this project work are described briefly.

Table 1.1 Vedic Sutras

#	Name	Corollory	Meaning
1	Ekadhikena Purvena	Anurupyena	By one more than the previous one
2	Nikhilam Navatashcaramam Dashatah	Sisyate Sesasamjnah	All from 9 and the last from 10
3	Urdhva-Tiryagbyham	Adyamadyenantyamantyena	Vertically and crosswise
4	Paraavartya Yojayet	Kevalaih Saptakam Gunyat	Transpose and adjust
5	Shunyam Saamyasamuccaye	Vestanam	When the sum is the same that sum is zero
6	Anurupye Shunyamanyat	Yavadunam Tavadunam	If one is in ratio, the other is zero
7	Sankalana-vyavakalanabhyam	Yavadunam Tavadunikritya Varga Yojayet	By addition and by subtraction
8	Puranapuranabyham	Antyayordashake'pi	By the completion or non-completion
9	Chalana-Kalanabyham	Antyayoreva	Differences and Similarities
10	Yaavadunam	Samuccayagunitah	Whatever the extent of its deficiency
11	Vyashtisamanstih	Lopanasthapanabhyam	Part and Whole
12	Shesanyankena Charamena	Vilokanam	The remainders by the last digit
13	Sopaantyadvayamantyam	Gunitasamuccayah Samuccayagunitah	The ultimate and twice the penultimate
14	Ekanyunena Purvena	Dhvajanka	By one less than the previous one
15	Gunitasamuchyah	Dwandwa Yoga	The product of the sum is equal to the sum of the product
16	Gunakasamuchyah	Adyam Antyam Madhyam	The factors of the sum is equal to the sum of the factors

Sutra 1 - Ekadhikena Purvena

Ekadhikena Purvena (One More than the Previous) is a sutra useful in finding squares of numbers (like 25x25, 95x95, 105x105, 992x992 etc) and special divisions like 1 divided by 19, 29, 39, …. 199 etc. just in one step.

Division:

To Divide 1 by numbers ending with 9 like 1 divided by 19, 29, 39, ….. 119 etc. is a tedious work, using conventional method. Some of these numbers like 19, 29, 59 are prime numbers and so cannot be factorized and division becomes all the more difficult and runs into many pages in the present conventional method and the chances of making mistakes are many.

The Vedic Solution is obtained by applying the Sutra (theorem) ***Ekadhikena Purvena*** which when translated means **By one more than the previous one.**

Multiplication:

This sutra can also be used to multiply two numbers.Consider numbers AB,AC in tenths place and B/C in ones place and B+C=10. Then, AB x AC = **(Ax(A+1))(BxC)** .

Sutra 2 - Nikhilam Navatashcaramam Dashatah

Nikhilam sutra (All from9 and the last from 10) basically means start from the left most digit and begin subtracting '9' from each of the digits; but subtract '10' from the last digit.

This sutra stipulates subtraction of a number from the nearest power of 10 ie 10, 100, 1000, etc. The powers of 10 from which the difference is calculated is called the Base. These numbers are considered to be references to find out whether given number is less or more than the Base. If the given number is 104, the nearest power of 10 is 100 and is the base. Hence the difference between the base and the number is 4, which is Positive and it is called NIKHILAM. The value of Nikhilam may be reference base, the Nikhilam of 87 is -13 and that of 113 is +13 respectively.

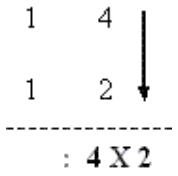
Sutra 3 - Urdhava Tiryagbyam

Urdhava-tiryagbhyam sutra is applicable to all cases of multiplication and in the division of a large number by another large number. It means "Vertically and cross wise".

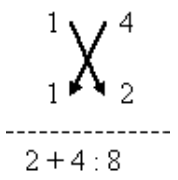
Ex1: Find the product of 14X12

The symbols are operated from right to left.

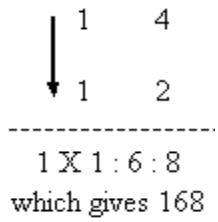
Step i):



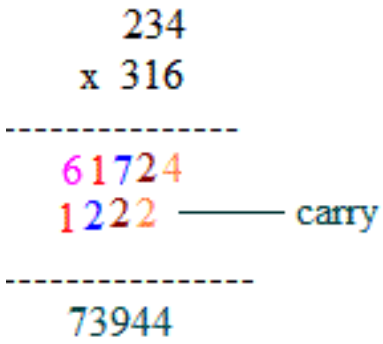
Step ii):



step iii):



3 digit multiplication is given by



Steps:

i) 4 X 6 = 24; 2, the carried over digit is placed below the second digit.

ii) (3 X 6) + (4 x 1) = 18 + 4 = 22; 2, the carried over digit is placed below third digit.

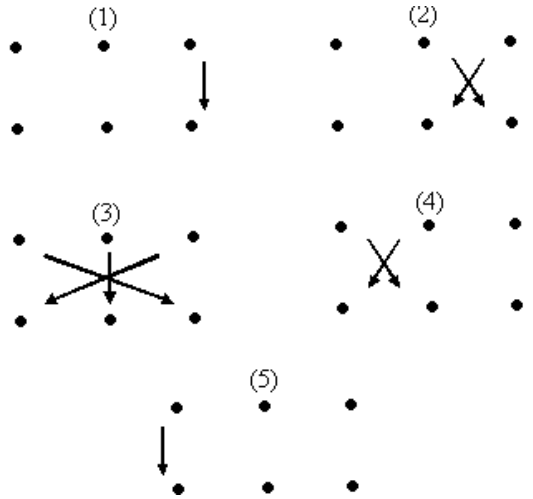
iii) (2 X 6) + (3 X 1) + (4 X 3) = 12 + 3 + 12 = 27; 2, the carried over digit is placed below fourth digit.

iv) (2 X 1) + (3 X 3) = 2 + 9 = 11; 1, the carried over digit is placed below fifth digit.

v) (2 X 3) = 6.

vi) Respective digits are added.

General rule for a 3 digit by 3 digit multiplication:



Sutra 4 - Paravartya Yogayat (Paravartya)

This is the concept of division in Vedic Mathematics when the divisor is closer and slightly greater than power of 10. This sutra requires some specific conditions:

- We apply this method when the Divisor is greater than power of 10 (10, 100, 1000, etc) starts with 1. The Divisor Need NOT to be JUST GREATER than power of 10 since large digits (6, 7, 8, 9) can be converted to smaller digits (1, 2, 3, 4) using Vinculum. Also if Divisor starts with other than 1 then we can apply Anurupyena Sutra.
- As the name (Transpose and Apply) suggests we change the sign and apply the changed number. i.e. Addition will be turned to Subtraction, Division will be changed to Multiplication and vice versa.
- Remainder Theorem and Horner's process of Synthetic Division are small parts of Paravartya Sutra.

1.1 Problem Statement

Arithmetic Logic Unit (ALU) is the main part of the Central Processing unit which performs various arithmetic and logical operations. The speed of arithmetic unit is of extreme importance and depends greatly on the speed of multiplier. Therefore, the technologies are always looking for new algorithm and hardware so as to implement this operation in much optimized way in the terms of area and speed. Vedic Mathematics deals with various branches of mathematics like arithmetic, algebra, geometry etc.

1.2 The Solution

The use of Vedic Mathematics concepts in the computation algorithm of a processor will reduce the complexity of execution time, area and power consumption etc. The efficiency of Urdhav Triyagbhyam Vedic method for multiplication, strikes a difference of actual process of multiplication, by enabling parallel generation of intermediate product, eliminating unwanted multiplication steps with zeros and scaled to higher bit level. This formula (Sutra) can used to build high speed power efficient multiplier in a processor.

Nikhilam division algorithm just involves the addition of numbers which is very much different from the traditional division technique including multiplication of big numbers by the trial digit of the quotient at each step and subtracts that result from dividend at each step.

This project aims to design arithmetic and logic unit using the technique of ancient Indian Vedic Mathematics to improve the performance of a processor.

1.3 Objectives of the Project

The objectives of the Project work are –

i) To design an 8-bit Arithmetic and Logic Unit (ALU) based on Vedic Mathematics.

ii) To implement the Vedic Mathematics based ALU on Spartan 6 Field Programmable Gate Array (FPGA) device.

1.4 Organization of Dissertation

Chapter 1 includes a brief introduction of the area of the project work, objectives, problem statement and literature review of the related work in the area of the project work.

Chapter 2 describes the system requirement specifications for the project work. It includes hardware and software requirements for the project work.

Chapter 3 describes the system design flow of the project. It includes the project description, block diagram and design flow flow-chart of the project work.

Chapter 4 includes the hardware description of the project work. It describes the features of the Spartan 6 FPGA development board used for the project work.

Chapter 5 describes the digital design of the algorithms of the vedic concepts for multiplication and division.

Chapter 6 includes various testing results of the project work.

Chapter 7 includes the hardware domain results of various operations performed by the ALU.

Chapter 8 includes the conclusion, the scope for future work, advantages and limitations of the project work.

1.5 Literature Survey

Vedic Mathematics concepts greatly simplify many arithmetic computations which would result in the implementation of efficient algorithms for the computational units. Many researchers have proposed ALUs and other computational units implemented based on the concepts of ancient Indian Vedic Mathematics concepts for different signal processing applications. These designs have proved to be robust compared to conventional arithmetic computational algorithms.

Garima Rawat et al. [1] have proposed an ALU design using Vedic Mathematics approach. A high-speed 8×8 bit multiplier is designed and analyzed which is based on the Vedic multiplier mechanism. This architecture is diverse from the conservative method of employing product of two numbers accomplished by the process of add and shift. The proposed method is efficient and fast, wherein the processing involves the vertical and crossed multiplication of precedent Vedic mathematics.

Rahul Nimje et al. [2] have proposed an ALU design using Vedic Mathematics approach. As the ever increasing demand in enhancing the ability of coprocessor to handle the complex and challenging processor as resulted in integration of number of processor cores into single chip, but still the load on the processor is not less in generic system. This load is reduced by connecting the main processor with co processor, which are designed to work on the specific types of function like numeric computation, signal processing, image processing and arithmetic operation. The efficiency of Urdhav Triyagbhyam Vedic method for multiplication, strikes a difference of actual process of multiplication, by enabling parallel generation of intermediate product, eliminating

unwanted multiplication steps with zeros and scaled to higher bit level. This formula (Sutras) is used to build high speed power efficient multiplier in coprocessor.

Abhishek Gupta et al. [3] have proposed an ALU design using Vedic Mathematics approach. Every digital domain based technology depends upon the operations performed by ALU either partially or whole. That's why it highly required designing high speed ALU, which can enhance the efficiency of those modules which lies upon the operations performed by ALU. The proposed ALU is able to perform three different arithmetic and eight different logical operations at high speed.

Suchita Kamble et al. [4] have proposed VHDL implementation of 8-bit arithmetic logic unit (ALU). ALU consist of two input registers to hold the data during operation, one output register to hold the result of operation, 8-bit fast adder with 2's complement circuit to perform subtraction and logic gates to perform logical operation. The maximum propagation delay is 13.588 ns and power dissipation is 38 mW. The ALU was designed for controller used in network interface card.

S.P.Pohokar et al. [5] have proposed VHDL implementation of 8-bit arithmetic logic unit (ALU). Vedic technique eliminates the unwanted multiplication steps thus reducing the propagation delay in processor and hence reducing the hardware complexity in terms of area and memory requirement.

Surabhi Jain et al. [6] have proposed that VLSI architecture have higher orders of time and space complexities. Vedic Mathematics on the other hand offers a new holistic approach to mathematics. In this work, optimized binary division architecture has been designed using sutras of Vedic Mathematics which are Nikhilam Sutra and Parvartya Sutra. This work discusses about these two algorithms of division and their application for calculating deconvolution.

Abhyarthana Bisoyil et al. [7] have proposed that Binary multipliers and addresses are used in the design and development of Arithmetic Logic Unit (ALU). The objective of this paper is to implement digital multipliers based on the concept of Vedic mathematics. In order to develop a digital multiplier, Urdhva-tiryakbyham sutra of Vedic mathematics is used to implement vertical and cross wise operations. Since these are digital multipliers, they are implemented on FPGA board.

Chapter 2

System Requirements Specifications

2.1 Software Requirements

The software requirements of the project are

- Operating System: Windows 7
- EDA/CAD Tool: Xilinx ISE 14.1

2.1.1 Windows 7 Operating System

Windows 7 (codenamed Vienna, formerly Blackcomb) is a personal computer operating system developed by Microsoft. It is a part of the Windows NT family of operating systems. Windows 7 was released to manufacturing on July 22, 2009, and became generally available on October 22, 2009, less than three years after the release of its predecessor, Windows Vista. Windows 7's server counterpart, Windows Server 2008 R2, was released at the same time.

Windows 7 was primarily intended to be an incremental upgrade to the operating system intending to address Windows Vista's poor critical reception while maintaining hardware and software compatibility. Windows 7 continued improvements on Windows Aero (the user interface introduced in Windows Vista) with the addition of a redesigned taskbar that allows applications to be "pinned" to it, and new window management features. Other new features were added to the operating system, including libraries, the new file sharing system HomeGroup, and support for multi touch input. A new "Action Center" interface was also added to provide an overview of system security and maintenance information, and tweaks were made to the User Account Control system to make it less intrusive. Windows 7 also shipped with updated versions of several stock applications, including Internet Explorer 8, Windows Media Player, and Windows Media Center.

Table 2.1 Operational System Hardware Requirements

Component	Operating system architecture	
	32-bit	**64-bit**
Processor	1 GHz IA-32 processor	1 GHz x86-64 processor
Memory (RAM)	1 GB	2 GB
Graphics card	DirectX 9 graphics processor with WDDM driver model 1.0	
Free hard drive space	16 GB	20 GB
Optical drive	DVD-ROM drive (Only to install from DVD-ROM media)	

2.1.2 Xilinx ISE 14.1 EDA/CAD Tool

Xilinx ISE (Integrated Synthesis Environment) is a software tool produced by Xilinx for synthesis and analysis of HDL designs, enabling the developer to synthesize ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer.

Xilinx ISE is a design environment for FPGA products from Xilinx, and is tightly-coupled to the architecture of such chips, and cannot be used with FPGA products from other vendors. The Xilinx ISE is primarily used for circuit synthesis and design, while ISIM or the Model Sim logic simulator is used for system-level testing. Other components shipped with the Xilinx ISE include the Embedded Development Kit (EDK), a Software Development Kit (SDK) and Chip Scope Pro.

Since 2012, Xilinx ISE has been discontinued in favor of Vivado Design Suite, that serves the same roles as ISE with additional features for system on a chip development. Xilinx released the last version of ISE in October 2013 (version 14.7), and states that "ISE has moved into the sustaining phase of its product life cycle, and there are no more planned ISE releases".

User Interface: The primary user interface of the ISE is the Project Navigator, which includes the design hierarchy (Sources), a source code editor (Workplace), an output console (Transcript), and a processes tree (Processes).

The Design hierarchy consists of design files (modules), whose dependencies are interpreted by the ISE and displayed as a tree structure. For single-chip designs there may

be one main module, with other modules included by the main module, similar to the main() subroutine in C++ programs. Design constraints are specified in modules, which include pin configuration and mapping.

The Processes hierarchy describes the operations that the ISE will perform on the currently active module. The hierarchy includes compilation functions, their dependency functions, and other utilities. The window also denotes issues or errors that arise with each function. The Transcript window provides status of currently running operations, and informs engineers on design issues. Such issues may be filtered to show Warnings, Errors, or both.

Simulation: System-level testing may be performed with ISIM or the Model Sim logic simulator, and such test programs must also be written in HDL languages. Test bench programs may include simulated input signal waveforms, or monitors which observe and verify the outputs of the device under test.

ModelSim or ISIM may be used to perform the following types of simulations:

- Logical verification, to ensure the module produces expected results
- Behavioural verification, to verify logical and timing issues
- Post-place & route simulation, to verify behaviour after placement of the module within the reconfigurable logic of the FPGA

Synthesis: The primary user interface of the ISE is the Project Navigator, which includes the design hierarchy (Sources), a source code editor (Workplace), an output console (Transcript), and a processes tree (Processes).

The Design hierarchy consists of design files (modules), whose dependencies are interpreted by the ISE and displayed as a tree structure. For single-chip designs there may be one main module, with other modules included by the main module, similar to the main() subroutine in C++ programs. Design constraints are specified in modules, which include pin configuration and mapping.

The Processes hierarchy describes the operations that the ISE will perform on the currently active module. The hierarchy includes compilation functions, their dependency

functions, and other utilities. The window also denotes issues or errors that arise with each function.

The Transcript window provides status of currently running operations, and informs engineers on design issues. Such issues may be filtered to show Warnings, Errors, or both.

Xilinx's patented algorithms for synthesis allow designs to run up to 30% faster than competing programs, and allows greater logic density which reduces project time and costs.

Also, due to the increasing complexity of FPGA fabric, including memory blocks and I/O blocks, more complex synthesis algorithms were developed that separate unrelated modules into slices, reducing post-placement errors.

IP Cores are offered by Xilinx and other third-party vendors, to implement system-level functions such as digital signal processing (DSP), bus interfaces, networking protocols, image processing, embedded processors, and peripherals. Xilinx has been instrumental in shifting designs from ASIC-based implementation to FPGA-based implementation.

Editions: The Subscription Edition is the licensed version of Xilinx ISE, and a free trial version is available for download.

The Web Edition is the free version of Xilinx ISE that can be downloaded and used for no charge. It provides synthesis and programming for a limited number of Xilinx devices. In particular, devices with a large number of I/O pins and large gate matrices are disabled.

The low-cost Spartan family of FPGAs is fully supported by this edition, as well as the family of CPLDs, meaning small developers and educational institutions have no overheads from the cost of development software.

License registration is required to use the Web Edition of Xilinx ISE, which is free and can be renewed an unlimited number of times.

2.2 Hardware Requirements

The hardware requirements of the project are

- Intel Pentium IV Processor (minimum requirement), 1 GB RAM
- Xilinx Spartan-6 FPGA Development Board (XC6SLX9 CSG324 Device)

2.2.1 Intel Pentium IV Processor (minimum requirement), 1 GB RAM

Pentium 4 was a line of single-core desktop, laptop and entry level server central processing units (CPUs) introduced by Intel on November 20, 2000 and shipped through August 8, 2008. They had a seventh-generation x86 (32-bit) microarchitecture, called NetBurst, which was the company's first all-new design since the introduction of the P6 microarchitecture of the Pentium Pro CPUs in 1995. Net Burst differed from P6 (Pentium III, II, etc.) by featuring a very deep instruction pipeline to achieve very high clock speeds. Intel claimed that Net Burst would allow clock speeds of up to 10 GHz in future chips; however, severe problems with heat dissipation (especially with the Prescott Pentium 4) limited CPU clock speeds to a much lower 3.8 GHz. In 2004, the initial 32-bit x86 instruction set of the Pentium 4 microprocessors was extended by the 64-bit x86-64 set.The first Pentium 4 cores, codenamed Willamette, were clocked from 1.3 GHz to 2 GHz.

They were released on November 20, 2000, using the Socket 423 system. Notable with the introduction of the Pentium 4 was the 400 MT/s FSB. It actually operated at 100 MHz but the FSB was quad-pumped, meaning that the maximum transfer rate was four times the base clock of the bus, so it was marketed to run at 400 MHz. The AMD Athlon's double-pumped FSB was running at 100 or 133 MHz (200 or 266 MT/s) at that time.

Pentium 4 CPUs introduced the SSE2 and, in the Prescott-based Pentium 4s, SSE3 instruction sets to accelerate calculations, transactions, media processing, 3D graphics, and games. Later versions featured Hyper-Threading Technology (HTT), a feature to make one physical CPU work as two logical CPUs. Intel also marketed a version of their low-end Celeron processors based on the NetBurst microarchitecture (often referred to as Celeron 4), and a high-end derivative, Xeon, intended

for multiprocessor servers and workstations. In 2005, the Pentium 4 was complemented by the Pentium D and Pentium Extreme Edition dual-core CPUs.

2.1.2 Spartan-6 FPGA Development Board (XC6SLX9 CSG324 Device)

Spartan-6 devices are the most cost-optimized FPGAs, offering industry leading connectivity features such as high logic-to-pin ratios, small form-factor packaging, and a diverse number of supported I/O protocols. Built on 45nm technology, the devices are ideally suited for a range of advanced bridging applications found in automotive infotainment, consumer, and industrial automation.

Applications

- High Resolution Video and Graphics
- Industrial Ethernet Switch
- Full-HD Intelligent Digital Signage

Table 2.2 Xilinx Spartan-6 FPGA Features

Value	Features
Programmable System Integration	• Highest pin-count to logic ratio for I/O connectivity • Over 40 I/O standards for simplified system design • PCI Express with integrated endpoint block
Increased System Performance	• Up to 8 low power 3.2Gb/s serial transceivers • 800Mb/s DDR3 with integrated memory controller
BOM Cost Reduction	• Cost-optimized for system I/O expansion • MicroBlaze soft IP to eliminate processor or MCU components

Total Power Reduction	• 1.2V core voltage or 1.0V core voltage option • Zero power with hibernate power-down mode
Accelerated Design Productivity	• Enabled by ISE Design Suite—a no-cost, front-to-back FPGA design solution for Linux and Windows • Fast design closure using integrated wizards

Chapter 3

System Design

3.1 Project Description

In any processor, major hardware units are control unit, ALU and memory read/write. ALU is an execution unit which not only performs the arithmetic operation, but also logical operation and therefore ALU is called a heart of Microprocessor, Microcontroller, and CPUs. In every technology, the operations are either fully or partially performed by ALU.

Vedic mathematics is used to solve the complex calculations involved in usual mathematics. The Vedic multiplier is based on the algorithm named as "Urdhva Tiryakbhyam" sutra. The algorithm is viable for multiplication of any two numbers exhibiting bit length equal to n. Besides this, it introduces a parallel execution of partial products and sums. Similarly, the division based on "Nikhilam Sutra" of vedic mathematics has also been proved advantageous, despite of having many limitations. The digital design of arithmetic operations based on these vedic mathematics techniques has proven advantageous in combating with large delays and complexity of conventional multipliers [1-6].

In this project work, the design of ALU based on vedic mathematics principles is carried out. The ASIC digital design flow is employed for the design of the required ALU. The algorithms for various arithmetic operations are formed based on the vedic principles. The Verilog HDL is used to map these algorithms to the hardware domain using Xilinx 14.1 EDA (Electronic Design Automation) tool. The Isim simulator is used for the verification of the designed vedic mathematics based ALU. The verified design is implemented using Xilinx Spartan – 6 FPGA chip.

3.2 Block Diagram of the Proposed Vedic Mathematics Based ALU

Fig. 3.1 shows the block diagram of the proposed Vedic Mathematics based ALU design. The latches are used to store the two binary inputs and output result of an arithmetic or

logic operation. The multiplexer is used to route the result of a selected operation to the output latches. The ALU performs arithmetic and logical operations. The ALU has two sub-units: an arithmetic unit based on Vedic Mathematics to perform arithmetic operations and a logic unit to perform the logical operations.

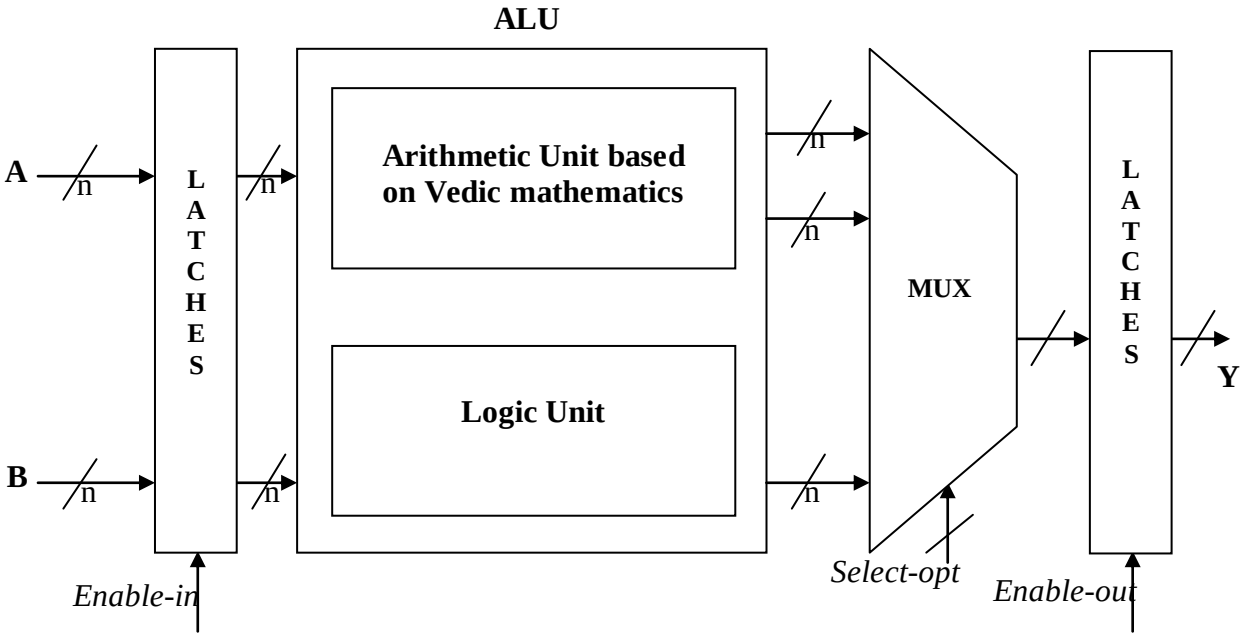


Fig. 3.1 Block diagram of the proposed vedic mathematics based ALU

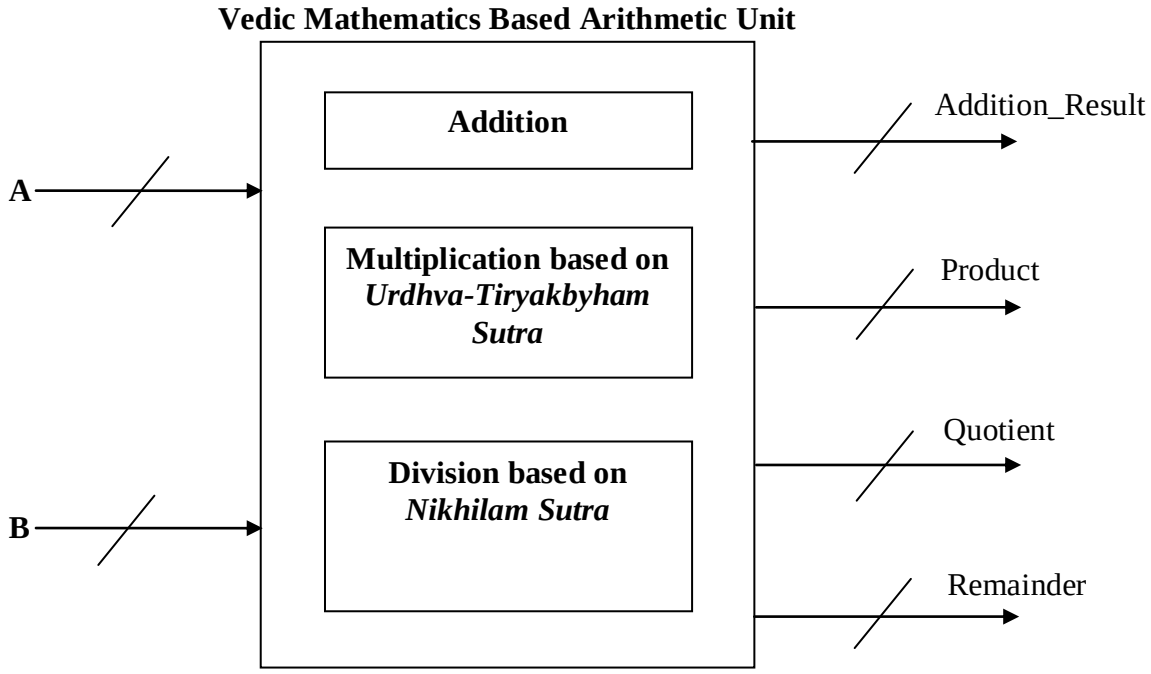


Fig. 3.2 Block diagram of arithmetic unit of the ALU

Fig 3.2 shows the block diagram of the Vedic Mathematics based arithmetic unit of the ALU. The addition is carried out using robust adders. The multiplication is carried out using the Urdhva-Tiryakbyham Sutra. The Urdhva-Tiryakbyham Sutra is used to construct 2x2 multiplier blocks. Several such 2x2 multipliers along with ripple carry adders are used to design 4x4 multiplier blocks. This procedure is repeated to obtain the higher order multiplier blocks. The binary division is carried out using Nikhilam Sutra.

3.3. ALU Design Flow

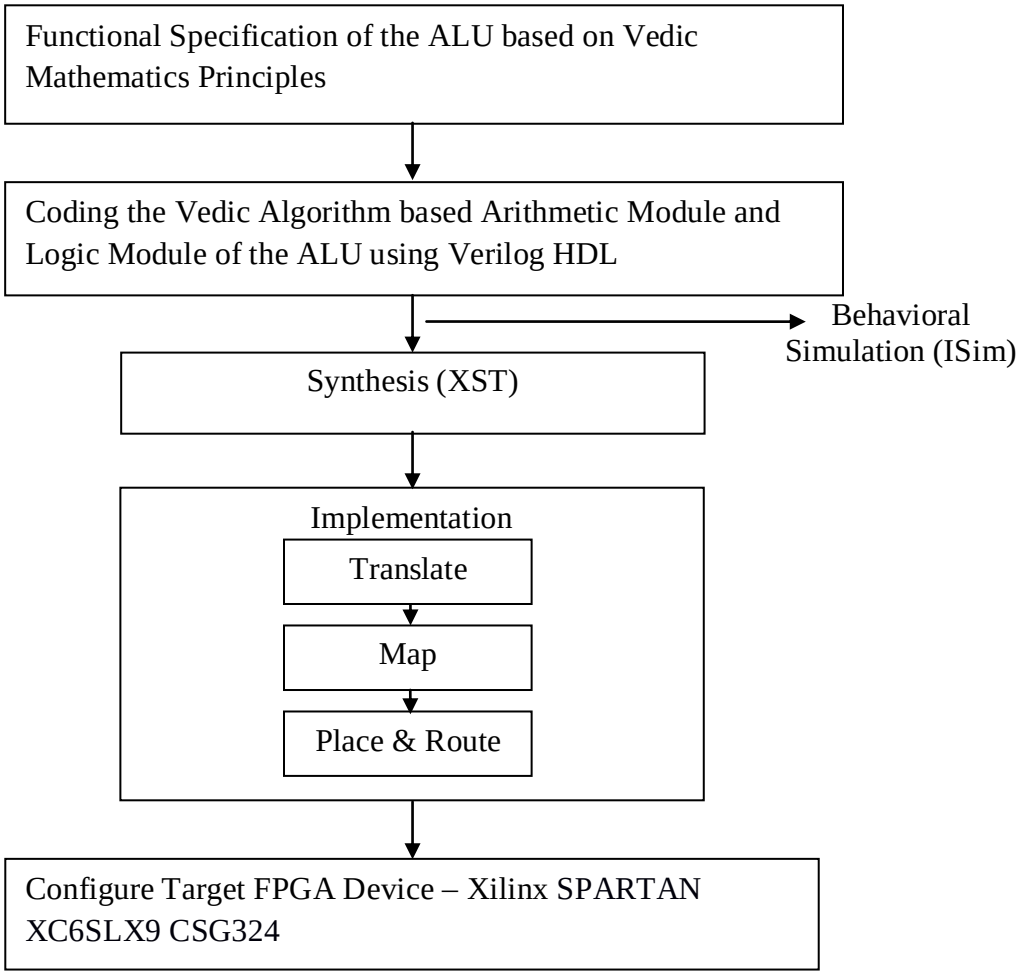


Fig. 3.3 FPGA design flow of the ALU

Fig 3.4.1 shows the FPGA design flow of the ALU. The proposed Vedic Mathematics based ALU is designed in high level hardware description language – Verilog using the EDA/CAD tool Xilinx ISE 14.1. The design is verified by behavioral simulation using the built-in simulator ISim. The verified design is then synthesized using XST synthesis tool

of the Xilinx ISE. The synthesized design is mapped to target FPGA device and programming file is generated. Finally, the generated programming file is downloaded on to Spartan 6 FPGA device for functional verification.

Chapter 4

Hardware Description

4.1 FPGA (Field Programmable Gate Array)

An FPGA is a device that contains a matrix of reconfigurable gate array logic circuitry. When a FPGA is configured, the internal circuitry is connected in a way that creates a hardware implementation of the software application. Unlike processors, FPGAs use dedicated hardware for processing logic and do not have an operating system. FPGAs are truly parallel in nature so different processing operations do not have to compete for the same resources. As a result, the performance of one part of the application is not affected when additional processing is added. Also, multiple control loops can run on a single FPGA device at different rates. FPGA-based control systems can enforce critical interlock logic and can be designed to prevent I/O forcing by an operator. However, unlike hard-wired printed circuit board (PCB) designs which have fixed hardware resources, FPGA-based systems can literally rewire their internal circuitry to allow reconfiguration after the control system is deployed to the field. FPGA devices deliver the performance and reliability of dedicated hardware circuitry. A single FPGA can replace thousands of discrete components by incorporating millions of logic gates in a single integrated circuit (IC) chip.

4.1.1 Advantages of FPGA

1. **Long time availability:** FPGAs (Field Programmable Gate Arrays) enable you to make yourself independent from component manufacturers and distributors since the functionality is not given by the device itself but in its configuration. The configuration can be programmed to be portable between miscellaneous FPGAs without any adaptations.
2. **Can be updated and upgraded at your customer's site:** FPGAs in contrast to traditional computer chips are completely configurable. Updates and feature enhancement can be carried out even after delivery at your customer's site.
3. **Extremely short time to market:** Through the use of FPGAs the development of hardware prototypes is significantly accelerated since a big part of the hardware

development process is shifted into ip core design, which can take place in parallel. Additionally, because of the early availability of hardware prototypes, time-consuming activities like the start-up and debugging of the hardware are brought forward concurrently to the overall development.

4. **Fast and efficient systems:** Available standard components address a broad user group and consequently often constitute a compromise between performance and compatibility. With FPGAs, systems can be developed that are exactly customized for the designated task and for this reason works highly efficient.
5. **Performance gain for software applications:** Complex tasks are often handled through software implementations in combination with high-performance processors. In this case FPGAs provide a competitive alternative, which by means of parallelization and customization for the specific task even establishes an additional performance gain.
6. **Real time applications:** FPGAs are perfectly suitable for applications in time-critical systems. In contrast to software based solutions with real time operating systems, FPGAs provide real deterministic behavior. By means of the featured flexibility even complex computations can be executed in extremely short periods.
7. **Massively parallel data processing:** The amount of data in contemporary systems is ever increasing which leads to the problem that systems working sequential are no longer able to process the data on time. Especially by means of parallelization, FPGAs provide a solution to this problem which in addition scales excellently.

4.2 Spartan 6 FPGA Development Board

Spartan-6 FPGA is specially designed for experimenting and learning system design with FPGAs. This development board features SPARTAN XC6SLX9 CSG324 FPGA with on-board 512Mb DDR SDRAM. The USB 2.0 interface provides fast and easy configuration download to the on-board SPI flash. Fig. 4.1 shows Spartan 6 FPGA Development Board. Fig. 4.2 shows the connection diagram of Spartan 6 development board.

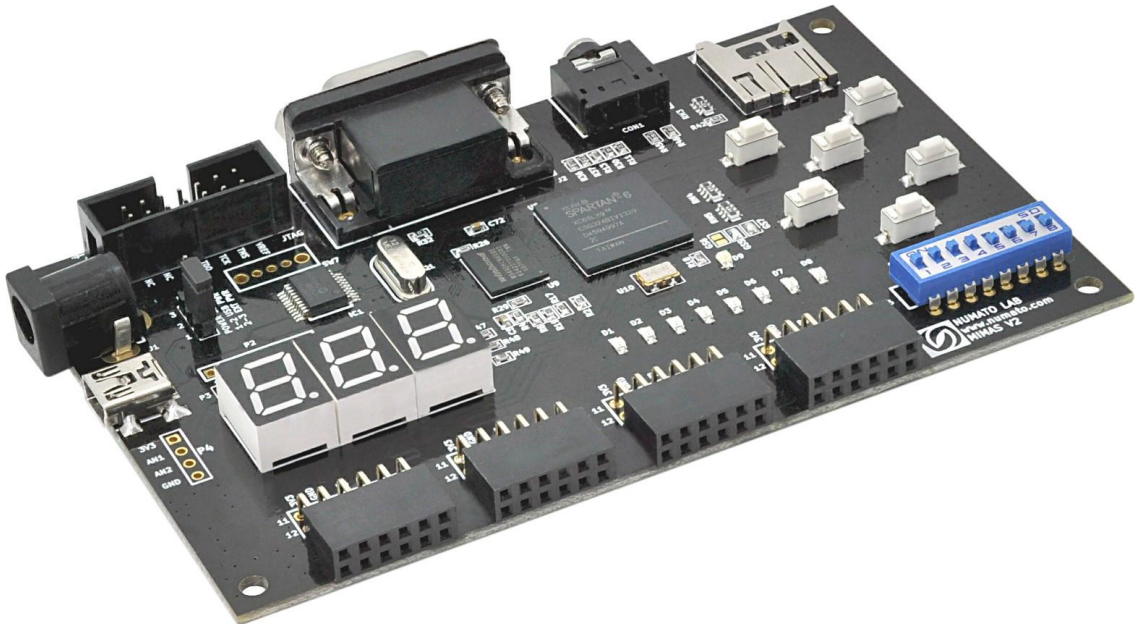


Fig. 4.1 Spartan 6 FPGA development board

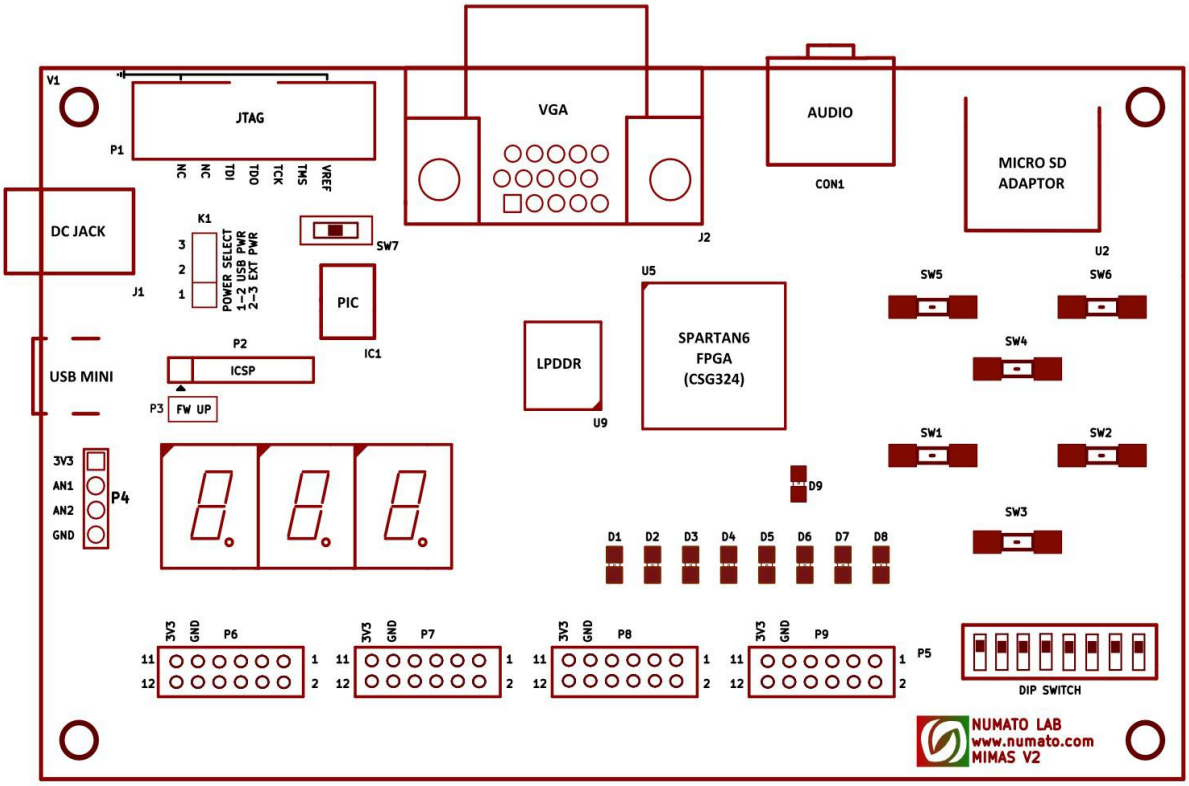


Fig. 4.2 Spartan 6 FPGA board connection diagram

4.2.1 Board Features

- FPGA: Spartan XC6SLX9 in CSG324 package
- DDR: 166MHz 512Mb LPDDR (MT46H32M16LF/W949D6CBHX6E)
- Flash memory: 16 Mb SPI flash memory (M25P16)
- USB 2.0 interface for On-board flash programming
- FPGA configuration via JTAG and USB
- 8 LEDs Six Push Buttons and 8 way DIP switch for user defined purposes
- VGA Connector
- Stereo Jack
- Micro SD Card Adapter
- Three Digit Seven Segment Display
- 32 IOs for user defined purposes
- Four 6×2 Expansion Connectors
- On-board voltage regulators for single power rail operation

4.2.2 USB Interface

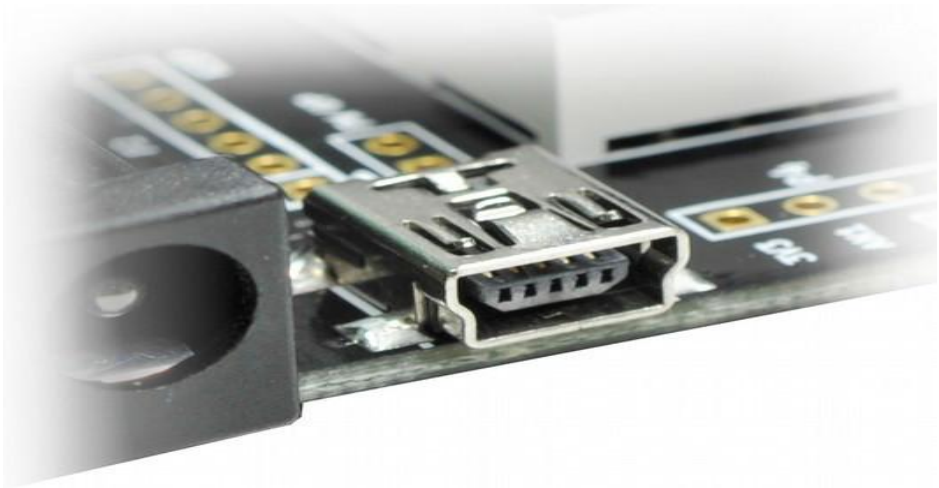


Fig. 4.3 On-board USB Interface Slot

Fig. 4.3 shows the on-board USB interface slot. The on-board full speed USB controller helps a computer to communicate with this module, using a USB A to Mini B cable to

connect with a PC. By default the module is powered from USB so make sure not to overcrowd unpowered USB hubs.

4.2.3 Segment LED Display

This board features three 7-segment LED display multiplexed for low pin count operation. Each module can be separately turned on and off with the three switching transistors.

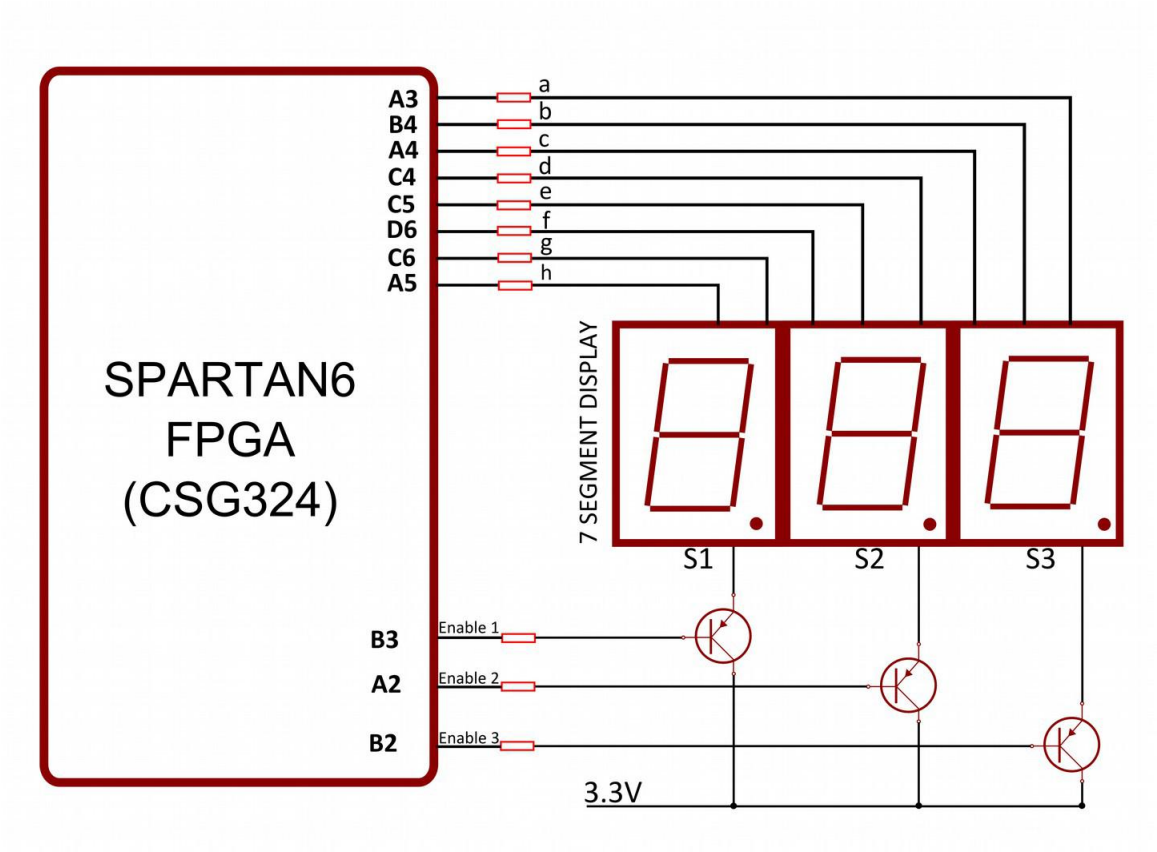


Fig. 4.4 Seven segment display

4.2.4 DC Power Supply

This module uses +5V power supply to function properly. By default the board is configured to use +5V supply from USB. So an external +5V power is not required unless USB port is unable to supply enough current. In most cases USB ports are capable of providing enough current for the module. Current requirement for this board largely depends on your application. If for any reason, an external 5V power supply needs to be

used for the module, the Power select jumper should be configured properly before connecting the power supply. Please refer to the marking on the board for more details.

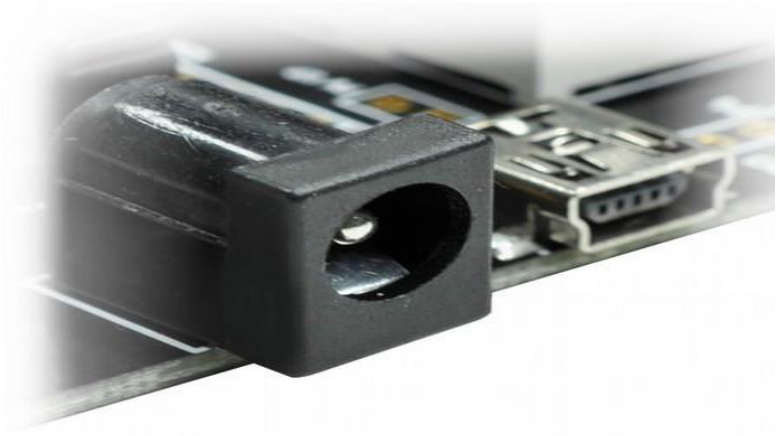


Fig. 4.5 DC power supply

4.2.5 Power Select

The Power Select header K1 is used to configure the power source for the board. The jumper in pin 1 and 2 is shorted to switch the power source to on board USB port and pin 2 and 3 to use the external DC power.

4.2.6 JTAG Connector

JTAG connector provides access to FPGA's JTAG pins. A XILINX platform cable can be used for JTAG programming.

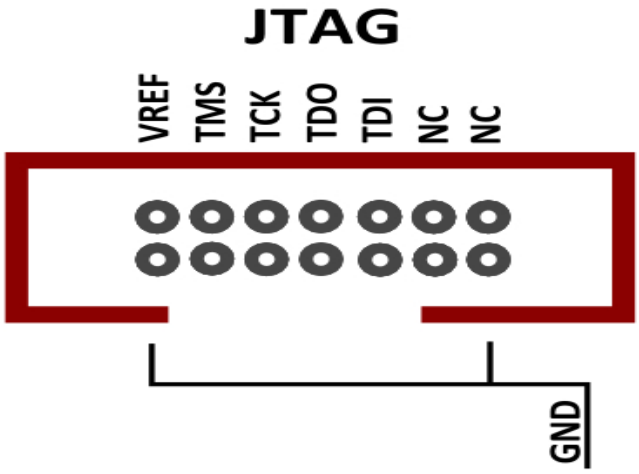


Fig. 4.6 JTAG connector

4.2.7 Configuration Mode Selection

Slide switch SW7 is used to switch between the USB configuration mode and UART. Slide the switch to Position 1 to download bit stream through USB configuration tool and Position 2 to use the interface as a UART in order to communicate from your code in FPGA with the PC. By default the board is shipped with slide switch position in USB configuration tool mode.

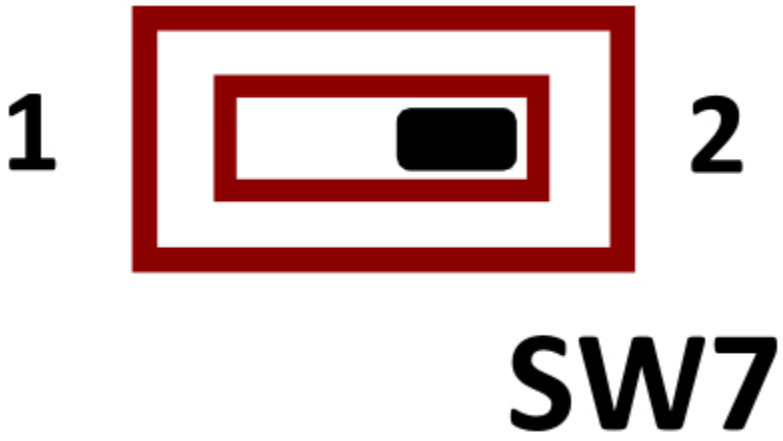


Fig. 4.7 Configuration Mode Selection

4.2.8 UART

The MIMAS V2 includes USB-UART, which helps to establish the communication between the code in the FPGA and any application running on the PC. Data can be sent and received from the FPGA by using Serial Terminal at baud rate 19200.

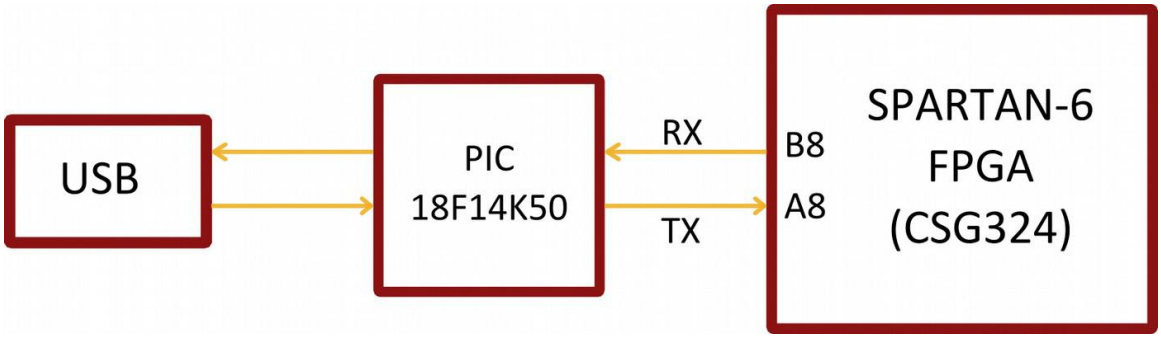


Fig. 4.8 UART

4.2.9 LED, Push Button and Dip Switch

MIMAS V2 has six push button switches, an eight position DIP switch and eight LEDs for human interaction. All switches are directly connected to Spartan 6 FPGA and can be used in your design with minimal effort.

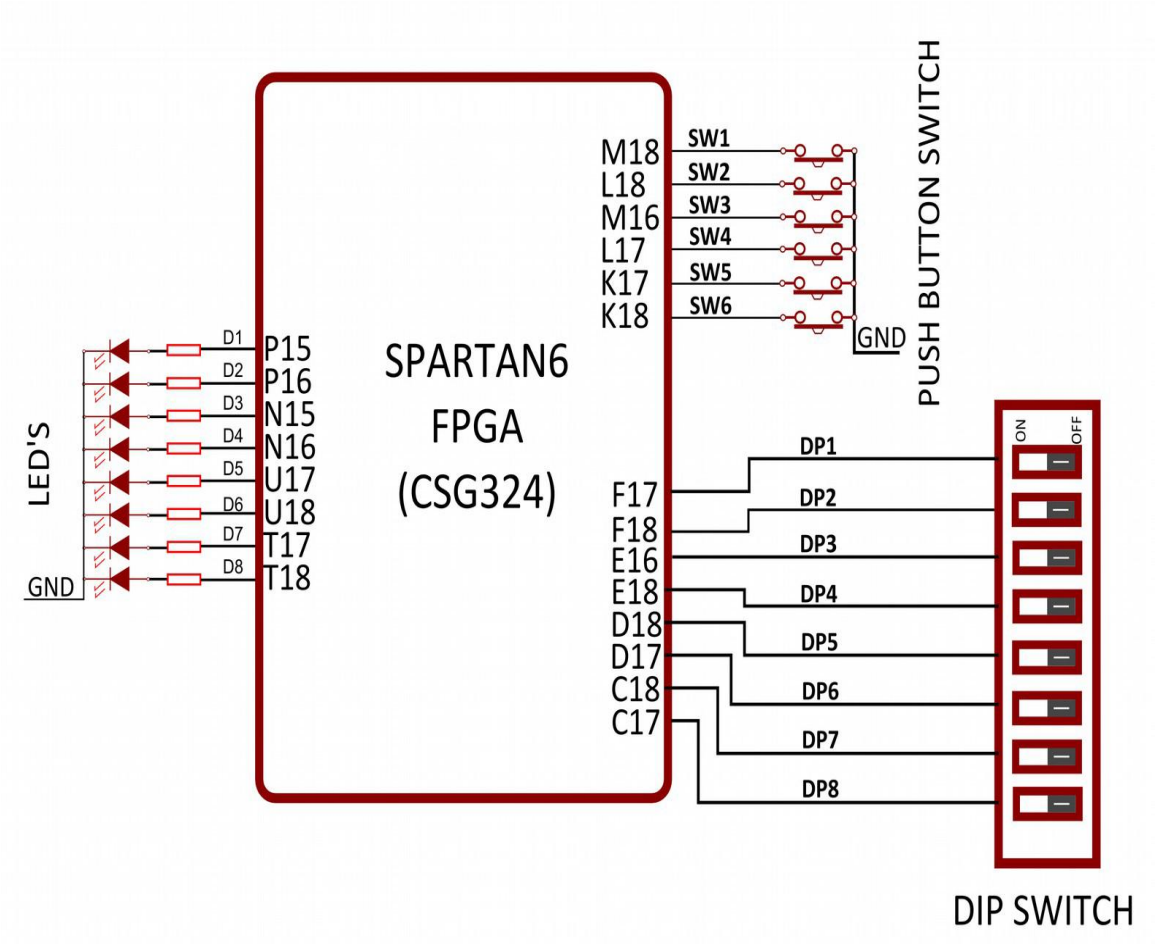


Fig. 4.9 Connection diagram of LEDs, push buttons and DIP switches

4.2.10 Micro SD

MIMAS V2 features a Micro SD adapter on-board. By installing a Micro SD card, you can add data logging, media storage and other file storage to your design.

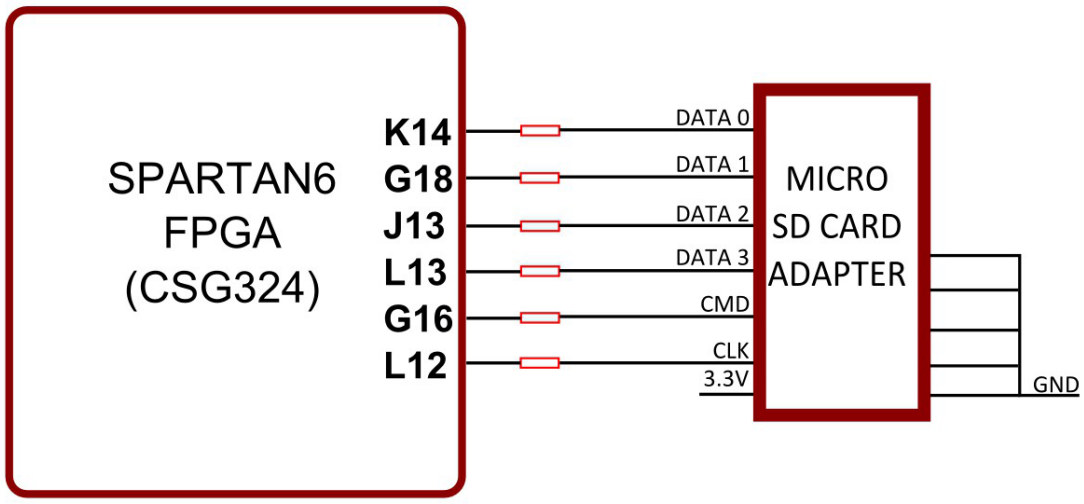


Fig. 4.10 Micro SD

4.2.11 Technical Specifications of the FPGA Development Board

Table 4.1 Technical specifications of FPGA development board

Parameter *	Value	Unit
Basic Specifications		
Number of GPIOs	32	
Number of LEDs	8	
Number of Push Buttons	6	
SPI Flash Memory (M25P16)	16	Mb
Power supply voltage (USB or external)	5 - 7	V
FPGA Specifications		
Internal supply voltage relative to GND	–0.5 to 1.25	V
Auxiliary supply voltage relative to GND	–0.5 to 3.75	V
Output drivers supply voltage relative to GND	–0.5 to 3.75	V

Chapter 5

Digital Design of Vedic Algorithms

The 8-bit vedic mathematics based ALU has the following modules:

1. Adder (8 - bit) – Designed using ripple carry addition scheme
2. Subtractor (8 - bit) – Designed using ripple carry scheme
3. Multiplier (8 - bit) – Designed based on vedic mathematics principles
4. Divider (8 - bit) – Designed based on vedic mathematics principles
5. Logic unit (8 - bit)

The design of multiplier and divider is described in this chapter. The vedic algorithms for multiplier and divider are implemented with the help of Verilog HDL coding.

5.1 Multiplication Based on Vedic Mathematics

5.1.1 Design of 2x2 Vedic Multiplier

\

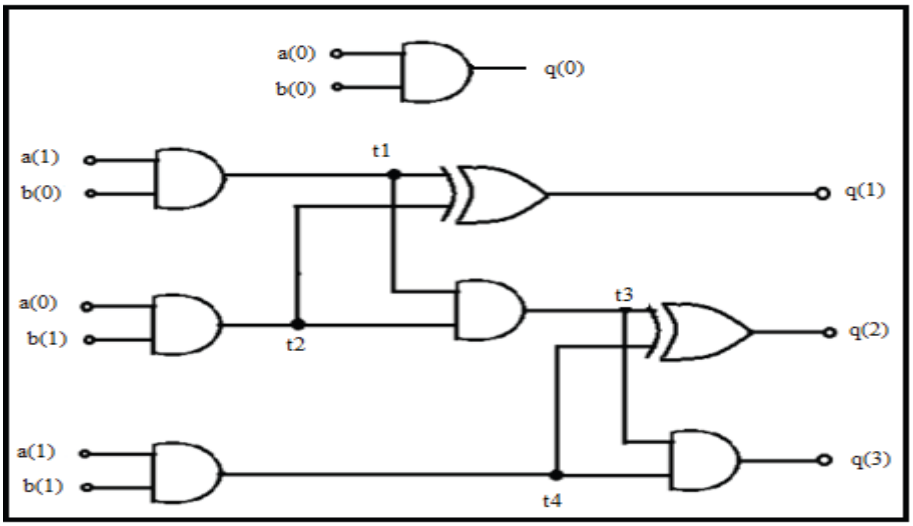


Fig.5.1 Implementation of 2 x 2 multiplier block

The design of 2 x 2 Vedic multiplier is shown in Fig.5.1. This is used as a basic building block for design of 4 x 4 bit Vedic multiplier. This is done by adding partial products

using carry save adders. Then by using 4 x 4 bit Vedic multiplier as a building block, 8 x 8 bit Vedic multiplier has been designed.

5.1.2 Design of 4x4 Vedic Multiplier

The 4×4 bit Vedic multiplication unit is further realized by incorporating four similar modules of 2×2 multipliers. The processing in the form of block diagram is depicted in Fig.5.2 for 4×4 multiplier.

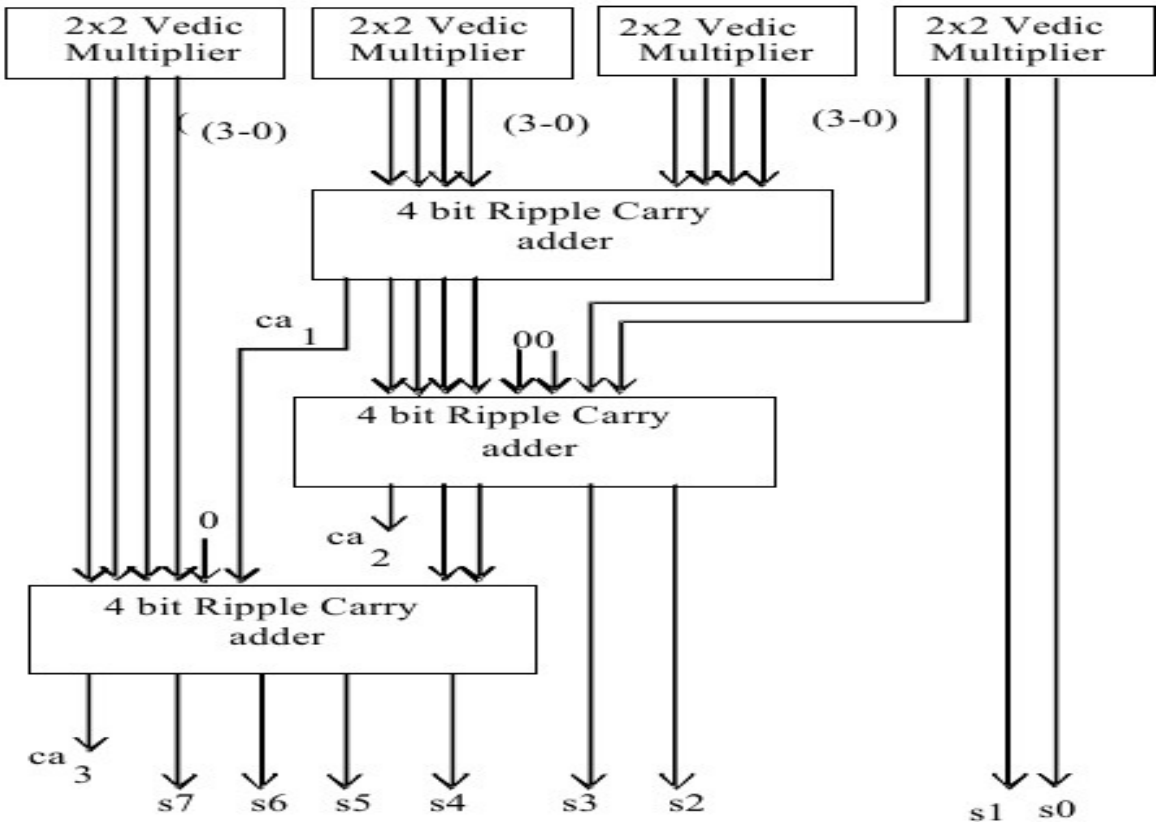


Fig. 5.2 Implementation of 4x4 bit Vedic mathematics based multiplier

5.1.3 Design of 8x8 Vedic Multiplier

The 8×8 Vedic multiplier module is realized using four 4×4 multiplier modules. The processing of 8×8 multiplier based on Vedic methodology is depicted in Fig. 5.3. The process of multiplication is accomplished using four bits simultaneously feeding to the 4-bit multiplier unit and its output is added further to obtain the end result of 2 eight bit numbers.

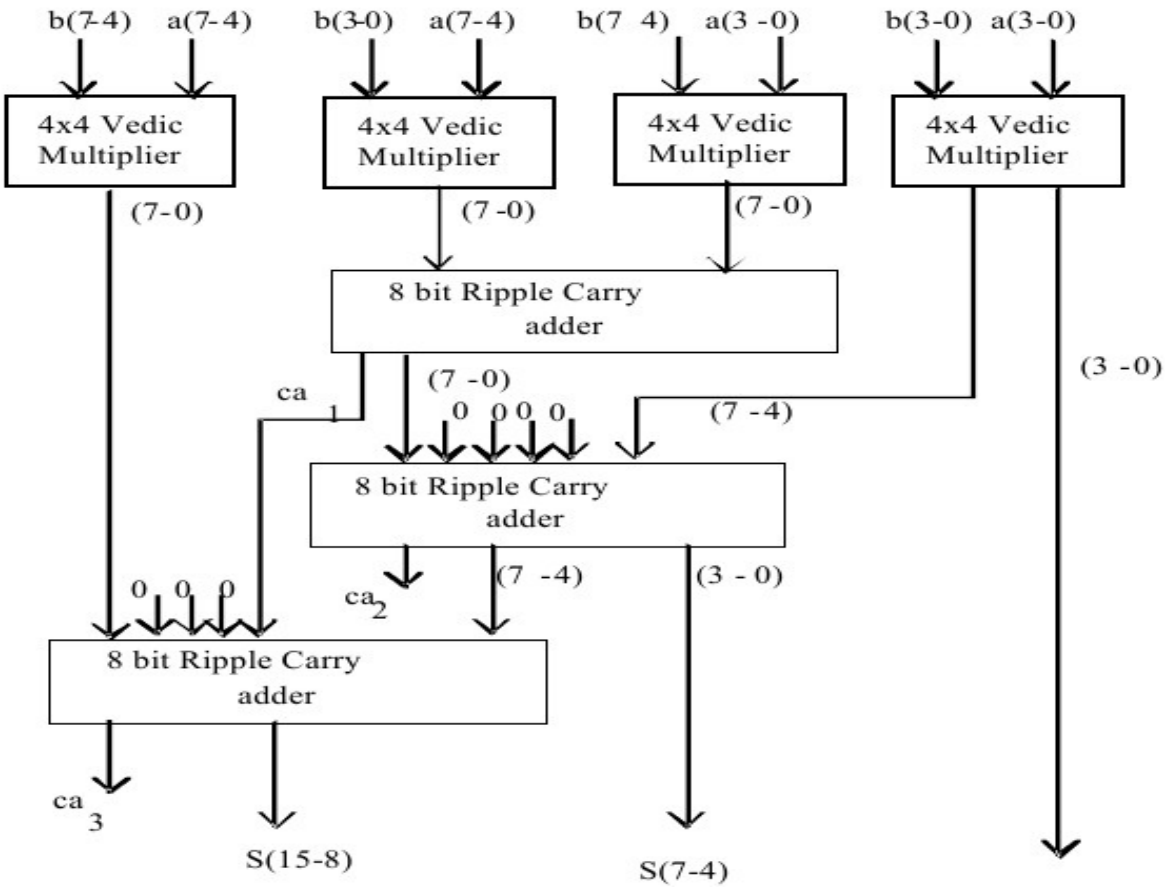


Fig. 5.3 Implementation of 8x8 bit Vedic mathematics based multiplier

5.2 Division Based on Vedic Mathematics

Flow chart diagram of the proposed divider using Nikhilam formula for 8 bit by 4 bit which has been adopted from ancient Vedic Mathematics has been shown in Fig. 5.4.

The architecture for division using 'Nikhilam' Sutra consists of four major sub-segments:-

(i) Complement circuitry

(ii) Adder

(iii) Incrementer

(iv) Multiplier

Assume that, A and B are dividend and divisor respectively. The 'n' bit input from divisor is fed to the complement circuitry. Complement methodology that has been used here, is the two's complement method. The result of complement is fed to the multiplier with 4 MSB of dividend and same 4 MSB of dividend is fed to the incrementer which is

initialized by ’0’. The result of multiplier is fed to the adder with 4 LSB of dividend. If the result of the adder is greater than divisor then the output from the adder is again fed to the multiplier as a new dividend. The operation is repeated again until the result of the incrementer is n/2 bit or n/2+1 bit. The output from the adder is the actual remainder and the result of the incrementer is the quotient.

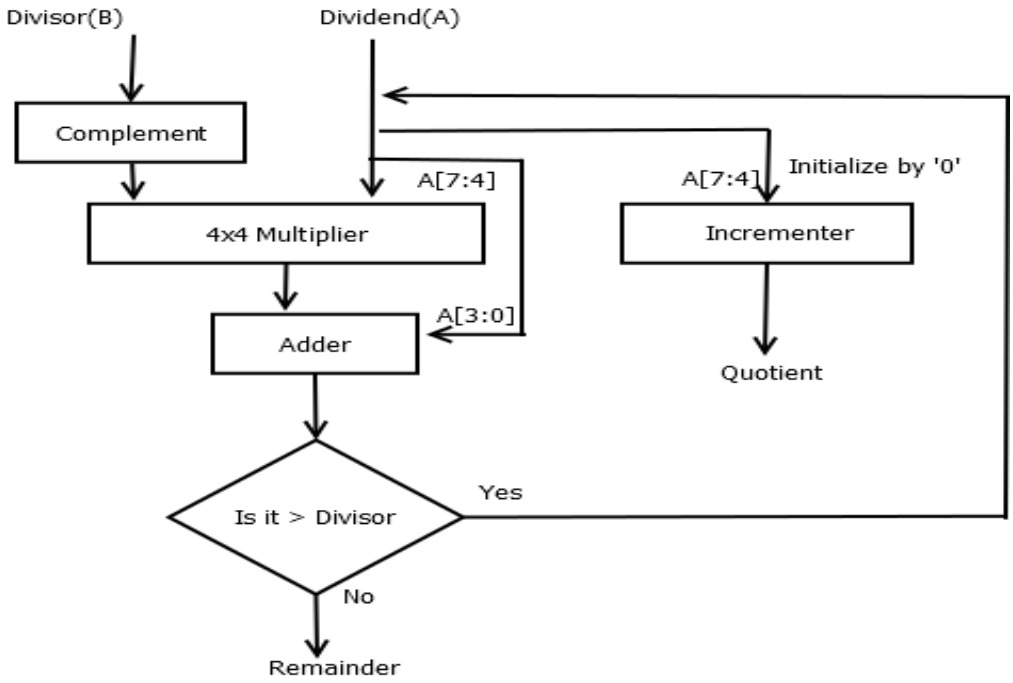


Fig. 5.4 Flowchart of division operation using Nikhilam Sutra

Chapter 6

Testing

6.1 Scope

This project aims to design and implement vedic mathematics based ALU using Verilog HDL. The modules of ALU that perform arithmetic and logic operations are coded using Verilog HDL. These modules are verified individually with the help of Isim simulator of Xilinx EDA tool. The verified modules are integrated to form the ALU. This ALU design is then implemented on the Spartan 6 FPGA development board. The functionality of the ALU is tested in the hardware domain using the LEDs, DIP switches and Push buttons of the FPGA development board.

6.2 Functional Testing

The modules for arithmetic and logical operations designed using Verilog HDL are verified using Isim simulator. The simulation waveforms (test results) are recorded here.

6.2.1 8-bit Adder

The simulation results of the 8-bit adder are shown in Fig. 6.1

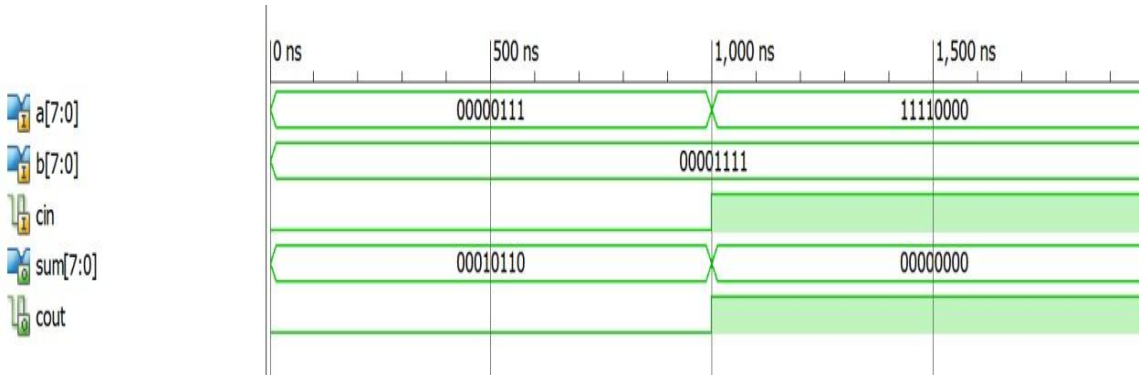


Fig. 6.1 Simulation waveforms of 8-bit adder

6.2.2 8-bit Subtractor

The simulation results of the 8-bit subtractor are shown in Fig. 6.2

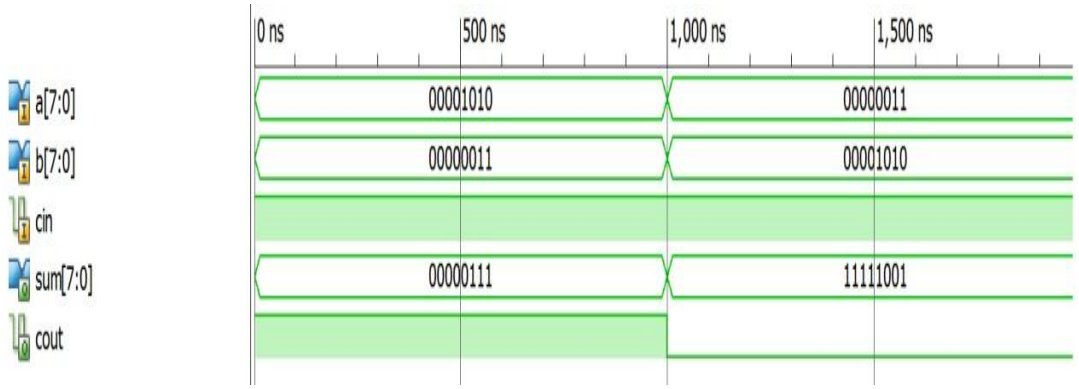


Fig. 6.2 Simulation waveforms of 8-bit subtractor

6.2.3 8x8 Multiplier

The simulation results of the 8x8 multiplier are shown in Fig. 6.3

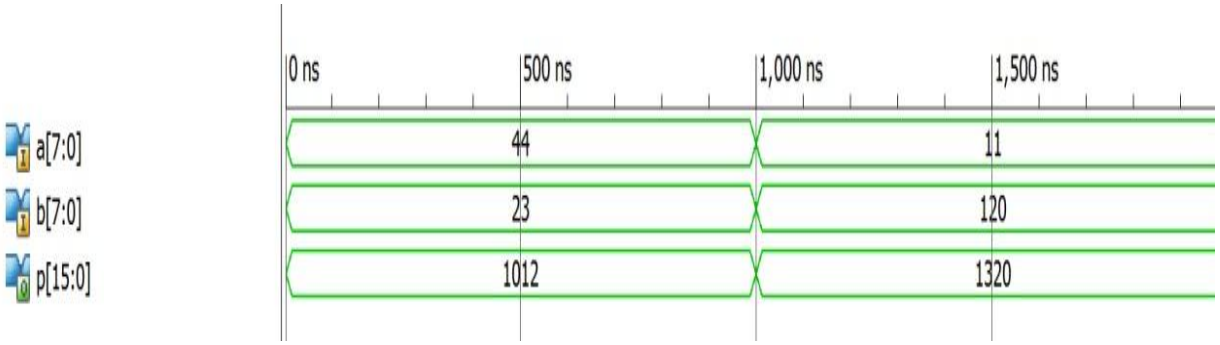


Fig. 6.3 Simulation waveforms of 8x8 multiplier

6.2.4 Division Module

The simulation results of the division module are shown in Fig. 6.4

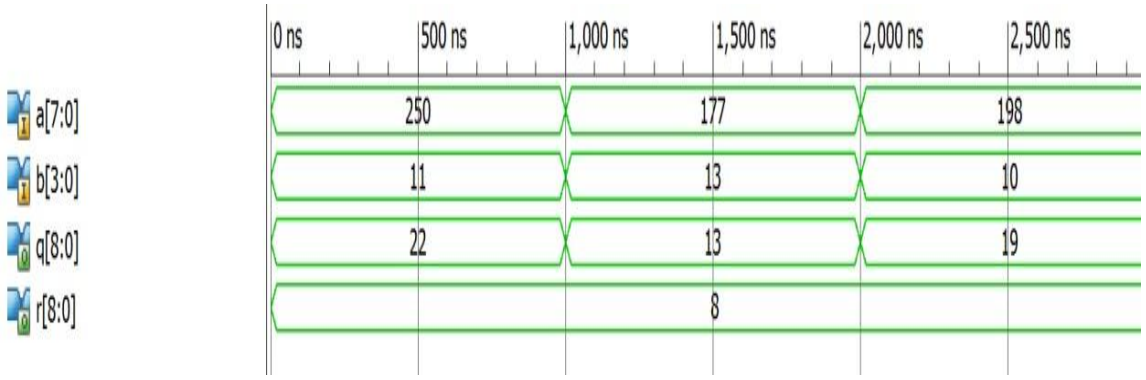


Fig. 6.4 Simulation waveforms of division module

6.2.5 Logic Unit

The simulation results of the logic unit are shown in Fig. 6.5

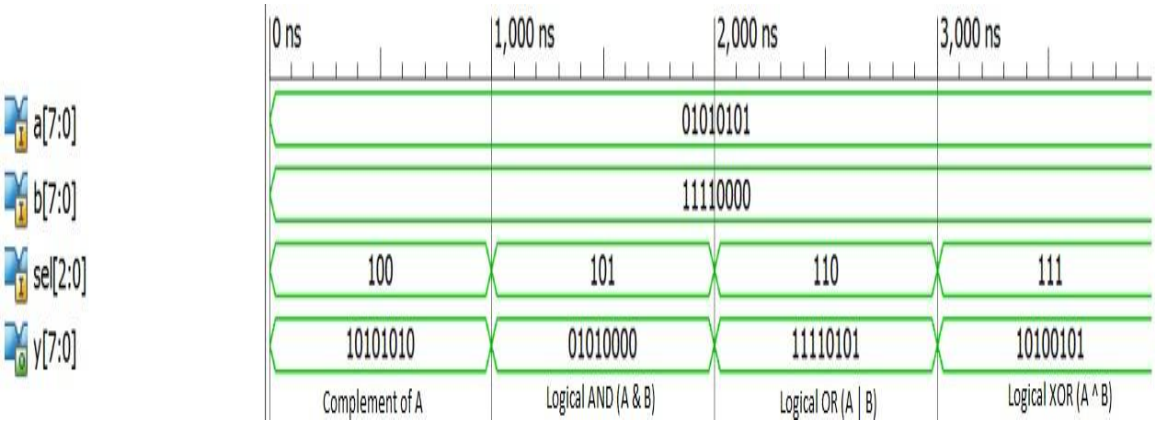


Fig. 6.5 Simulation waveforms of logic unit

6.3 Integration Testing

The functionally verified modules, described in 6.2 are integrated to form the required 8-bit ALU. The ALU is verified first using simulation and then verified in hardware domain. The test results of both domains are recorded here.

6.3.1 Simulation Results of ALU

The simulation results for various operations of the ALU are shown in Fig. 6.6. The functions performed by the ALU are shown in Table 6.1. The operands A and B are of 8-bit size.

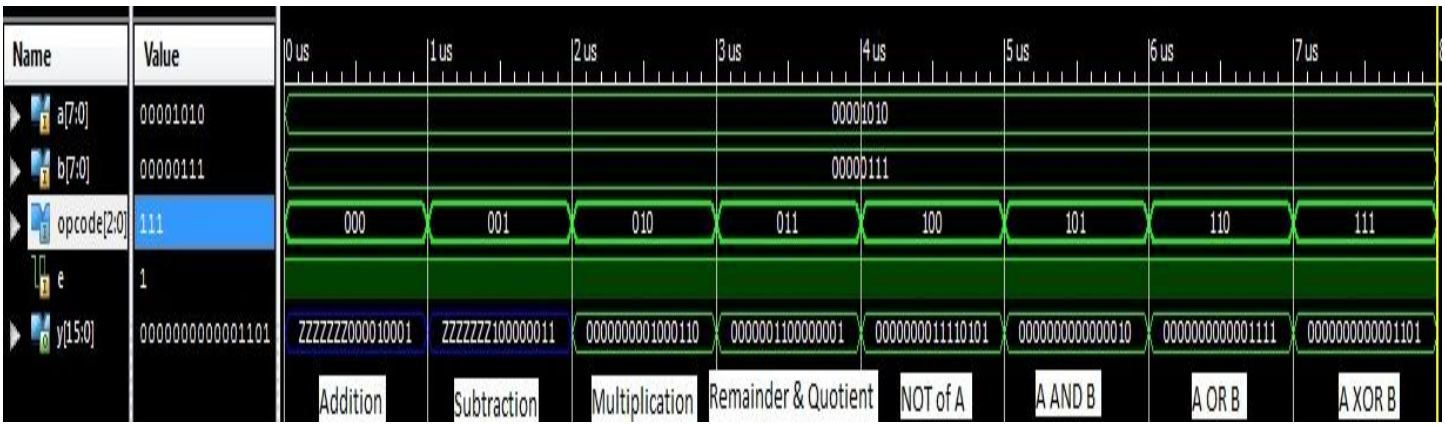


Fig. 6.6 Simulation waveforms of ALU module

Table 6.1 Function table of ALU

Opcode	Operation
000	Addition (A+B)
001	Subtraction (A-B)
010	Multiplication (A*B)
011	Division (A/B)
100	A Complement (~A)
101	Logical AND (A & B)
110	Logical OR (A | B)
111	Logical XOR (A ^ B)

6.3.2 Testing the ALU Design using Spartan 6 Development Board (Mimas V2)

6.3.2.1 Spartan 6 Driver Installation for Windows 7

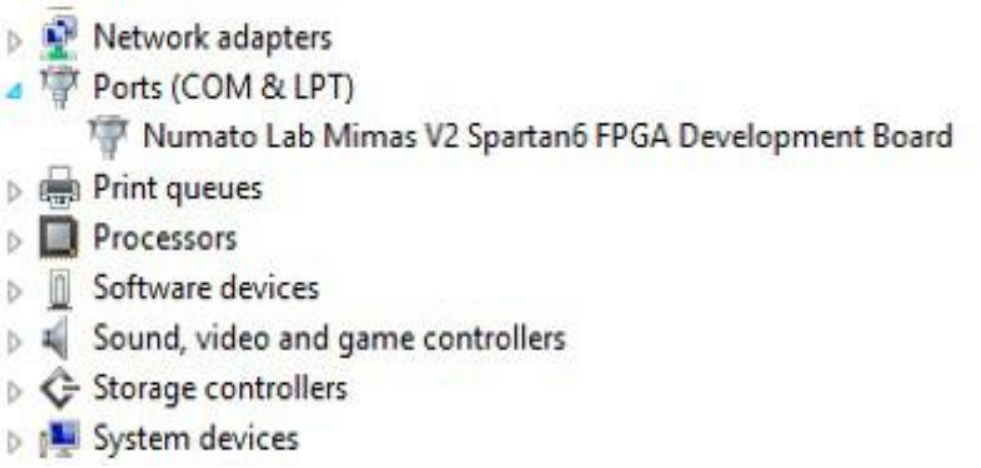


Fig. 6.7 Result of successful installation of the device driver

The Spartan 6 hardware requires a driver to be installed for proper functioning when used with Windows. The driver package can be downloaded from the product page. To install the driver, unzip the contents of the downloaded driver package to a folder. Attach USB cable to the PC and when asked by Windows device installation wizard, point to the folder where driver files are present. When driver installation is complete, the module should appear in Windows Device Manager as a serial port.

6.3.2.2 Generating Bit Stream for Mimas V2

HDL design needs to be converted to bit stream before it can be programmed to FPGA. MIMAS V2 configuration tool at this time accepts only binary (.bin) bit stream created by XILINX ISE (http://www.xilinx.com/tools/webpack.htm). Once the HDL is synthesized, it is easy to create a binary bit stream out of it. Please follow the Steps below to generate binary bit stream from your design using ISE Web Pack.

Step 1: Right click on the "Generate Programming File" option in "Processes" window.

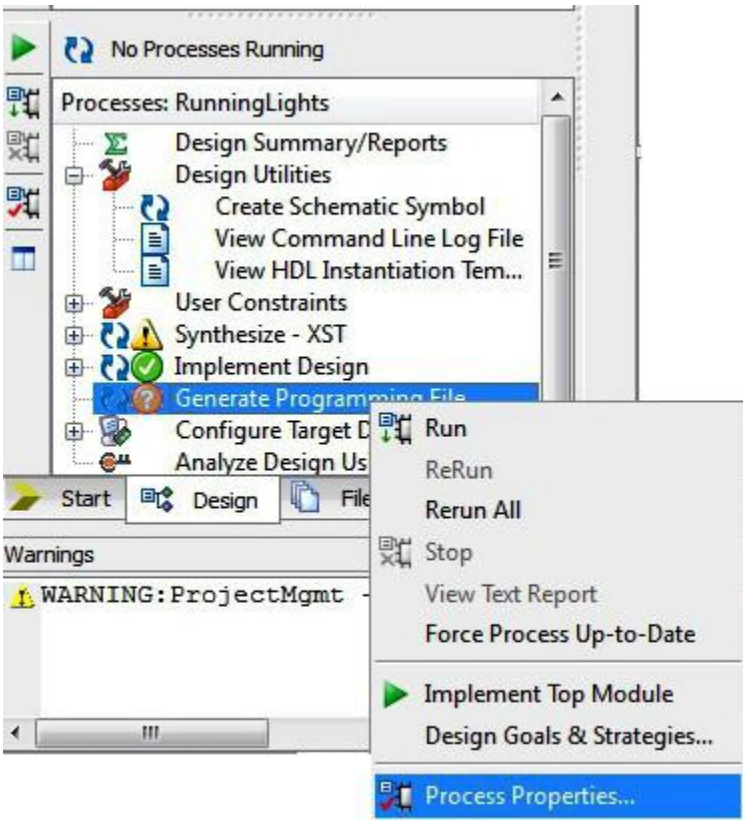


Fig. 6.8 Generation of programming file – step 1

Step 2: Select "Process Properties" from the pop up menu. In the dialog box, check "Create Binary Configuration File" Check box and click "Apply".

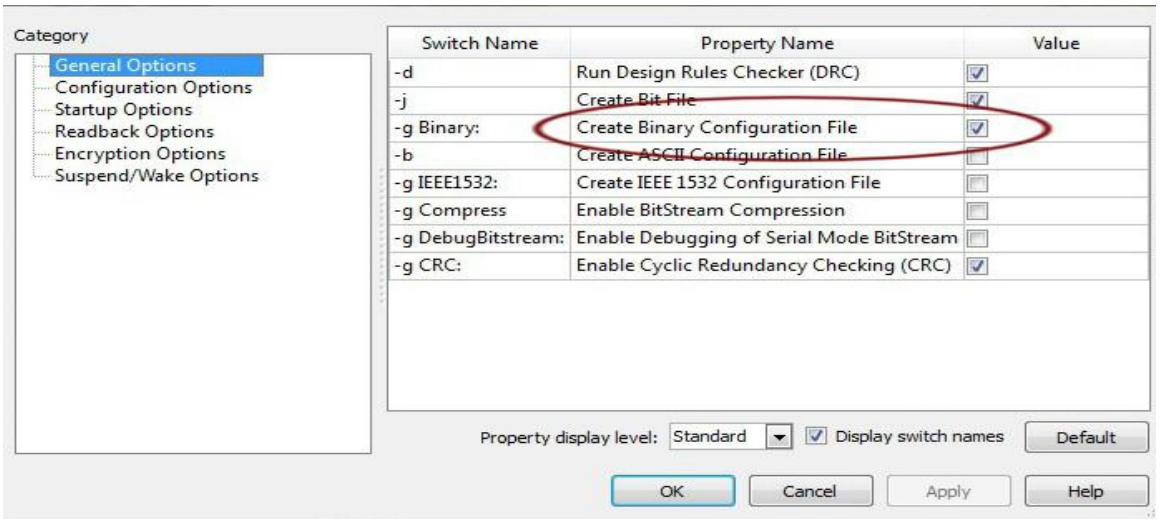


Fig. 6.9 Generation of programming file – step 2

Step 3**:** Click "OK" to close the dialog box. Right click on "Generate Programming File" option again and select "Run". Now you will be able to find a .bin file in the project directory and that file can be used for MIMAS V2 configuration.

6.3.2.3 Powering Up Mimas V2

MIMAS V2 can be powered directly from USB port so make sure that you are using a USB port that can power the board properly. It is recommended to connect the board directly to the PC instead using a hub. It is practically very difficult to estimate the power consumption of the board, as it depends heavily on your design and the clock used. XILINX provides tools to estimate the power consumption. In any case if power from USB is not enough for your application, an external supply can be applied to the board. MIMAS V2 requires two different voltages, a 3.3V and a 1.2V supply. On-board regulators derive these voltages from the USB/Ext power supply.

6.3.2.4 Configuring Mimas V2 Using Configuration Tool

MIMAS V2 has an on-board micro-controller which facilitates easy reprogramming of on board SPI flash through USB interface. The micro-controller receives bit stream from the host application and program it in to the SPI Flash and lets the FPGA boot from the flash. The MIMAS V2 configuration application can be downloaded from www.numato.com for free. When MIMAS V2 is connected to PC, it shows up as a COM port in Device Manager. Run configuration application, select correct COM Port before downloading bit stream. Click on "Open File" to select the bit stream file (.bin) and press "Program" button to download the bit stream. Wait till the download process is finished. Once the download process is over, the configuration controller will try to boot the FPGA from the SPI Flash automatically. Follow the below steps.

Step 1: Make sure you have selected USB configuration mode (Slide SW7 to position 1. Refer to the section "Configuration Mode Selection" for more information). Run MIMAS V2 Configuration Tool and select the correct port (Refer to section "Driver installation" for more information on finding port number). Click Open file button and select the .bin file.

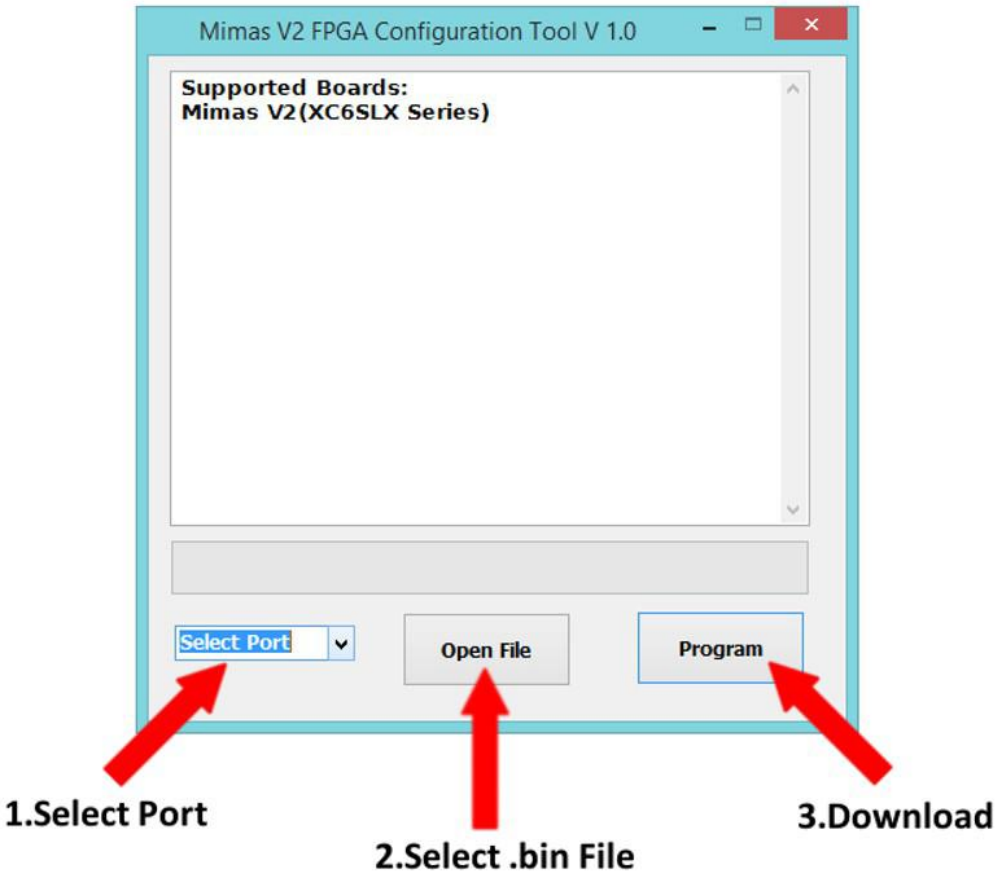


Fig. 6.10 Configuring Mimas V2 Using Configuration Tool – step 1

Step 2: Click on "Program" button. Wait till "Done" appears on the screen.

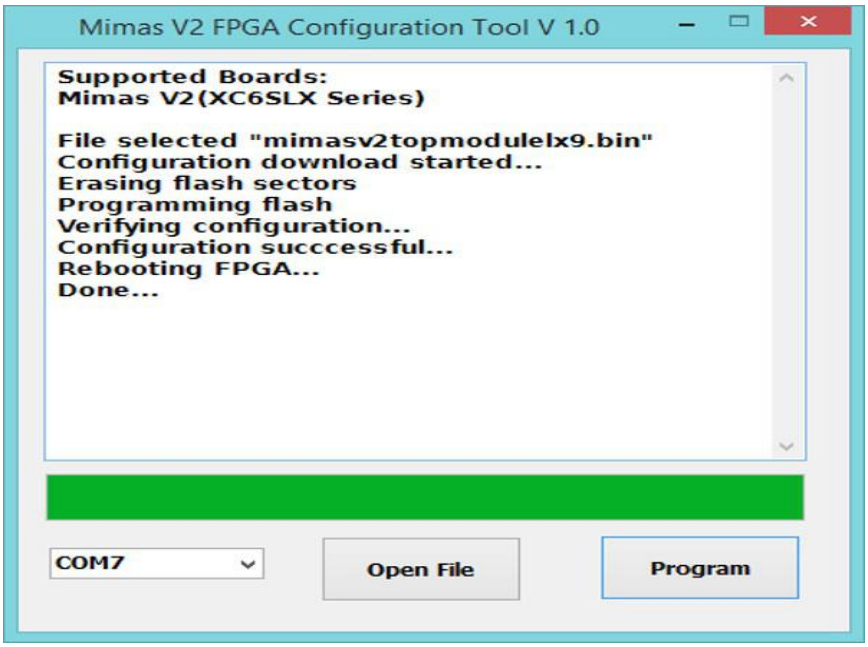


Fig. 6.11 Configuring Mimas V2 Using Configuration Tool – step 2

6.3.2.4 Apparatus Setup for Verification of ALU Design in Hardware Domain

Fig. 6.11 shows the arrangement made for the verification of the ALU design using Mimas V2 Spartan 6 Development Board. Table 6.2 shows the various operations to be performed on the Spartan 6 development board using LEDs, push buttons and DIP switches to test the ALU design.

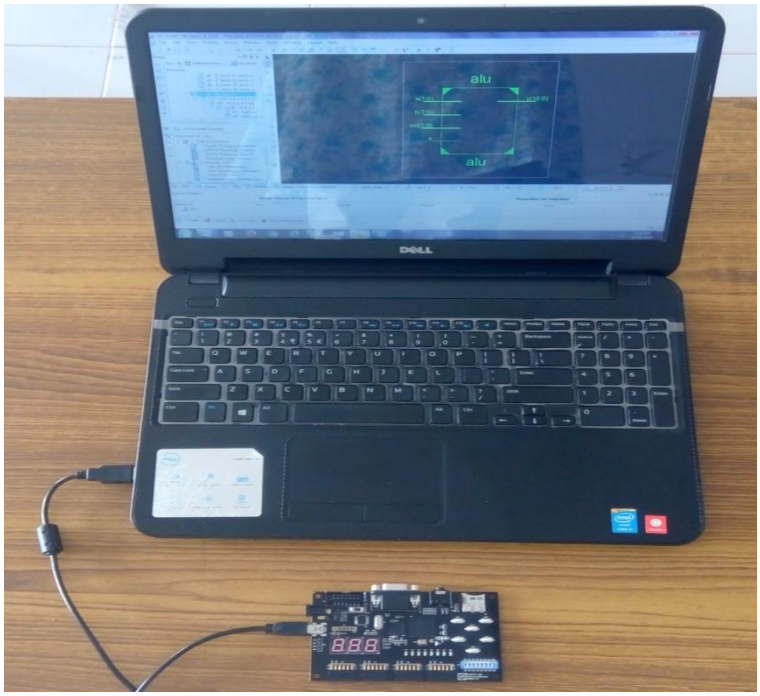


Fig. 6.12 Hardware set-up for design verification

Table 6.2 Operation summary

Sl. No	Push Button	Operation
1	SW6	Reads the 8-bit data given through DIP switches and applies to input A of the ALU. Data is also displayed on LEDs.
2	SW5	Reads the 8-bit data given through DIP switches and applies to input B of the ALU. Data is also displayed on LEDs.
3	SW4	Reads the 8-bit data given through DIP switches and applies lower 3 bits to OPCODE/SELECT input of the ALU. Data is also displayed on LEDs.
4	SW3	Displays the lower byte result of an operation on the LEDs.
5	SW2	Displays the higher byte result of an operation on the LEDs
6	SW1	Enables the ALU to perform the operation according to the given opcode.

Chapter 7

Results and Discussion

7.1 Results Obtained from FPGA Board

The results of various operations performed by the ALU are recorded here.

7.1.1 Addition Operation

Inputs A = 6 = $(00000110)_2$ and B = 3 = $(00000011)_2$; Output Y = 9 = $(00001001)_2$;

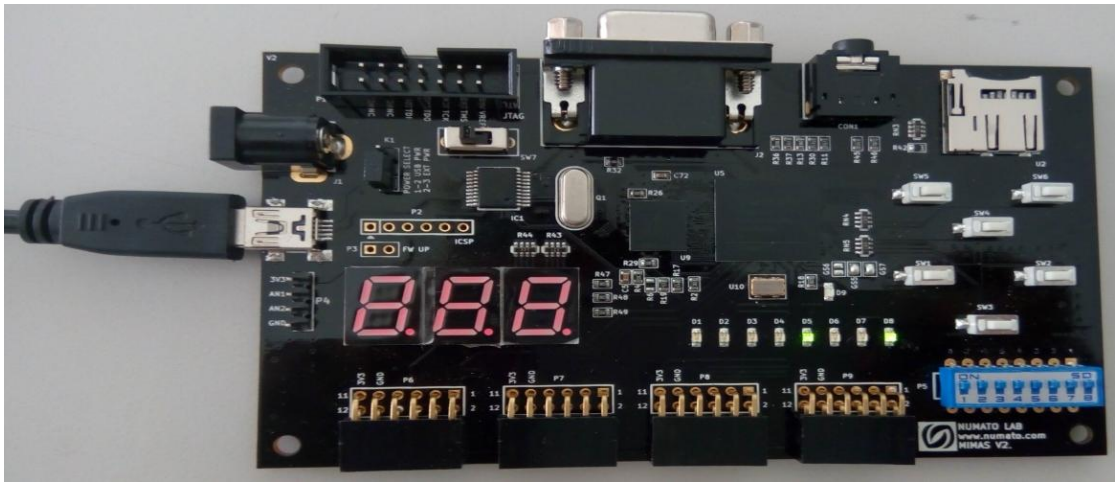


Fig. 7.1 Addition Result

7.1.2 Subtraction Operation

Inputs A = 6 = $(00000110)_2$ and B = 3 = $(00000011)_2$; Output Y = 3 = $(00000011)_2$;

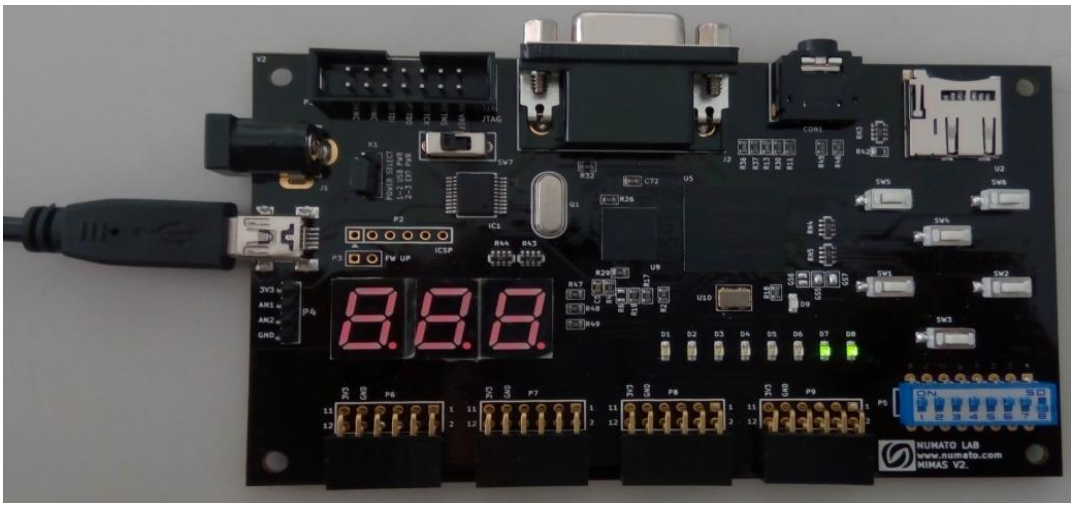


Fig. 7.2 Subtraction Result

7.1.3 Multiplication Operation

Inputs A = 6 = $(00000110)_2$ and B = 3 = $(00000011)_2$; Output Y = 18 = $(00010010)_2$;

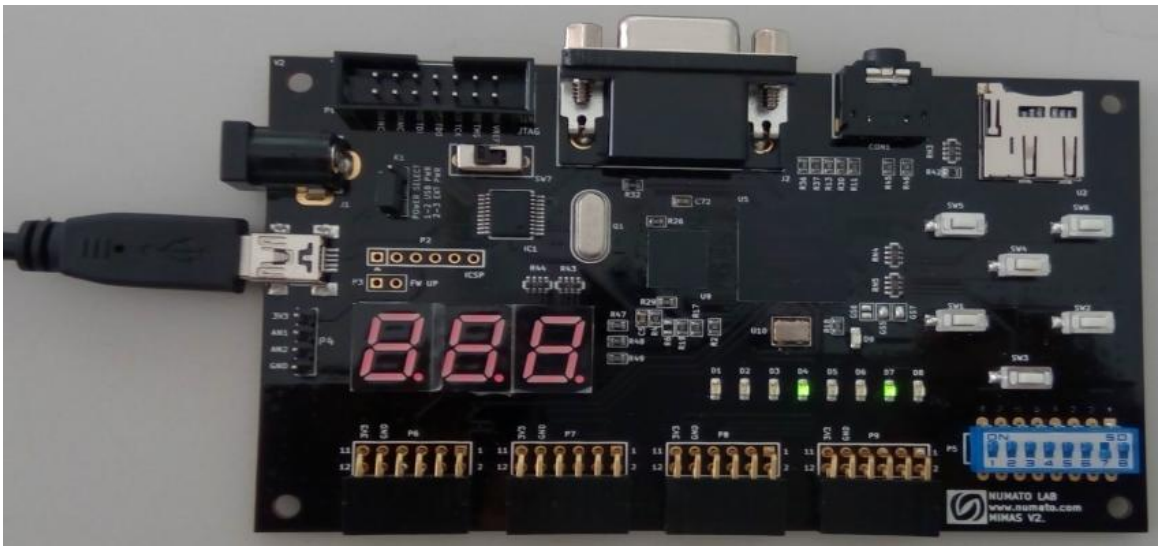


Fig. 7.3 Multiplication Result

7.1.4 Division Operation

Inputs A = 6 = $(00000110)_2$ and B = 3 = $(00000011)_2$;

Output Y = A / B = 2 = $(00000010)_2$

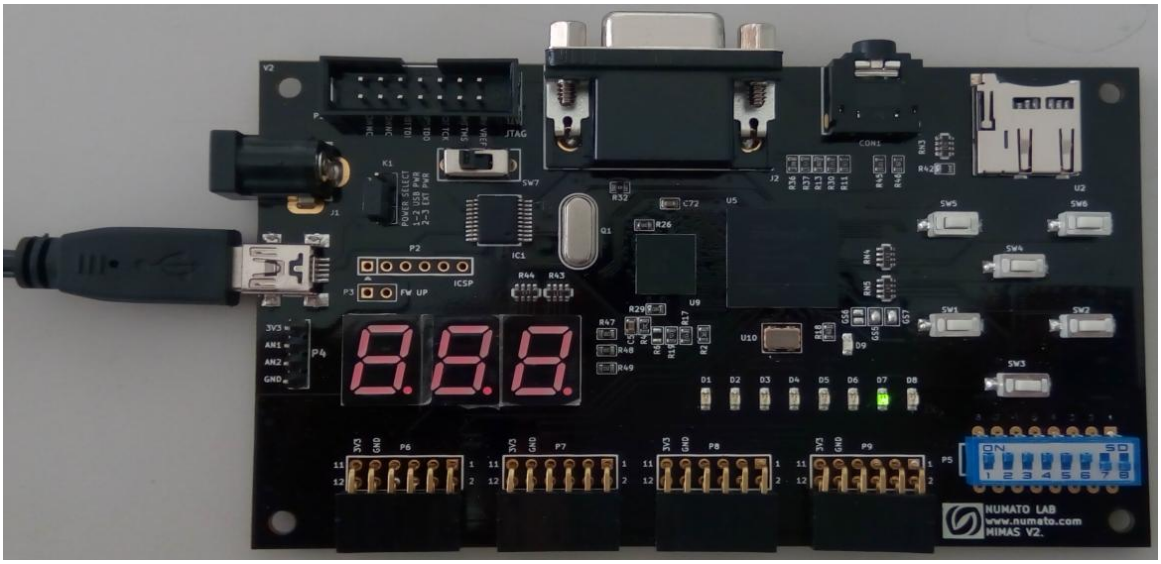


Fig. 7.4 Division Result (Quotient)

7.1.5 Complement of A

Input A = 6 = $(00000110)_2$;

Output Y = not (A) = 249 = $(11111001)_2$;

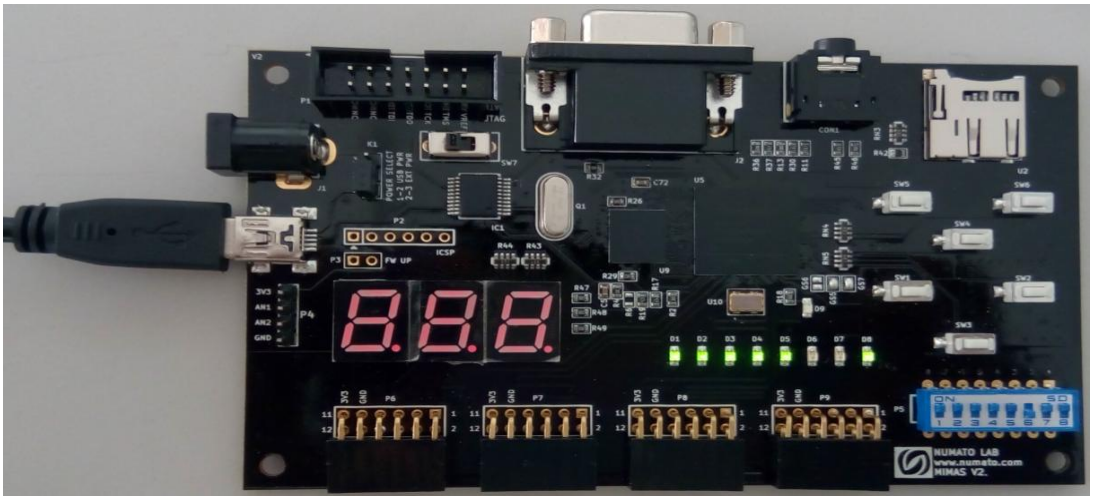


Fig. 7.5 Results of Logical NOT of A Operation

7.1.6 Logical AND Operation

Inputs A = 6 = $(00000110)_2$ and B = 3 = $(00000011)_2$;

Output Y = 2 = $(00000010)_2$;

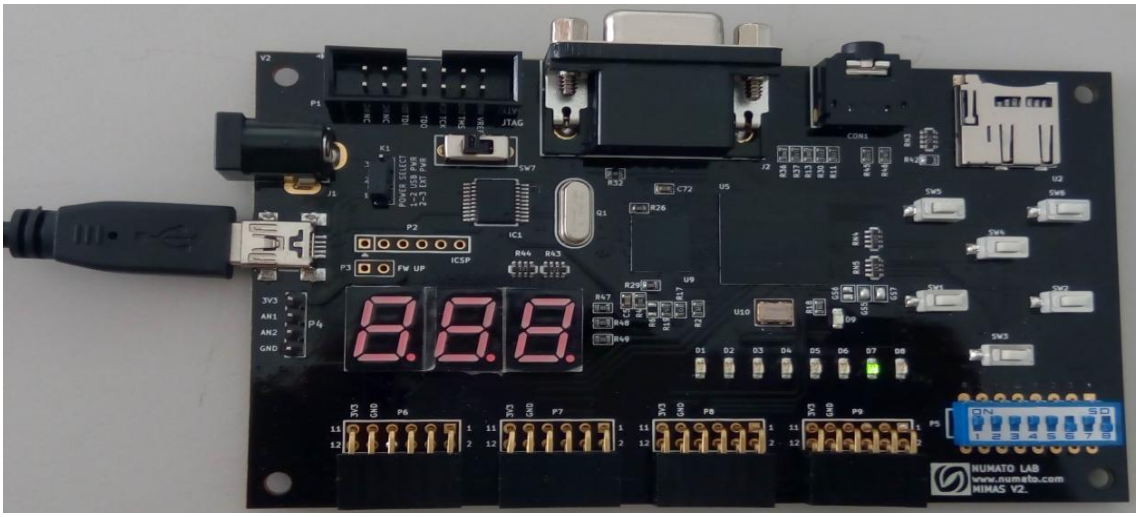


Fig. 7.6 Results of Logical AND Operation

7.1.7 Logical OR Operation

Inputs A = 6 = $(00000110)_2$ and B = 3 = $(00000011)_2$;

Output Y = 7 = $(00000111)_2$;

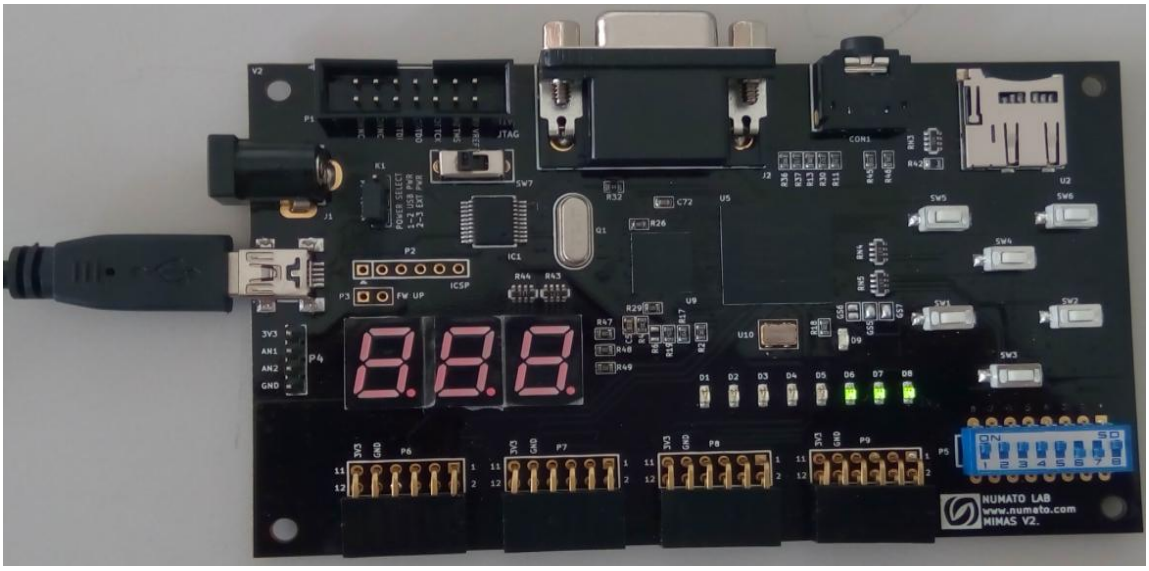


Fig. 7.7 Results of Logical OR Operation

7.1.8 Logical EX-OR Operation

Inputs A = 6 = $(00000110)_2$ and B = 3 = $(00000011)_2$;

Output Y = 5 = $(00000101)_2$;

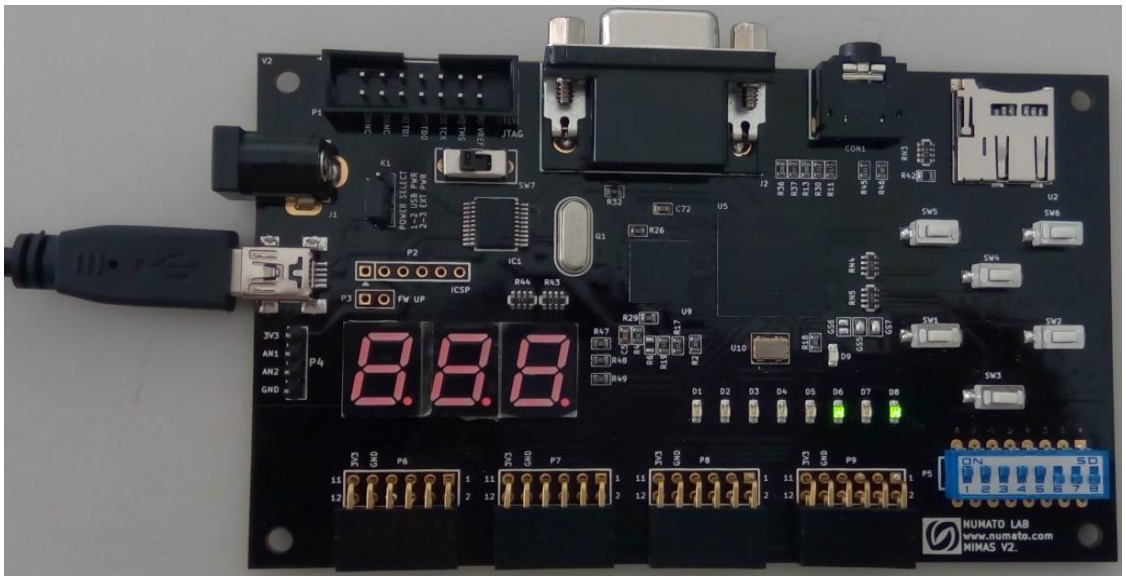


Fig. 7.8 Results of Logical EX-OR Operation

7.2 FPGA Device Utilization Summary

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	112	5720	1%
Number of fully used LUT-FF pairs	0	112	0%
Number of bonded IOBs	32	200	16%

Fig. 7.9 Estimated device utilization summary of the designed ALU

7.3 Xilinx Power Analyzer Report

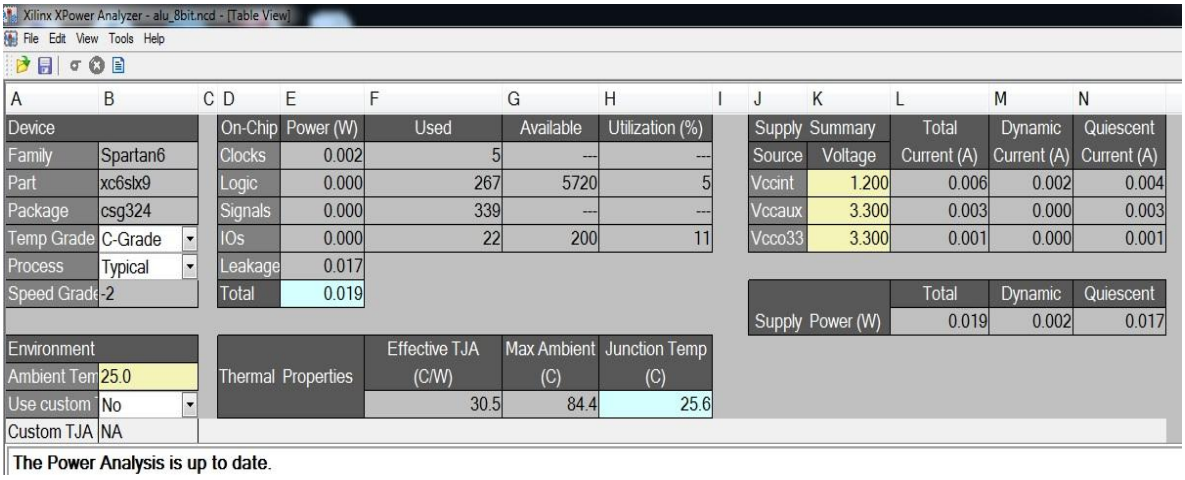


Fig. 7.10 Xilinx XPower Analyzer report for the used Spartan 6 FPGA

Chapter 8

Conclusion and Future Work

8.1 Conclusion

Vedic Mathematics concepts for various mathematical computations are proved to be efficient as compared to conventional methods. Many researchers have proposed computational units based on Vedic Mathematics for various signal processing applications and these designs are proved to be efficient ones. The proposed Vedic Mathematics based ALU uses Urdhva-Tiryakbyham Sutra and Nikhilam Sutra for the implementation of multiplication and division respectively. The proposed ALU architecture would find application in various signal processing areas.

FPGA is advantageous in terms of cost, development time, cost, maintainability and practicability. Stability and minimizing the tendency of mistakes in designs can be solved with all designs in FPGA. FPGA is used to upgrade obsolete integrated circuits reduces hardware circuit board changes, increases productivity, and ensures that the operational constraints are met.

8.2 Limitations

- Signed arithmetic operations are not considered
- Size of the input operands is limited to 8-bit

8.3 Future Work

The designed ALU uses vedic concepts like Urdhva-Tiryakbyham Sutra and Nikhilam Sutra for the implementation of multiplication and division respectively. The ALU can accept 8-bit operands.

The scope for future work lies in the following aspects:

1. Operand size: The operand size can be increased to 16-bit or 32-bit.

2. Increase in speed of operation: The adders used in this design are ripple carry adders. As the operand size increases, these adders consume more time for the computation.

Therefore, for high speed operation look-ahead adders can be used in place of ripple carry adders.

3. Signed arithmetic operations: The designed ALU performs arithmetic operations only on unsigned numbers. The work has to be carried out in future to make the ALU to operate on signed numbers as well.

4. IC design: The designed ALU is implemented on FPGA device. If an optimized synthesis is obtained, then in future a full custom or semi custom IC design of the ALU based on the vedic concepts can be carried out using suitable EDA tools.

8.4 Application

This Vedic Mathematics based ALU can be used in different processor designs to perform various signal processing operations.

8.5 Advantages

- Reduced time delay
- Reduced power consumption
- Cost effective
- Reduced complexity than conventional multipliers

References

Conference/ Journal Papers:

[1] Garima Rawat, Khyati Rathore, Siddharth Goyal, Shefali Kala and Poornima Mittal, "Design and Analysis of ALU: Vedic Mathematics Approach", Proc of International Conference on Computing, Communication and Automation (ICCCA2015), IEEE, 15-16 May 2015, pp 1372 – 1376.

[2] Rahul Nimje, Sharda Mungale, "Design of arithmetic unit for high speed performance using vedic mathematics, International Journal of Engineering Research and Applications, April 2014, pp 26 – 31.

[3] Abhishek Gupta, Utsav Malviya, Vinod Kapse, "A novel approach to design high speed arithmetic logic unit based on ancient vedic multiplication technique", International Journal of Modern Engineering Research, Vol. 2, no. 4, July, 2012, pp 2695 – 2698.

[4] Suchita Kamble, N. N. Mhala, "VHDL implementation of 8- bit ALU", IOSR Journal of Electronics and Communication Engineering, Vol. 1, no. 1, May 2012, pp 07 – 11.

[5] S.P.Pohokar, R.S.Sisal, K.M.Gaikwad, M.M.Patil, Rushikesh Borse, "Design and Implementation of 16 x 16 Multiplier Using Vedic Mathematics", Proc of International Conference on Industrial Instrumentation and Control (ICIC), IEEE, 28-30 May 2015, pp 1174 – 1177.

[6] Surabhi Jain, Mukul Pancholi, Harsh Garg, Sandeep Saini, "Binary Division Algorithm and High Speed Deconvolution Algorithm (Based on Ancient Indian Vedic Mathematics)", Proc of 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), IEEE, 14-17 May 2014, pp 1 – 5.

[7] Abhyarthana Bisoyil, Mitu Baratl, Manoja Kumar Senapati, "Comparison ofa 32-Bit Vedic Multiplier With A Conventional Binary Multiplier", Proc of IEEE International Conference on Advanced Communication Control and Computing Technologies (lCACCCT), 8-10 May 2014, pp 1757 – 1760.

Websites:

[8] http://www.vedicmathsofindia.com/

[9] https://en.wikibooks.org

[10] https://bineeshthomas.wordpress.com

Textbooks:

[11] Nazeih M.Botros, HDL Programming (VHDL and Verilog), Dreamtech press (Available) through John Wiley – India and Thomson Learning), 2006 Edition.

[12] Samir Palnitkar, Verilog HDL, Pearson Education, IEEE Conference Publications, 1994.

[13] J.Bhaskar, AVerilog HDL Primer, BS Publications.