DESIGN OF HIGH SPEED MULTIPLIERS USING NIKHIALM SUTRA ALGORITHM

1.Babu Rao Kodavati 2.Tholada Appa Rao 3.Gollamudi Naveen Kumar

ABSTRACT:***This work is devoted for the design and FPGA implementation of a 16bit Multiplier, which uses Vedic Mathematics algorithms.In this thesis work we formulate this (VEDIC) mathematics for designing the multiplier architecture with two clear goals in mind such as: i) Simplicity and modularity multiplications for VLSI implementations and ii) Reduction in Area iii) Reduction in Delay.The implementation of the Vedic mathematics and their application to the multiplier ensure substantial reduction of propagation delay in comparison with DA based architecture and parallel adder based implementation which are most commonly used architectures. For arithmetic multiplication various Vedic multiplication techniques like Urdhvatiryakbhyam, Nikhilam, It has been found that Proposed multiplier Using Nikhilam Sutra is most efficient Sutra (Algorithm), giving minimum delay for multiplication of all types of numbers, either small or large.Further, the Verilog HDL coding of Urdhvatiryakbhyam Sutra for 16 bits, Nikhialm Sutra for 16 bit and Verilog code of proposed Multiplier Using Nikhilam Sutra for 16 bit, and the synthesize results shows that, combinational delay for Multiplier Using nikhilam Sutra is 23.751ns, and for Multiplier Using UrdhvaTiryagbhyam Sutra is 15.953ns, but whereas combinational delay for Proposed Multiplier Using nikhialm sutra is 13.118ns, which shows that there is 47% of Increment in the Combinational delay.***

I. INTRODUCTION:

There are number of techniques that to perform binary multiplication. In general, the choice is based upon factors such as latency, throughput, area, and design complexity. More efficient parallel approach uses some sort of array or tree of full adders to sum partial products. Array Multiplier, Booth Multiplier and Wallace Tree Multipliers are some of the standard approaches to have hardware implementation of binary, which are suitable for VLSI implementation at CMOS level.

Arithmetic is the oldest and most elementary branch of Mathematics. The name Arithmetic comes from the Greek word **άριθμός**(arithmos). Arithmetic is used by almost everyone, for tasks ranging from simple day to day work like counting to advanced science and business calculations. As a result, the need for a faster and efficient Multiplier in computers has been a topic of interest over decades. The work presented in this thesis, makes use of Vedic Mathematics and design a Vedic Multiplier. Multiplication basically is the mathematical operation of scaling one number by another. Talking about today's engineering world, multiplication based operations are some of the frequently used Functions, currently implemented in many Digital Signal Processing (DSP) applications such as Convolution, Fast Fourier Transform, filtering and in Arithmetic Logic Unit (ALU) of Microprocessors. Since multiplication is such a frequently used operation, it's necessary for a multiplier to be fast and power efficient and so, development of a fast and low power multiplier has been a subject of interest over decades.

Minimizing delay for digital systems involves optimization at all levels of the design. This optimization means choosing the optimum Algorithm for the situation, this being the highest level of design, then the circuit style, the topology and finally the technology used to implement the digital circuits. Depending upon the arrangement of the components, there are different types of multipliers available, Particular multiplier architecture is chosen based on the application.

Methods of multiplication have been documented in the Egyptian, Greek, Babylonian, Indus Valley and Chinese civilizations. In early days of Computers, multiplication was implemented generally with a sequence of addition, subtraction and shift operations. There exist many algorithms proposed in literature to perform multiplication, each offering different advantages and having trade off in terms of delay, circuit complexity, area occupied on chip and power consumption.

For multiplication algorithms performing in DSP applications, latency and throughput are two major concerns from delay perspective. Latency is the real delay of computing a function. Simply it's a measure of how long the inputs to a device are stable is the final result available on outputs. Throughput is the measure of how many multiplications can be performed in a given period of time. Multiplier is not only a high delay block but also a major source of power dissipation. So, it is of great interest to reduce the delay by using various optimization methods.

Two most common multiplication algorithms followed in the digital hardware are array multiplication algorithm and Booth multiplication algorithm. The computation time taken by the array multiplier is comparatively less because the partial products are calculated independently in parallel. The delay associated with the array multiplier is the time taken by the signals to propagate through the gates that form the multiplication array. Booth multiplication is another important multiplication algorithm. Large booth arrays are required for high speed multiplication and exponential operations which in turn require large partial sum and partial carry registers. Multiplication of two *n*-bit operands using a radix-4 booth recording multiplier requires approximately n/(2m) clock cycles to generate the least significant half of the final product, where *m* is the number of Booth recorder adder stages.

First of all, some ancient and basic multiplication algorithms have been discussed to explore Computer Arithmetic from a different point of view. Then some Indian Vedic Mathematics algorithms have been discussed. In general, for a multiplication of a n bit word with another n bit word, n2 multiplications are needed.To challenge this,"Urdhvatiryakbhyam Sutra" or "Vertically and Crosswise Algorithm" for multiplication has been discussed and then used to develop digital multiplier architecture. This looks quite similar to the popular array multiplier architecture. This Sutra shows how to handle multiplication of a larger number (N x N, of N bits each) by breaking it into smaller numbers of size (N/2 = n, say) and these smaller numbers can again be broken into smaller numbers (n/2 each) till we reach multiplicand size of (2 x 2).This work presents a systematic design methodology for fast and area efficient digit multiplier based on Vedic Mathematics.

II. Proposed Architecture of Vedic Multiplier

In order to reduce further delay and Area of the multiplier using Vedic mathematics, we are approaching for modified architecture. In this Proposed Multiplier we are using ADDER in the place of ADDER-SUBSTRACTOR.Hardware implementation of this mathematics is shown in Fig.1

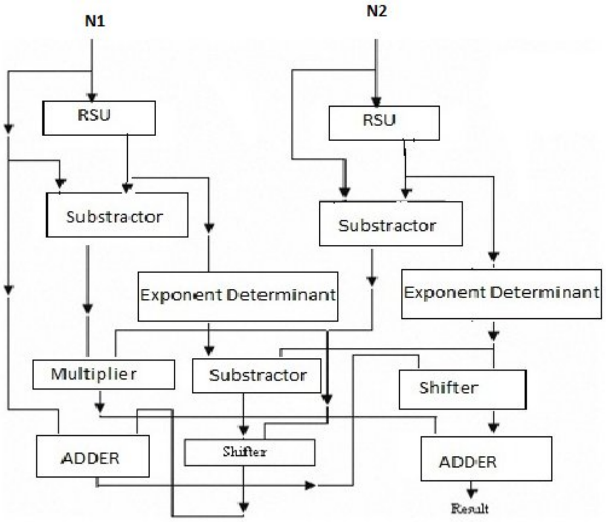


Fig .1 Proposed Multiplier Using Nikhilam Sutra of Vedic Mathematics

A. **Mathematical Formula:**Consider two numbers x1 and x2; k1 and k2 are exponent determinant of x1, x2 respectively.

x1=$2^{k1}+z1$

x2=$2^{k2}+z2$

Proposed multiplication can be as follows;

x1 * 2^{k1+k2} = $2^{k1}+z2\ 2^{k1-k2}$

x1 * x2 * 2^{k1-k2} = (2^{k1} +z1) ($2^{k1}+z2\ 2^{k1-k2}$)

=2^{2k1}+$z12^{k1}$+$z2\ 2^{2k1-k2}$+z1z2$2^{k1-k2}$

=$2^{k1}(2^{k1}+z1+\ z2\ 2^{k1-k2})$
+$z1\ z2\ 2^{k1-k2}$

=$2^{k1}(M+\ z2\ 2^{k1-k2})$ +$z1z2\ 2^{k1-k2}$

P= x1*x2 = $2^{k2}(M+\ z2\ 2^{k1-k2})$ + z1z2

B. Implementation of Proposed Radix Selection Unit:

The modified Radix selection unit will consist of two blocks (i) Exponent Determinant (ED), (ii) Shifter. Exponent determinant block will be same as in proposed Vedic multiplier. When a input no is applied to RSU block, first it will calculate the value of exponent determinant, the output will be given to the shifter, and other input to the shifter is (n+1) bit representation of '1', and shifter output will give you the value of Radix of a given number .For example if the output of the exponent determinant block is n, then the output of the shifter is 2^n, which will give the output of RSU Block.

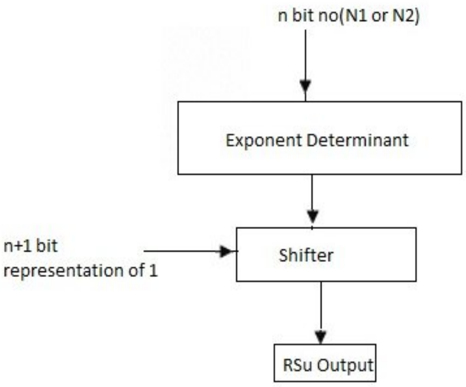
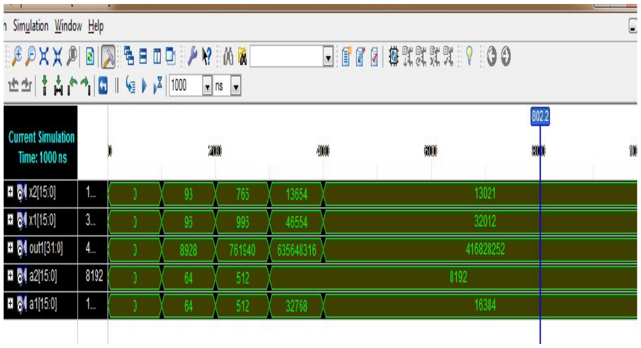


Fig 2:Proposed Radix Selection Unit

Consider if the output of the Exponent determinant is n, then the output of Radix Selection Unit will be 2^n, which is the output of the Proposed Radix selection Unit.

III.SYNTHESIS &SIMULATION RESULTS

A. Simulation Results of Proposed Multiplier



x2[15:0] 1... 0 93 765 13654 13021
x1[15:0] 3... 0 95 993 46554 32012
out1[31:0] 4... 0 8928 761940 635648316 416828252
a2[15:0] 8192 0 64 512 8192
a1[15:0] 1... 0 64 512 32768 16384

Fig 3: Simulation Waveform of Proposed Multiplier Using Nikhilam Sutra

B. Proposed Radix Selection Unit

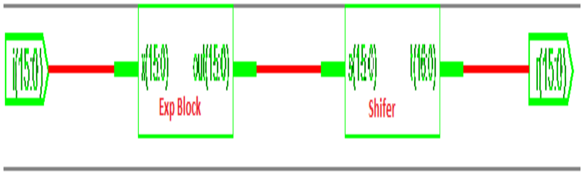


Fig 4: Block View of Proposed Radix Selection Unit

C. Description

i input data of 16bit to multiplier
r input data of 16 to bit multiplier
out output of 32 bit of Exponent Determinant
r output of the proposed RSU

RTL view of Exponent Determinant is same as in previous section

D. Device utilization summary:

Selected Device: Family: Vertex 5, Device : XC5VLX30

Package: FF324, Speed Grade: -3

Number of Slice LUT's	30 out of 19200	1%
Number of occupied Slices	16out of 4800	1%
Number of bonded IOBs	32out of 220	14%
Total equivalent gate count for design	196	

E. Proposed 16 bit Multiplier

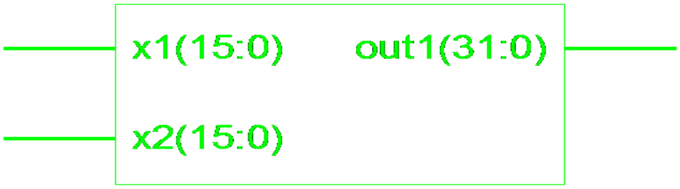


Fig 5: Block view of proposed 16 bit multiplier

F. Description

x1 input data of 16bit to multiplier
x2 input data of 16 to bit multiplier
out1 output of 32 bit of multiplier

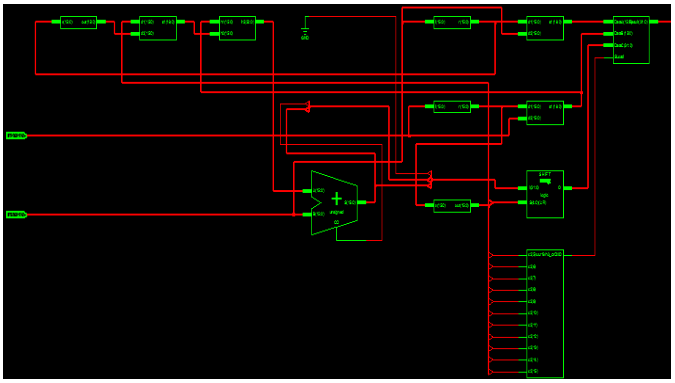


Fig 7: RTL View of Proposed 16 bit Multiplier

Device utilization summary:

Selected Device: Family: Vertex 5, Device : XC5VLX30,
Package: FF324, Speed Grade: -3

Number of Slice registers	1 out of 19200	1%
Number of Slice LUT's	217 out of 19200	1%
Number of occupied Slices	107out of 4800	2%
Number of bonded IOBs	64out of 220	29%
Number of DSP48Es	1 out of 32	3%
Total equivalent gate count for design		1812

IV: Results Comparison

Comparison of Vedic Multipliers

Vertex 5, Device : XC5VLX30, Package: FF324, Speed Grade: -3	Multiplier Using Urdhva-TiryagByham Sutra	MultiplierUsing Nikhima Sutra	**Proposed Multiplier Using Nikhilam Sutra**
Combinational Delay	15.953ns	23.751ns	**13.118ns**
No of LUT's(Out of 19200)	588	669	**217**
No of Gate Count	4116	7402	**1812**

Table.1:Comparison of Vedic Multipliers

Comparison of ProposedNikhilam Sutrawith Conventional Multipliers

Vertex 5, Device : XC5VLX30, Package: FF324, Speed Grade: -3	Shift and Add Multiplier	**Proposed Multiplier Using Nikhilam Sutra**
Combinational Delay	25.130ns	**13.118ns**
No of LUT's(Out of 19200)	428	**217**
No of Gate Count	2996	**1812**

Table.2: Comparison of Proposed Nikhilam Sutra with Conventional Multipliers

V.CONCLUSION

Conclusion:

The design of 16 bit Vedic multiplier Using Nikhilam Sutra, UrdhvaTiryagbhyam Sutra and 16 bit Proposed Vedic Multiplier Using Nikhilam Sutra has been realized on Vertex 5, Device : XC5VLX30, Package: FF324, Speed Grade: -3device .Thecombination delay For Vedic Multiplier Using Nikhilam Sutra is 23.751ns, for the Vedic Multiplier unit Using UrdhvaTiryagbhaym Sutra is 15.953 ns, which clearly show improvement in performance.In this thesis work we have implemented a Proposed Multiplier Using Nikhilam Sutra, for this combinational delay is 13.118 ns, which shows a great improvement in terms of delay about a 47% of increment in Speed. And there is a huge amount of decrease in the number of LUT's, when compared to Normal Vedic Multiplier Using Nikhialm Sutra.

Even though UrdhvaTiryakbhyam Sutra is fast and efficient but one fact is worth noticing, that is 2x2 multiplier being the basic building block of 4x4 multiplier and so on. This leads to generation of a large number of partial products and of course, large fanout for input signals a and b. To tackle this problem, a 4x4 multiplier can be formed using other fast multiplication algorithms possible, and keeping UrdhvaTiryakbhyam for higher order multiplier blocks.

In this work, some steps have been taken towards implementation of fast and efficient Multiplier using Vedic Mathematics, Using this Proposed Multiplier we can go for design of Arithmetic Logic Unit and MAC Unit in future. The idea of a very fast and efficient Multiplier using Vedic Mathematics algorithm is made real in future.

VI.REFERENCES

[1]. Honey Durga Tiwari, GanzorigGankhuyag, Chan Mo Kim, Yong BeomCho,“Multiplier Design based on Ancient Indian Vedic Mathematics”, International SoC,Design Conference, Nov. 2008, pp.65-68.

[2]. Parth Mehta, DhanashriGawali, “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]. M.Pradhan, R.Panda, “Design and Implementation of Vedic Multiplier”, A.M.S.E. Journal,SeriesD,Computer Science and Statistics, France ,vol.15,Issue2,pp.1-19,July 2010.

[4]. Jagadguru Swami Sri BharatiKrisnaTirthaji Maharaja, “Vedic mathematics”, MotilalBanarsidass Publishers Pvt. Ltd, Delhi, 2009.

[5]. HimanshuThapliyal and Hamid R. Arabnia, “A Time-Area- Power Efficient Multiplier and Square Architecture Based On Ancient Indian Vedic Mathematics”, Department of Computer Science, The University of Georgia, 415 Graduate Studies Research Center Athens, Georgia 30602-7404, U.S.A.

[6]. P. K. Saha, A. Banerjee, and A. Dandapat, "High Speed Low Power Complex Multiplier Design Using Parallel Adders and Subtractors," International Journal on Electronic and Electrical Engineering, (/JEEE), vol 07, no. II, pp 38-46, Dec. 2009.

[7]. Morris Mano, “Computer System Architecture”,PP. 346-347, 3rd edition,PHI. 1993.

[8]. Tam Anh Chu, “Booth Multiplier with Low Power High Performance Input Circuitary”, US Patent, 6.393.454 B1,May 21, 2002.

[9]. Ciminiera,L., and Serra,A.: " Arithmetic Array For Fast Inner Product Evaluation", Proceedings ofSixth Symposium On Computer Arithmetic, 1983, pp, 61-66.

[10].GensukeGoto,”High Speed Digital Parallel
Multiplier”, United States Patent-5,465,226,
November 7 1995.

[11]. A.P. Nicholas, K.R Williams, J. Pickles , “Application of Urdhava Sutra”, Spiritual Study Group, Roorkee (India),1984

[12]. A.P. Nicholas, K.R Williams, J. Pickles, “ Lectures on Vedic Mathematics”, Spiritual Study Group, Roorkee (India),1982.

[13]. Wey,C.L., and Chang, T.Y.: " Design and analysis of VlSI Based Parallel Multipliers",IEE Proc.E.1990,137,(4), pp. 328-336.

[14]. Shaunak Vaidya, Ashish Joglekar, and Vinayak Joshi, *“An Efficient Binary Multiplication Algorithm basedon Vedic Mathematics”*, Proceedings National Conference on Trends in Computing Technologies 2008, Pg 37-43

1.KODAVATI BABURAO,
Asst. Prof , Department of ECE,
Gandhi Institute of Engineering and Technology,
Gunupur,Rayagada
Orissa-765022.

2.T.APPARAO ,
Asso. Prof ,Department of EEE ,
Gandhi Institute of Engineering and Technology,
Gunupur,Rayagada
Orissa-765022.

3.G.NAVEEN KUMAR,
Asst. Prof ,Department of ECE,
Gandhi Institute of Engineering and Technology,
Gunupur,Rayagada
Orissa-765022.