Design of High Performance 64-bit ALU using Vedic Mathematics

Charan Kumar Pagadala
Department of Electronics and Communication Engineering
Narayana Engineering College Nellore
Nellore,India-524002
charanpagadala0117@gmail.com

Husamuddin Ahmed MD
Department of Electronics and Communication Engineering
Narayana Engineering College Nellore
Nellore,India-524002
jkhusam786@gmail.com

Saketh Kumar Muppasani
Department of Electronics and Communication Engineering
Narayana Engineering College Nellore
Nellore,India-524002
saketh8899@gmail.com

Leela Sathvik Muppaneni
Department of Electronics and Communication Engineering
Narayana Engineering College Nellore
Nellore,India-524002
m.l.sathvikbinnu02@gmail.com

Mr. A. Siva Sai Kumar
Assistant Professor
Department of Electronics and Communication Engineering Nellore
Narayana Engineering College
Nellore,India-524002

Abstract— Nowadays due to the growing demand for improving processor performance in handling the complex algorithms and multi functioning making the all-processor cores are going to integrate on single chip. Even though the burden on the processor is not reducing. In order to reduce this we should provide the coprocessor for supporting operations done by main processor, these coprocessors will perform numeric operation like addition, multiplication, DSP application, etc. The speed of the processor will depend on the speed of the coprocessors. Vedic mathematics is the ancient type of mathematics which are having unique technique of 16 formulas to find solution of various application in the fast way. The paper provides the details of a 64-bit ALU design based on Vedic Sutras like Urdhva Tiryakbhyam. The results show that the Vedic sutras are applicable for multiplication operation. The arithmetic module implemented is most efficient in terms of delay reduction because of Vedic sutras. Here we are designing an ALU which was based on these maths using Verilog HDL and synthesised in Xilinx ISE, found that it's having enhanced performance.

I. INTRODUCTION

One of the arithmetic operations is multiplying one number by another. Operations like multiplication are mostly required functions, presently enforced in several DSPs for applications like convolution, FFTs, filters and ALU (Arithmetic Logic Unit) of Microprocessor. A recurrently preferred operation is multiplication, it's necessary to design multiplier with reduced delay and efficient power utilization. Arithmetic calculations are working for many operations considering from simple routine work such as counting or multiplying to advanced science and business calculations.

Hence, there is requirement of a fast and effective arithmetic unit in computers. Array Multiplication takes less time when compared to the partial product parallel calculation method. The delay produced is the time required for the signals to pass through the gates of multiplication array. Booth multiplier uses massive booth arrays for multiplier designs and exponents calculations with prime speed that successively demands partial add and partial carry registers. Two n-bit operands multiplication employing a radix-4 booth recording multiplier factor wants around n-bits/ (2k) clock cycles to get part of the final product, where k indicates the number of booth recorder adder stages.

Urdhva Tiryakbhyam Sutra based "Vertical Crosswise Algorithm" can be used to design digital multiplier like the Array multiplier. The ancient formula (sutra) provides the way to find product of N x N, of N bits each multiplicand by considering smaller parts of size (N/2 = n), say. These parts will once more be fragmented into smaller numbers (n/2 each) until it tends to reach 2x2 sizes. Thus, it is streamlining the operation into tree like structure.

VEDIC MATHEMATICS

Vedic mathematics - a gift given to this world by the ancient sages of India. A system which is far simpler and more enjoyable than modern mathematics. The word "Vedic" is derived from the word "Veda" which means the store-house of all knowledge. Vedic math was rediscovered from the ancient Indian scriptures between 1911 and 1918 by Sri Bharati Krishna Tirthaji (1884-1960), a scholar of sanskrit, Mathematics, History and Philosophy.

Vedic mathematics is part of four Vedas (books of wisdom). It is part of Sthapatya- Veda (book on civil engineering and architecture), which is an upa-veda (supplement) of Atharva Veda. It gives explanation of several mathematical terms including arithmetic, geometry (plane, co-ordinate), trigonometry, quadratic equations, factorization and even calculus. His Holiness Jagadguru Shankaracharya Bharati Krishna Teerthaji Maharaja (1884-1960) comprised all this work together and gave its mathematical explanation while discussing it for various applications. Swamiji constructed 16 sutras (formulae) and 16 Upa sutras (sub formulae) after extensive research in Atharva Veda. Obviously these formulae are not to be found in present text of Atharva Veda because these formulae were constructed by Swamiji himself. Vedic mathematics is not only a mathematical wonder but also it is logical. That's why

it has such a degree of eminence which cannot be disapproved.

Design of 64x64 Multiplier

Similar to the previous design of 64x64 multiplier, we need 4 such 32x32 multipliers along with 2 adders to develop 64x64 multiplier. Here we need to first design 64 bit and 96 bit adders and by proper instantiating of the module and connections as shown in the figure we have designed an 64x64 bit multiplier. At this point of time it's necessary for you to even verify the RTL code and check if the hardware is as per your design. Plan Ahead tool by Xilinx gives better view of the hardware design with elaborate option. Refer the addition tree diagram to know the process for 64x64 multiplier:

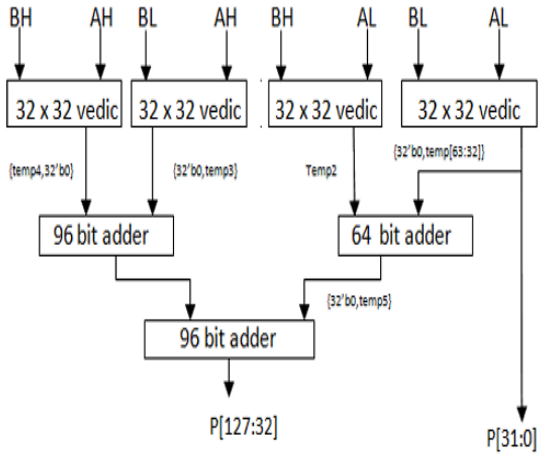

Fig :- 64 Bit Vedic Multiplier

II. LITERATURE SURVEY

[1] Multiplier is one of the most desirable components in most of the processors designed today. The speed of multiplier determines the speed of the processor. So there is a need of high speed multiplier. In this paper, a novel method for Multiplication is proposed by combining Modified Booth algorithm, Wallace tree architecture and Hybrid adder design. Modified Booth Multiplier reduces the number of partial products and has least latency as compared to other multiplier designs. Wallace Tree increases the speed by parallel addition of partial products. Adders play an important role in addition of partial products. If the speed at which the addition operation is performed is increased than the overall speed of the multiplier design will increase. So the main focus in this paper is to increase the speed of the adder. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

A novel hybrid adder design is used in the multiplier design which, has less delay and occupies less area. Area, delay and power complexities of the proposed Multiplier design are reported. The proposed Modified Booth Multiplier design shows better performance compare to conventional method using Carry Look-Ahead Adder and has advantages of reduced area overhead and critical path delay. The proposed multiplier design has been synthesized using Xilinx ISE 10.1 design tool and simulated using ModelSim15.7g.The programming language used is Verilog HDL.

[2] Vedic mathematics is the name given to the ancient Indian system of mathematics that was rediscovered in the early twentieth century from ancient Indian sculptures (Vedas). It mainly deals with Vedic mathematical formulae and their application to various branches of mathematics. The algorithms based on conventional mathematics can be simplified and even optimized by the use of Vedic Sutras.

These methods and ideas can be directly applied to trigonometry, plain and spherical geometry, conics, calculus (both differential and integral), and applied mathematics of various kinds. In this paper new multiplier and square architecture is proposed based on algorithm of ancient Indian Vedic Mathematics, for low power and high speed applications. It is based on generating all partial products and their sums in one step. The design implementation on ALTERA Cyclone –II FPGA shows that the proposed Vedic multiplier and square are faster than array multiplier and Booth multiplier.

[3] Vedic mathematics is the name given to the ancient Indian system of mathematics that was rediscovered in early twentieth century. Vedic mathematics is mainly based on sixteen principles or word-formulae which are termed as Sutras. We discuss a possible application of Vedic mathematics to digital signal processing in the light of application of Vedic multiplication algorithm to digital multipliers. A simple digital multiplier (referred henceforth as Vedic multiplier) architecture based on the Urdhva Tiryakbhyam (Vertically and Cross wise) Sutra is presented. This Sutra was traditionally used in ancient India for the multiplication of two decimal numbers in relatively less time. In this paper, after a gentle introduction of this Sutra, it is applied to the binary number system to make it useful in the digital hardware. The hardware architecture of the Vedic multiplier is presented and is shown to be very similar to that of the popular array multiplier. It is also equally likely that many such similar technical applications might come up from the storehouse of knowledge, Veda, if investigated properly.

[4] The ever increasing demand in enhancing the ability of processors to handle the complex and challenging processes has resulted in the integration of a number of processor cores into one chip. Still the load on the processor is not less in generic system. This load is reduced by supplementing the main processor with Co-Processors, which are designed to work upon specific type of functions like numeric computation, Signal Processing, Graphics etc. The speed of ALU depends greatly on the multiplier. In

algorithmic and structural levels, numerous multiplication techniques have been developed to enhance the efficiency of the multiplier which concentrates in reducing the partial products and the methods of their addition but the principle behind multiplication remains the same in all cases. Vedic Mathematics is the ancient system of mathematics which has a unique technique of calculations based on 16 Sutras. Employing these techniques in the computation algorithms of the coprocessor will reduce the complexity, execution time, area, power etc. Though there are many sutras employed to handle different sets of numeric, exploring each one gives new results. Our work has proved the efficiency of Urdhva Triyagbhyam– Vedic method for multiplication which strikes a difference in the actual process of multiplication itself. It enables parallel generation of intermediate products, eliminates unwanted multiplication steps with zeros and scaled to higher bit levels using Karatsuba algorithm with the compatibility to different data types. This sutra is to be used to build a high speed power efficient multiplier in the coprocessor.

III. PROPOSED SYSTEM

Proposed multiplier architecture of 64x64 bit Vedic multiplier and major change adopted here is use of half adder for addition of partial products. The proposed research work specifies the modified version of binary Vedic multiplier using vedic sutras of ancient vedic mathematics. It provides modification in preliminarily implemented vedic multiplier. The modified binary vedic multiplier is preferable has shown improvement in the terms of the time delay and also device utilization. The proposed technique was designed and implemented in Verilog HDL. For HDL simulation, modelsim tool is used and for circuit synthesis, Xilinx is used. The simulation has been done for 4 bit, 8 bit, 16 bit, 32 bit, 64 bit operation. Only for 64 bit binary Vedic multiplier technique the simulation results are shown.

Any proposed system must be efficient in terms of power, speed and size as per growing technology. In early days Vedic mathematics is based on 16 Vedic sutras. By using Vedic methods the mathematical operations are fast and the processing speed to perform the operations can be improved. There has been many existing binary multipliers which are efficient.

The proposed system is used to reduce the carry propagation time and to optimize the hardware complexity level for 64 bit Vedic multiplier operation. The proposed system is to implement vertical and cross-wise equation based logical 64 bit multiplier operation.

The high speed processor requires high speed multipliers and the Vedic Multiplication technique is very much suitable for this purpose Multipliers are extensively used in Microprocessors, DSP and communication applications. For higher order multiplications, a huge number of adders are to be used to perform the partial product addition. The need of high speed multiplier is increasing as the need of high speed processors are increasing higher throughout arithmetic operation.

The multiplier architecture is mainly used to the ALU UNIT. Because the architecture optimization process is only possible in the VLSI domain. The Verilog HDL language is mainly used to improve the accuracy compare to another type of process. The XILINX 14.5 software is used in our project. Our process was developed in Verilog HDL language using XILINX software. It is an ancient technique, which simplifies multiplication, divisibility, complex numbers, squaring, cubing, square roots and cube roots. Even recurring decimals and auxiliary fractions can be handled by Vedic mathematics.

Our work is to modify the 64-bit multiplier architecture using Vedic mathematics technique. This technique is to implement the vertical and cross-wise equation based process. First we design a 4-bit multiplication operation and to develop the 8-bit multiplication process using 4bit multiplication result. Finally we design a 64 bit multiplier architecture using a 4-bit, 8-bit ,16bit and 32bit multiplication process. This structure is to reduce the complexity and time level.

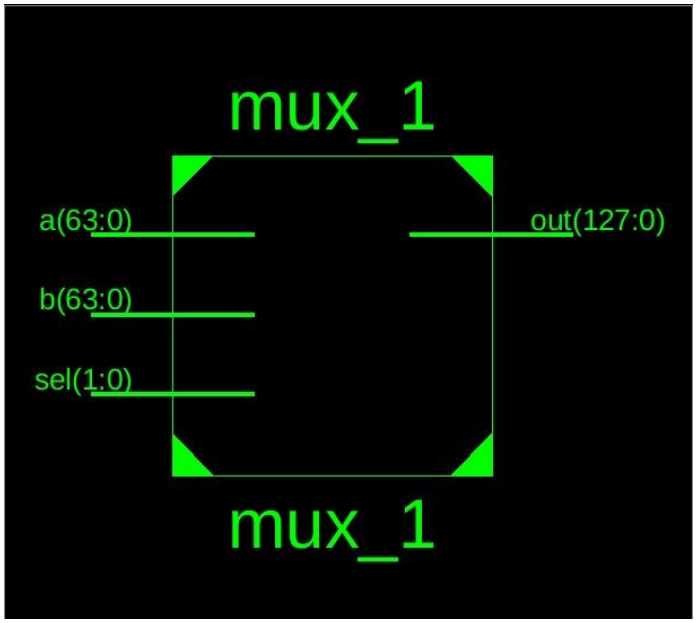

Fig :- 64x64 Pin Diagram

A 64-bit Vedic multiplier is designed by using Urdhva Tiryagbhyam Sutra of ancient Vedic mathematics using VHDL and simulated on Xilinx ISE 14.7.

IV. INTRODUCTION TO VLSI

Digital systems are highly complex at their most detailed level. They may consist of millions of elements i.e., transistors or logic gates. For many decades, logic

schematics served as then Gur Franca of logic design, but not anymore. Today, hardware complexity has grown to such a degree that a schematic with logic gates is almost useless as it shows only a web of connectivity and not functionality of design. Since the 1970s, computer engineers, electrical engineers and electronics engineers have moved toward Hardware description language (HDLs).

Digital circuit has rapidly evolved over the last twenty five years. The earliest digital circuits were designed with vacuum tubes and transistors. Integrated circuits were then invented where logic gates were placed on a single chip. The first IC chip was small scale integration (SSI) chips where the gate count is small. When technology became sophisticated, designers were able to place circuits with hundreds of gates on a chip. These chips were called MSI chips with advent of LSI; designers could put thousands of gates on a single chip. At this point, design process is getting complicated and designers felt the need to automate these processes. With the advent of VLSI technology, designers could design single chip with more than hundred thousand gates. Because of the complexity of these circuits computer aided techniques became critical for verification and for designing these digital circuits.

One way to lead with increasing complexity of electronic systems and the increasing time to market is to design at high levels of abstraction. Traditional paper and pencil and capture and simulate methods have largely given way to the described UN synthesized approach. For these reasons, hardware description languages have played an important role in describe and synthesis design methodology. They are used for specification, simulation and synthesis of an electronic system. This helps to reduce the complexity in designing and products are made to be available in market quickly.

VLSI DESIGN FLOW

Typical design flow for designing VLSI circuits is shown in the tool flow diagram. This design flow is typically used by designers who use HDLs. In any design, specification is first. Specification describes the functionality, interface and overall architecture of the digital circuit to be designed. At this point, architects need not think about how they will implement their circuit. A behavioural description is then created to analyze the design in terms of functionality, performances and other high level issues. The behavioural description is manually converted to an RTL (Register Transfer Level) description in an HDL. The designer has to describe the data flow that will implement the desired digital circuit. From this point onward the design process is done with assistance of CAD tools.

Logic synthesis tools convert the RTL description to a gate level net list. A gate level net list is a description of the circuit in terms of gates and connections between them. The gate level net list is input to an automatic place and route tool, which creates a layout. The layout is verified and then fabricated on a chip. Thus most digital design activity is concentrated on manually optimizing the RTL description of the circuit. After the RTL description is frozen, CAD tools are available to assist the designer in further process Designing at RTL level has shrunk design cycle times from years to a few months.

EMERGENCE OF HARDWARE DESCRIPTION LANGUAGE

As designs got larger and complex, logic simulation assumed an important role in design process. For a long time, programming languages such as Fortran, Pascal & c were been used to describe the computer programs that were been used to describe the computer programs that were sequential in nature. Similarly in digital design field, designers felt the need for a standard language to describe digital circuits. Thus HDL is come in to existence. HDLs allowed the designers to model the concurrency of processes found in hardware elements. HDLs such as VERILOG HDL & VHDL (Very-high speed integrated circuit hardware description language).

HISTORY OF VERILOG

Verilog was started in the year 1984 by Gateway Design Automation Inc as a proprietary hardware modelling language. It is rumoured that the original language was designed by taking features from the most popular HDL language of the time, called HiLo, as well as from traditional computer languages such as C. At that time, Verilog was not standardized and the language modified itself in almost all the revisions that came out within 1984 to 1990.

Verilog simulator first used in 1985 and extended substantially through 1987. The implementation of Verilog simulator sold by Gateway. The first major extension of Verilog is Verilog-XL, which added a few features and implemented the infamous "XL algorithm" which is a very efficient method for doing gate-level simulation. Later 1990, Cadence Design System, whose primary product at that time included thin film process simulator, decided to acquire Gateway Automation System, along with other Gateway products., Cadence now become the owner of the Verilog language, and continued to market Verilog as both a language and a simulator. At the same time, Synopsys was marketing the top-down design methodology, using Verilog. This was a powerful combination.

In 1990, Cadence organized the Open Verilog International (OVI), and in 1991 gave it the documentation for the Verilog Hardware Description Language. This was the event which "opened" the language.

Distinct parts of a Verilog module consists of are as shown in below figure. The keyword module is the beginning of a module definition. In a module definition the module name, port list, port declarations, and optional parameters must come first in its definition. If the module has any ports to interact with the external environment then only Port list and port declarations are present. There are five components within a module

- Variable declarations,
- Dataflow statements
- Instantiation of lower modules
- Behavioural blocks
- Tasks or functions.

The end module statement must always come last in a module definition. All components except module, module name, and end module are optional and can be mixed and matched as per design needs. Multiple modules definition in a single file are allowed by Verilog. In the file the modules can be defined in any order.

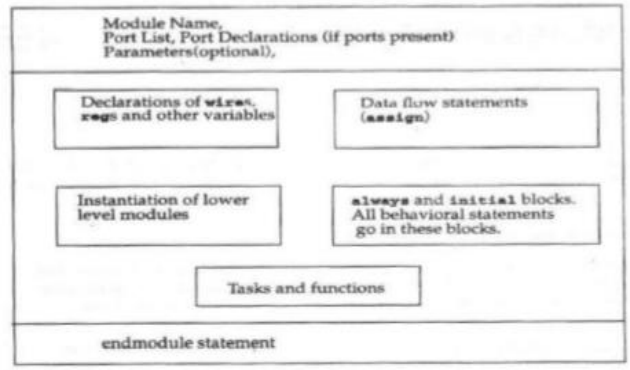

Fig :- Module Design

PORT CONNECTION RULES

A port consisting of two units, primary unit is into the module and secondary unit is out of the module. The primary and secondary units are connected. When modules are instantiated within other modules there are rules governing port connections within module. If any port connection rules are violated then the Verilog simulator complains. The figure 5.6 shows the port connection rules.

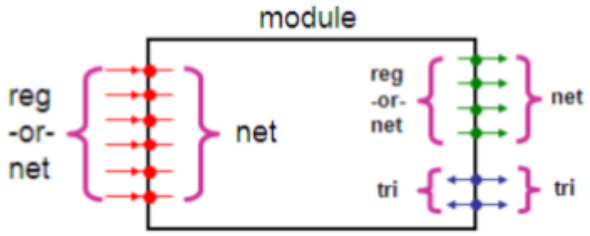

Fig :-Port Connection Rules

V. SOFTWARE REQUIREMENT

XILINX ISE:

Create a New Project Create a new ISE project which will target the FPGA device on the Spartan-3 Start-up Kit demo board. To create a new project: 1. Select File > New Project... The New Project Wizard appears. 2. Type tutorial in the Project Name field. 3. Enter or browse to a location (directory path) for the new project. A tutorial subdirectory is created automatically. 4. Verify that HDL is selected from the Top-Level Source Type list. 5. Click Next to move to the device properties page. 6. Fill in the properties in the table as shown below:

- ♦ Product Category: All
- ♦ Family: Spartan3
- ♦ Device: XC3S200
- ♦ Package: FT256
- ♦ Speed Grade: -4
- ♦ Top-Level Source Type: HDL

VI. RESULTS

Fig :- Output of 64x64 Vedic Miltiplier

VII. CONCLUSION

In this paper 64 bit ALU using Vedic mathematics has implemented using Vedic sutras. In this we are using Vedic sutra - that is Urdhva Tiryakbhyam. This sutra is based up on vertical cross-coupled algorithm. By using this sutra, the multiplication process will be done in simple way and its consumes less time too, when compare to other multiplier. In this we are using addition, subtraction, multiplier (Vedic multiplier) is connecting to mux it will be operate by control unit.

The modules designed using Vedic mathematics is verified through Xilinx ISE. Urdhva Tiryakbhyam Sutra is highly efficient algorithm for multiplication. Urdhva Tiryakbhyam sutras help to bring down time delay in multiplication operation.

REFERENCES

[1] P. Burton and D. R. Noaks, "High - Speed Iterative Multiplier", Electronics Letters, Vol. 4, p. 262,1968.

[2] Soojin Kim and Kyeongsoon Cho, "Design of High Speed Modified Booth Multipliers".

[3] Jagadguru Swami Sri Bharati Krishna Tirthji Maharaja, "Vedic Mathematics", Motilal Banarasidas, Varanasi, India,1986.

[4] Honey Durga Tiwari, Ganzorig Gankhuyag, Chan Mo Kim, Yong Beom Cho, "Multiplier design based on ancient Indian Vedic Mathematics".

[5] Harpreet Singh Dhillon, Abhijit Mitra," A Digital Multiplier Architecture using UrdhavaTiryakbhyam Sutra of Vedic Mathematics", Indian Institute Of Technology, Guwahatti.

[6] Aamir A Farooqui, Vojin G Oklobdzija "General Data Path Organization of a MAC Unit for VLSI Implementation of DSP Processors", 1998.