IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)
e-ISSN: 2278-1676, p-ISSN: 2320-3331
PP 12-16
www.iosrjournals.org

FPGA implementation of 64 bit RISC processor with Vedic multiplier using VHDL

Tariquzzaman[1], Syed Rizwan Ali[2], Nahid Kausar[3]
[1]*Electronics & telecomm. Engg. A.C.E.T/R.T.M. Nagpur University, INDIA*
[2]*Electronics & telecomm. Engg. A.C.E.T/R.T.M. Nagpur University, INDIA*
[3]*Electronics & telecomm. Engg. A.P.N/M.S.B.T.E. Pune , INDIA*

ABSTRACT: *In this paper the FPGA based 64 bit RISC processor with Vedic multiplier feature implemented using VHDL .The VHDL code is versatile language and supports FPGA. System- On- Chip (SoC) and Sparton 3E kit. In today's technology RISC processor plays a vital role and RISC system shorten execution time by reducing the clock cycle per instruction and it can address enormous amount of memory upto 16 Exabyte. Multiplication is Fundamental function arithmetic operation in Digital signal processing application such as FFT, Convolution. Typical multiplier requires considerable amount of computation time to implement result , thus efficiency of Vedic method for multiplication improves the performance this is the unique technique of computation based on 16 sutra(Methods). It also eliminates the unwanted multiplication step. This paper is based on Nikhlam sutra.*

Keywords:- *FPGA, Nikhlam sutra, RISC , VHDL, Vedic.*

I. INTRODUCTION

In today's technology, RISC Processors are playing a vital role RISC system shorten execution time by reducing the clock cycle per instruction and it can address enormous amount of memory upto 16 Exabyte and also the RISC with BIST feature is one of the more dominant test pattern which can provides, in system testing of the Circuit-Under-Test (CUT). BIST design is becoming more complicated with the increase of IC size. Though the RISC has less instruction set, as its the bit processing size increases then the test pattern becomes complicated and the structural faults are maintained high. And BIST is highly reliable, low cost. BIST is beneficial in many ways: First, it can reduce dependency on external Automatic Test Equipment (ATE). In addition, BIST can provide at speed, in system testing of the Circuit-Under-Test (CUT). This is crucial to the quality component of testing. Also, BIST can overcome pin limitations due to packaging, make efficient use of available extra chip area, and provide more detailed information about the faults present. In our thesis, a 64 bit RISC processor with limited functionality is designed with an architecture that supports BIST. The proposed design is done by implementing MICA (Minimal Instruction Set Computer Architecture) architecture. The design is implemented on Xilinx ISE 10.1i Simulator and programmed by using VHDL. The programmed code is supports FPGA Spartan-3E Kit. However, contemporary CAD tools allow the designer of hardwired control units almost as easy as micro programmed ones. This enables the single cycle rule to be enforced, while reducing transistor count. In order to facilitate the implementation of most instruction as register-to-register operations, ALU is analyzed and an exhaustive set of test patterns is developed. It has features like :

1) True 64-bit microprocessor,
2) High-performance microprocessor,
3) High level of integration, Low-power operation, Standard operating system support and Fully software compatible.
4) There register provide fast access to data during sequential program execution.
5) They can also be employed to reduce the overhead typically caused by passing parameters on the stack Instead of pulling parameters off of a stack, the subroutine is directed to use a subset of registers

II. MULTIPLICATION

Multiplication is Fundamental function in arithmetic operation and Digital signal processing and also multipliers are basic building block in Standard Digital Signal Processors (DSP) In Microprocessors, 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 throughput arithmetic operations are important to achieve the desired performance in many real time signal and image processing applications. Reducing the time delay and power consumption are extremely essential requirements for many applications**.** Multiplication method generally required a sequence of addition; subtraction and shift operations as it is complex operations form other arithmetic operations. Most modern computers directly support multiplication based operations in hardware.

III. VEDIC MATHEMATICS

The word „Vedic is derived from the word „veda which means the store-house of all knowledge. Mathematics, derived from the Veda, provides one line, mental and superfast methods along with quick cross checking systems. Vedic mathematics was rediscovered in the early twentieth century from ancient Indian sculptures (Vedas). What we call VEDIC MATHEMATICS is a mathematical elaboration of '**Sixteen Simple Mathematical formulae from the Vedas**' as brought out by **Sri Bharati Krishna Tirthaji**.

The conventional mathematical algorithms can be simplified and even optimized by the use of Vedic mathematics. The use of Vedic mathematics reduces the typical calculations in conventional mathematics to very simple one. The Vedic algorithms can be applied to arithmetic, trigonometry, plain and spherical geometry, calculus. The methodology of arithmetic rules that allow more efficient speed implementation. It also provides some effective algorithms which can be applied to various branches of engineering such as computing.

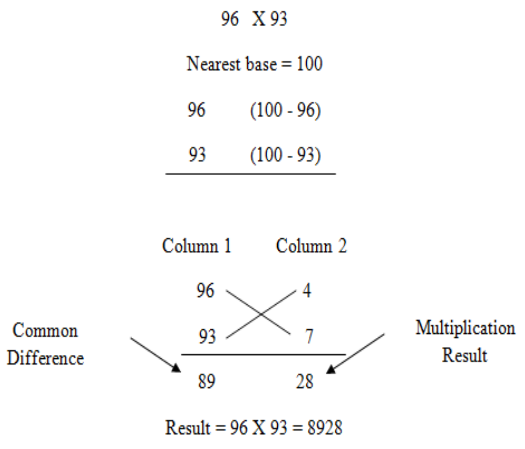
Vedic mathematics is mainly based on 16 Sutras (or aphorisms) dealing with various branches of mathematics like arithmetic, algebra, geometry etc. These Sutras along with their brief meanings are enlisted below alphabetically.

1) (Anurupye) Shunyamanyat – If one is in ratio, the other is zero.
2) Chalana-Kalanabyham – Differences and Similarities.
3) Ekadhikina Purvena – By one more than the previous One.
4) Ekanyunena Purvena – By one less than the previous one.
5) Gunakasamuchyah – The factors of the sum is equal to the sum of the factors.
6) Gunitasamuchyah – The product of the sum is equal to the sum of the product.
7) Nikhilam Navatashcaramam Dashatah – All from 9 and last from 10.
8) Paraavartya Yojayet – Transpose and adjust.
9) Puranapuranabyham – By the completion or noncompletion.
10) Sankalana- vyavakalanabhyam – By addition and by subtraction.
11) Shesanyankena Charamena – The remainders by the last digit.
12) Shunyam Saamyasamuccaye – When the sum is the same that sum is zero.
13) Sopaantyadvayamantyam – The ultimate and twice the penultimate.
14) Urdhva-tiryakbhyam – Vertically and crosswise.
15) Vyashtisamanstih – Part and Whole.
16) Yaavadunam – Whatever the extent of its deficiency.

Out of the above mentioned sutra, NIKHLLAM SUTRA is our intended for performing multiplication. These formulae can be used for the implementation and optimization of digital multipliers in the design of the digital systems possessing the multiplier blocks.

3.1 Nikhlam Sutra

The ―Nikhilam Navatascaram Dasatah literally means “All from Nine and the last from Ten”. The sutra basically means start from the left most digit and begin subtracting ‗9‘ from each of the digits; but subtract ‗10‘ from the last digit. The following example illustrates the way in which this Sutra could reduce the number of iterations to reduce the whole Multiplication. We first illustrate this Sutra by considering the multiplication of two decimal numbers (96 * 93) where the chosen base is 100 which is nearest to and greater than both these two numbers.



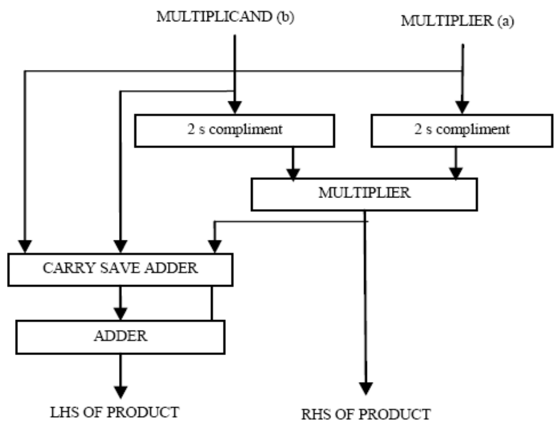


Fig.1: Multiplication Using Nikhilam Sutra **Fig.2:** Architecture of Vedic Multiplier using Nikhilam Sutra

The Block Diagram of Multiplier using Nikhilam Sutra is shown in Fig.2.The two inputs „a and „b represents 64-bit multiplier operand and 64-bit multiplicand operand respectively. The compliment of multiplicand or multiplier is represented by taking 2 s compliment of that number .So the 2 s complement block produces the complemented output of 64-bit multiplier operand (- a) and 64-bit multiplicand operand (-b). The complemented output of 64-bit multiplier operand and 64-bit multiplicand operand are two inputs to multiplier. The number of bits required in the right hand side of the product should have 64- bits irrespective of the number of bits in the product of compliments. So the surplus 64-bits of the right hand side part of the product is fed to one of the input of carry save adder for left hand part of the result. The left hand side part is implemented using 64-bit carry save adder. The negative of complemented multiplicand is implemented by taking output of the 2 scomplement block in the left hand side. The three 64-bits input operands of carry saver are 64-bit multiplier, the negative of complemented multiplicand and the surplus 64-bits of the right hand side part of the product. The two output of carry save adder i.e. sum vector and carry vector are inputs to binary adder block. The output of the adder represents left hand side of the answer.

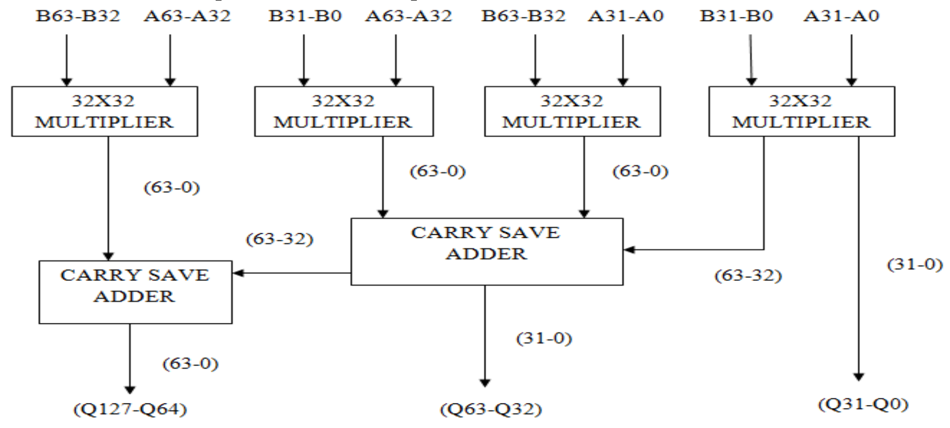


Fig.3: Architecture of 64x64 bits Vedic Multiplier: RESULT = (Q127-Q64) & (Q63-Q32) & (Q31-Q0)

Internal architecture of 64X64 bits Vedic multiplier is shown in Fig.3. Here 2's complement form of A and B is applied to 32X32 bit multipliers and the outputs of 32X32 bit multipliers are added accordingly to obtain the 128 bits final product. Thus, in the final stage two adders are also required.

IV. ADVANTAGES

Architecture of RISC Processor with Vedic multiplier based on speed Specification is designed here for following criteria

1) Increase the Speed of the system
2) To acquire good efficiency of the system

3) Reduce the time delay as well as path delay in the multiplier
4) The combinational path delay of Vedic multiplier obtained after compared with normal multipliers and found that the proposed Vedic multiplier.

V. SIMULATION RESULT

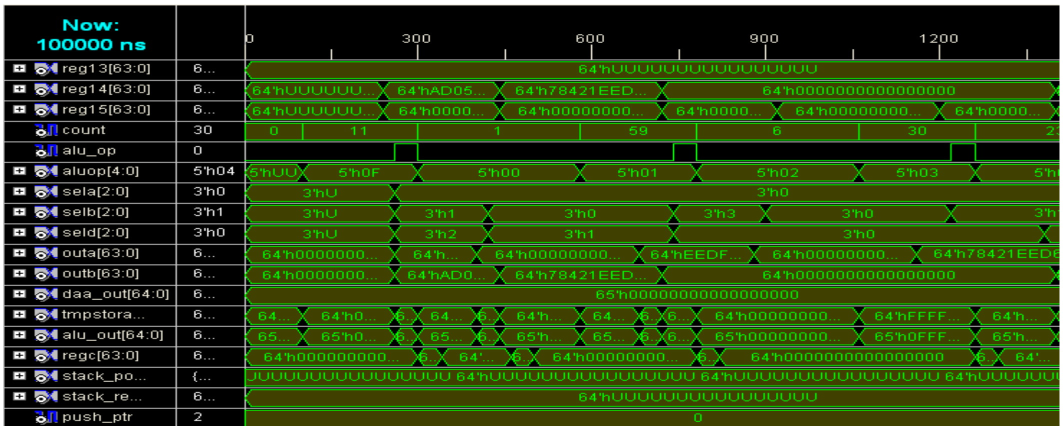


Fig4: Simulation results for the ALU outputs of 64 bit RISC Processor

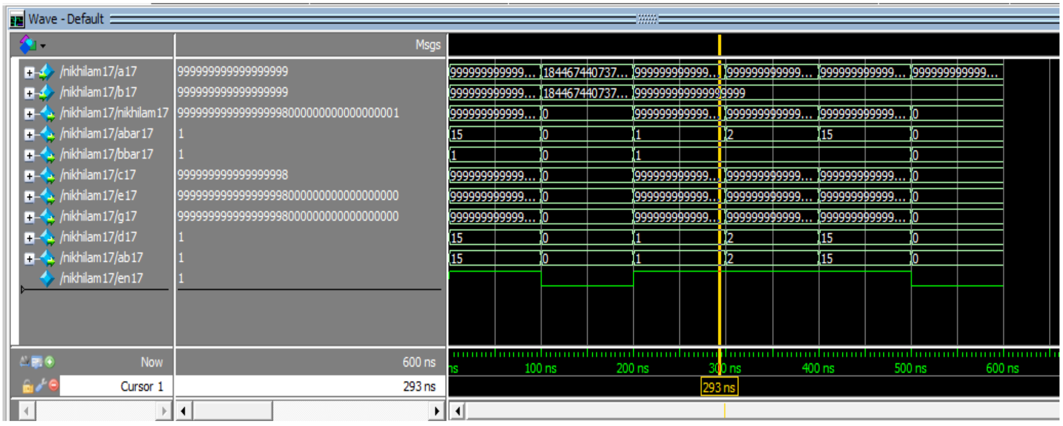


Fig.5: Simulation result of Proposed 64x64 Nikhlam multiplier

Here if both ‘a’ and ‘b’ are within the permissible limits of the Nikhilam condition, ‘en’ becomes ‘1’ and if ‘a’ and ‘b’ are outside Nikhilam limit, ‘en’ becomes ‘0’ and Nikhilam stops functioning (i.e.) “OFF”.

VI. CONCLUSION AND FUTURE SCOPE

The 64-bit RISC Processor with Vedic multiplier architecture has been designed and it can be implemented on FPGA. The design is verified on Xilinx ISE 10.1i simulator and programmed by using VHDL. The programmed code can be implemented on FPGA Spartan-3E Kit. ALU is analyzed and an exhaustive set of test patterns is developed. Here Vedic multiplier architecture is 135.05% faster than conventional multiplier which implies further reduction in time in multiplication operation.

Future work will be added by increasing the number of instructions and make a pipelined design with less clock cycles per instruction and includes the integration of the divider block, multiply and accumulate (MAC) unit, thereby making it into a Vedic Arithmetic and Logical unit (ALU)

REFERENCES

[1] Honey Durga Tiwari, Ganzorig Gankhuyag, Chan Mo Kim, Yong Beom Cho, *“Multiplier Design based on Ancient Indian Vedic Mathematics”, International SoC,Design Conference, Nov. 2008, pp.65-68.*
[2] Parth Mehta, *Dhanashri Gawali, “Conventional versus Vedic Mathematical Method for Hardware Implementation of a Multiplier”, 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, Trivandrum, Kerala, India, December 2009, pp. 640-642.*
[3]Jagadguru Swami, Sri Bharati Krishsna Tirthji Maharaja, *“Vedic Mathematics”, Motilal Banarsidas,*

Varanasi,India,1986, pp. 40-63

[4] Neeraj Mishra, Asmita Haveliya *An advancement in the N×N Multiplier Architecture Realization via the Ancient Indian Vedic Mathematics, International Journal of Electronics Communication and Computer Engineering IJECCE Volume 4, Issue 2*

[5] Ramalatha M, Thanushkodi K, Deena *Design and Implementation of a 64-bit RISC Processor using System On Chip (SOC), 2011, IJCSCN, 360-370 Design and Implementation of a 64-bit RISC Processor using VHDL, 2009 IEEE.*

[6]Shamim Akhter,"VHDL Imp lementation Of Fast NXN Multiplier Based On Vedic Mathematics", Jay p ee Institute of Information Technology University, Noida, 201307 UP, INDIA, 2007 IEEE.

[7] Atul Gupta*, The Power of Vedic Math, ISBN 81-7992-357-6, Jaico Publishing House,PhirozShah Mehta Road, Mumbai*