FPGA Implementation of a high speed Vedic multiplier

[1] Tanya Wanchoo, [2] Vittal Niddodi Kamath, [3]Sudheendra Prabhu
[1]Bachelor of Engineering –Student, [2] Bachelor of Engineering – Student, [3]Faculty
Department of Electrical and Electronics
Manipal Institute of Technology, Manipal, India.
[1]tanyawanchoo@gmail.com, [2]vittal92@gmail.com, [3]sudhiprabhu@hotmail.com

***Abstract* - Since most of the important DSP algorithms such as Fast Fourier Transforms, Convolution etc incorporate complex multiplication computations, the overall time utilized is high. In the proposed design we have implemented a vedic multiplier using the 'Urdhva Tiryagbhyam' and 'Nikhilam Navatashcaramam Dashatah' sutras, resulting in the reduction of delay in the multiplier thereby increasing the overall speed of the system.**
HDL designer version 2012.1 has been used for coding which is synthesized using Xilinx device Spartan-3 MXS3FK-PQ208 FPGA. Results indicate 14% reduction in time delay when compared to a conventional multiplier.

***Index Terms* - Vedic Mathematics, Urdhva Tiryagbhyam Sutra, Carry-skip technique**

I. INTRODUCTION

High speed arithmetic operations are very important in many signal processing applications. Speed of the digital signal processor (DSP) is largely determined by the speed of its multipliers. In fact the multipliers are the most important part of all digital signal processors [2]; they are very important in realizing many important functions such as Fast Fourier transforms and convolutions. Since a processor spends a considerable amount of time performing multiplication, improvement in multiplication speed can improve the system performance.

Vedic mathematics is derived from the ancient "Vedas". It is based on the sixteen word-formulae which are termed as "sutras" [4]. This field presents certain multiplication techniques which when integrated with the multiplier design enhance the speed of the multiplication operation [1].The importance of Vedic mathematics lies in the fact that it reduces the typical calculations in conventional mathematics to very simple ones [1]. This is so because the Vedic formulae are claimed to be based on the natural principles on which the human mind works [4]. Vedic Mathematics is a methodology of arithmetic rules that allow more efficient speed implementation.

"Urdhva Tiryagbhyam" and "Nikhilam Navatashcaramam Dashatah " are the two sutras in Vedic mathematics involved in multiplication [4].These Sutras (algorithms) have been traditionally used for the multiplication of two numbers in the decimal number system. The same idea, mapped to the binary base, makes the algorithm compatible with the digital hardware [2].The partial product generation and additions are done concurrently in a Vedic multiplier, making it adaptable to parallel processing [1], which in turn reduces delay

II. LITERATURE REVIEW

A. Nikhilam Navatashcaramam Dashatah

The Nikhilam Sutra literally means "all from 9 and last from 10" [4]. It is more efficient when the numbers involved are large. The Nikhilam Sutra algorithm is efficient for multiplication only when the magnitudes of both operands are more than half their maximum values. For n-bit numbers, therefore both operands must be larger than 2^{n-1} [3]. Nikhilam Sutra is explained by considering the multiplication of two single digit decimal numbers 8 and 7 where the chosen base is 10 which is nearest to and greater than both these two numbers.

Nearest base = 10	
Column 1	Column 2
8	10-8=2
7	10-7=3
Common difference = 5	Multiplication of compliments= 6

1) Nikhilam Sutra in Binary Number

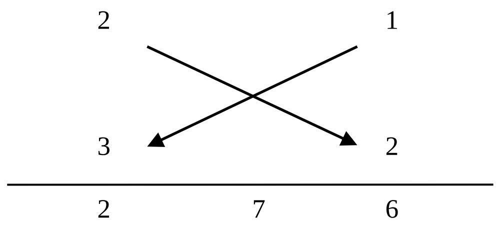
The Nikhilam sutra can also be applicable to binary number system. The compliment of multiplicand or multiplier is represented by taking 2's compliment of that number[3]. For an eight-bit Vedic multiplier, the right hand side part of the product is implemented using 8X8 bit multiplication. The left hand side part is implemented using 8-bit carry save adder. Hence the multiplication of two 8-bit numbers is reduced to the multiplication of their compliments and addition [3].

B) Urdhva Tiryagbhyam Sutra

The "Urdhva Tiryagbhyam‟ Sutra is a general multiplication formula applicable to all cases of multiplication. "Urdhva‟ and "Tiryagbhyam‟ words are derived from Sanskrit literature. "Urdhva‟ means "Vertically" and "Tiryagbhyam‟ means "crosswise" [4].

The multiplication of two 2-digit decimal numbers 21 and 32 is shown below. The least significant digit 1 of multiplicand is multiplied vertically by least significant digit 2 of the multiplier, get their product 2 and set it down as the least significant part of the answer. Then 2 and 2, 1 and 3 are multiplied crosswise, add the two, get 7 as the sum and set it down as the middle part of the answer. Then 2 and 3 is multiplied vertically, get 6 as their product and put it down as the last the left hand most part of the answer [1,4].

21 x 32 =276



1) Urdhva Tiryagbhyam Sutra in Binary Number

The "Urdhva Tiryagbhyam‟ algorithm can be implemented for binary number system in the same way as decimal number system. Let us consider the multiplication of two 2-bit binary numbers a1a0 and b1b0. Assuming that the result of this multiplication would be 4 bits, we express it as p2 p1 p0. The least significant bit a0 of multiplicand is multiplied vertically by least significant bit b0 of the multiplier, get their product p0 and set it down as the least significant part of the answer (p0). Then a1 and b0, and a0 and b1 are multiplied crosswise, add the two, get sum1 and carry1, the sum bit is the middle part of the answer (p1). Then a1 and b1 is multiplied vertically, and add with the previous carry (carry1) and get p2 (2 bit) as their product and put it downs as the left hand most part of the answer (p2). So a1a0 X b1b0=p2 p1 p0 [2]. Similarly the 2×2 Vedic multiplier module is then used to implement higher level multipliers [1,2].

III. ARCHITECTURE

A. Architecture of 2x2 Urdhva Multiplier

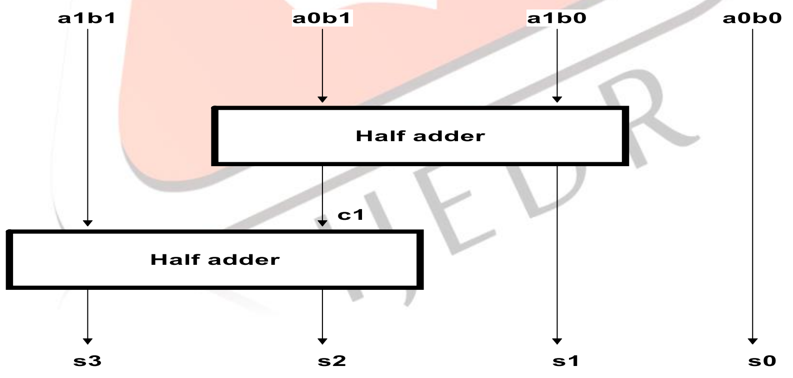


Fig 1. 2x2 Urdhva Multiplier

To elaborate this multiplication algorithm, let A and B be the two bit numbers to be multiplied. Initially, multiplication (AND) operation of the two least significant bits is performed (a0b0). This is the least bit of the final product. In the next step the least significant bit of the multiplicand is multiplied with the next higher bit of the multiplier generating a partial product, the product of Least significant bit of multiplier and next higher bit of the multiplicand is also calculated, thus generating one more partial product. These two partial products are added. This generates a sum and a carry. The sum gives second bit of the final product and the carry is added with the partial product obtained by multiplying the most significant bits. This again generates a sum and a carry the sum and the carry corresponds to the third and fourth bit of the final product.

The 2x2 multiplier uses two half adders and four AND gates.

B) Architecture of 4x4 Urdhva Multiplier

The 4x4 multiplier module consists of four 2x2 multiplier modules, a carry save adder and a ripple carry adder. The two LSBs of the right most multiplier (m0 (0) and m0 (1)) directly corresponds to last two bits of the final product. Travelling down the architecture, it is evident that these partial products are inputs to the carry save adder and the ripple carry adder. The carry save adder helps in adding three input variables. The addition is done by separate calculation, which results in an array of sum and carry. These arrays are added by padding a zero to the carry and sum arrays to the right and left accordingly resulting in a 5-bit sum. The last two bits of the five-bit adder output correspond to the next two higher bits of the final product. The architecture progresses further, where the first three MSBs of the five bit adder output with a padded zero on the right acts as one of the input to the four-bit ripple carry adder. The four bit output (m3) from the first 2x2 multiplier is the other input to the RC adder. The eight bit output corresponds to the most significant eight bits of the final product.

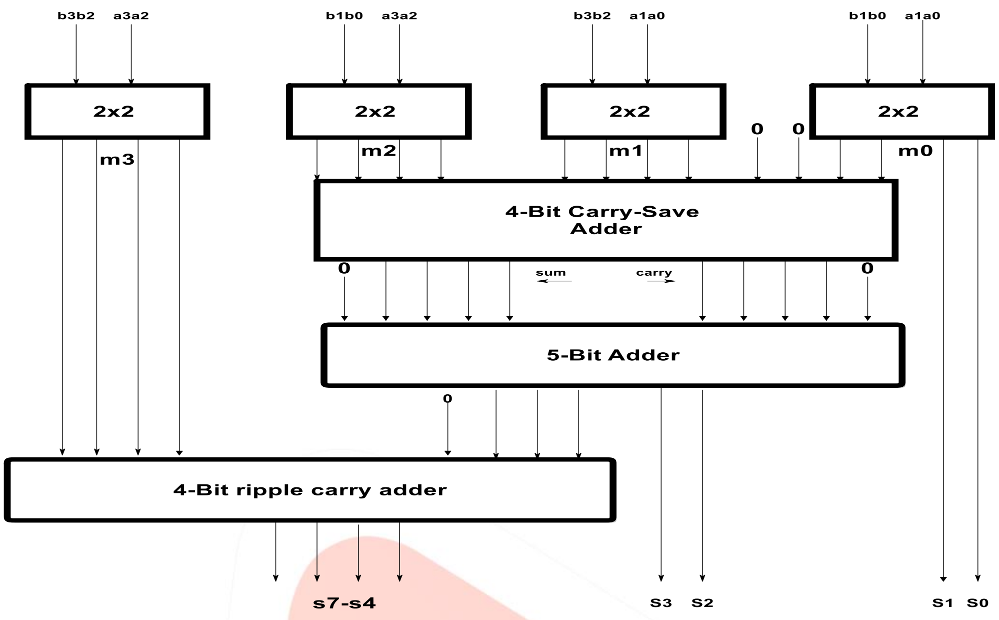


Fig 2. 4x4 Urdhva Multiplier

C) *Architecture of 8x8 Urdhva Multiplier*

The 8x8 multiplier architecture is implemented using 4x4 multipliers as components, which in-turn uses 2x2 multipliers as the basic components. The basic idea is to divide the eight bit numbers (multiplicand and multiplier) into four-bit components. These four bit components are passed on to the four-bit multiplier where the four-bits are divided into two-bit components. These two-bit components are passed on to a basic 2x2 multiplier. The general architecture of the 8x8 and 4x4 multiplier are very similar. The only difference lies in the number of bits of the adder. In the former, a four bit carry-save adder, a five-bit adder and a 4-bit ripple carry adder were used. An eight bit carry-save adder, a nine-bit adder and an eight-bit ripple carry adder are used in the latter. A and B are eight bit numbers which are to be multiplied.

A= a1 a2 a3 a4 a5 a6 a7   B= b0 b1 b2 b3 b4 b5 b6 b7

Four blocks, which consist of four bit inputs, are built.

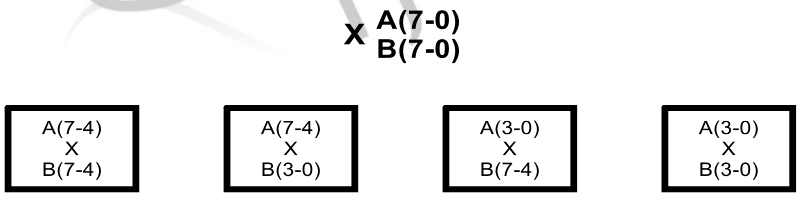


Figure4: 4x4 components in 8x8

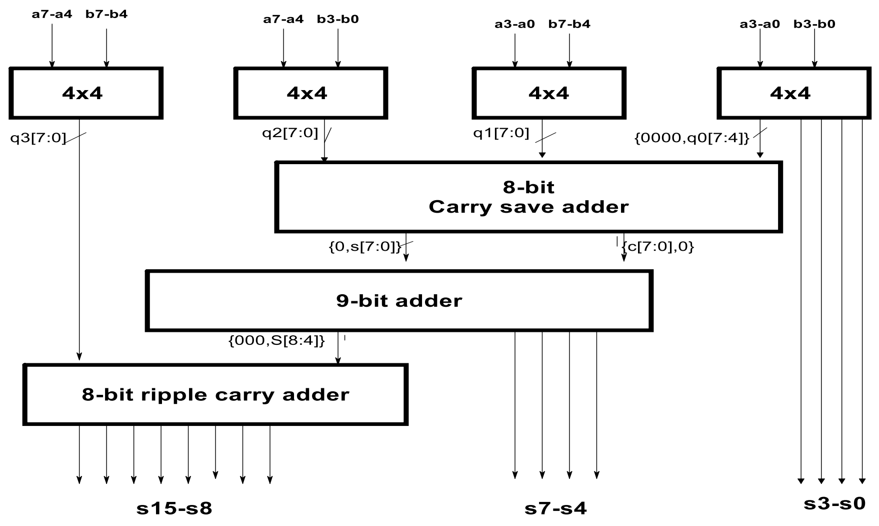


Fig 3. 8x8 Urdhva Multiplier

D) Architecture of 16x16 Urdhva Multiplier

The 16x16 multiplier is built using four 8x8 multiplier modules which are further developed using four 4x4 multipliers. One sixteen-bit carry-save adder and one sixteen-bit ripple carry is used in developing the same. One seventeen-bit adder is also used to add the sum and the carry arrays which are the intermediate results of the carry-save adder.

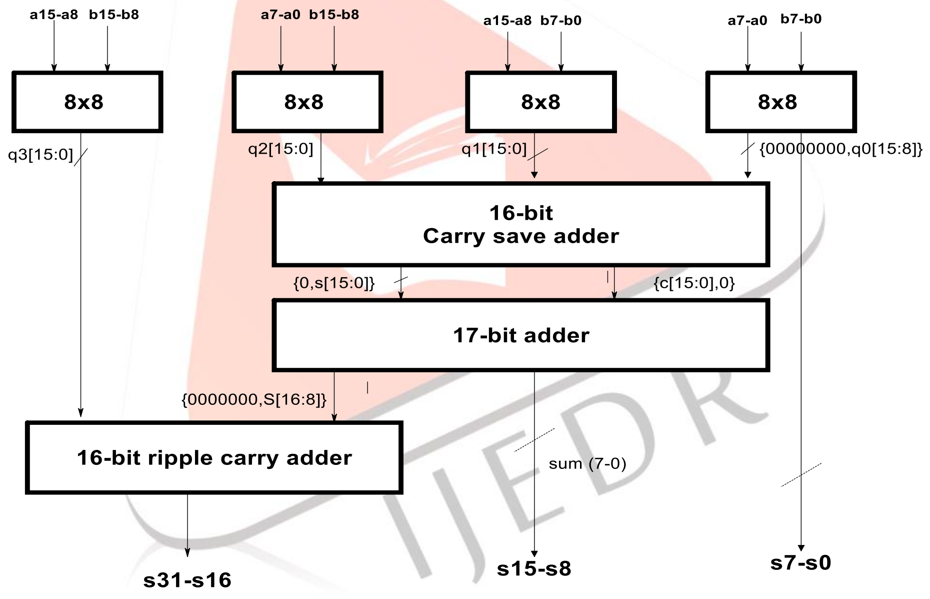


Fig 4. 16x16 Urdhva Multiplier

E) Architecture of 16x16 Nikhilam Multiplier

The Nikhilam algorithm calculates the deviation of a number from a prescribed base. The deviations, taken as residues are multiplied forming the lower significant part of the product. Then, the multiplicand or multiplier is added with the cross residue to form the upper significant part of the product. The second operation can be subtraction accounting to the fact that the deviation may be negative.

The two's compliment of the two numbers (multiplicand and multiplier) is calculated to find the deviation. The two residues (deviations) are multiplied using an eight bit urdhva multiplier. The least eight bits of the sixteen bit product corresponds to the least eight bits of the final product. The most significant bits of the product are given to the carry save adder, the other inputs being the multiplicand and the multiplier. The output of the adder corresponds to the higher eight bit of the final product. According to the architecture shown in Fig 5, when the inputs are higher bit numbers, their twos compliments are lower bit numbers which are in-turn the inputs to the intermediate multiplier. The delay associated with the multiplier for low-bit number is relatively lesser than the one that is associated with the higher bit number. Hence, the multiplier which is implemented using the "Nikhilam" sutra is suitable for higher bit numbers.

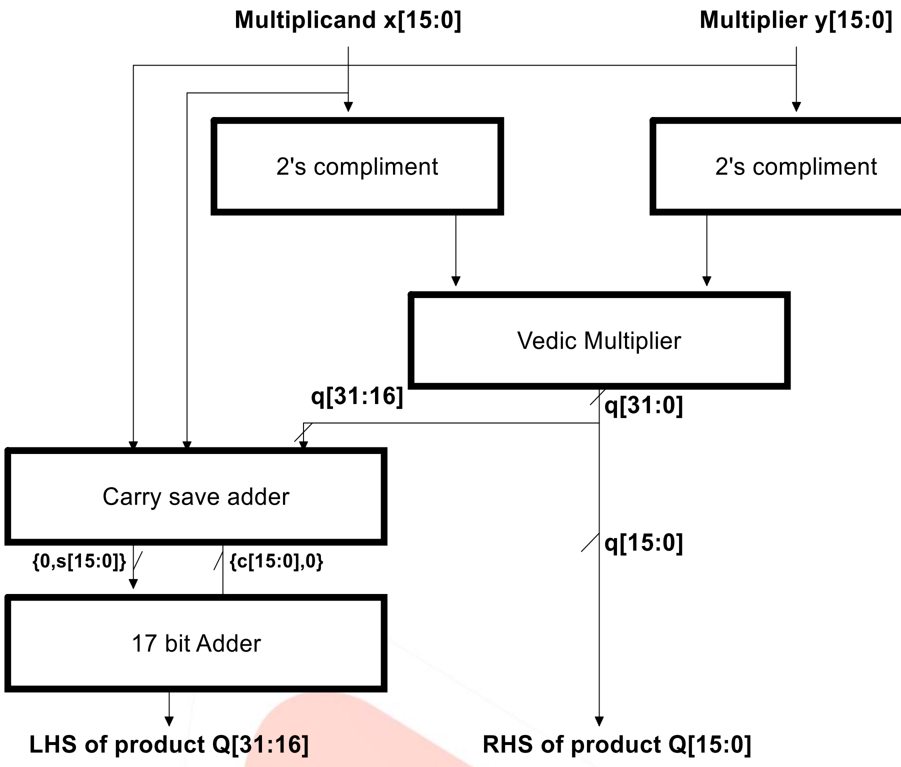


Fig 5. 16x16 Nikhilam Multiplier

*F) Architecture of a 16x16 Conventional Multiplier*T

The conventional multiplier developed for an extensive comparison between the same and the Vedic multiplier is the Array multiplier. Multiplier circuit is based on add and shift algorithm. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. The partial product are shifted according to their bit orders and then added.
The flowchart is shown in fig 6

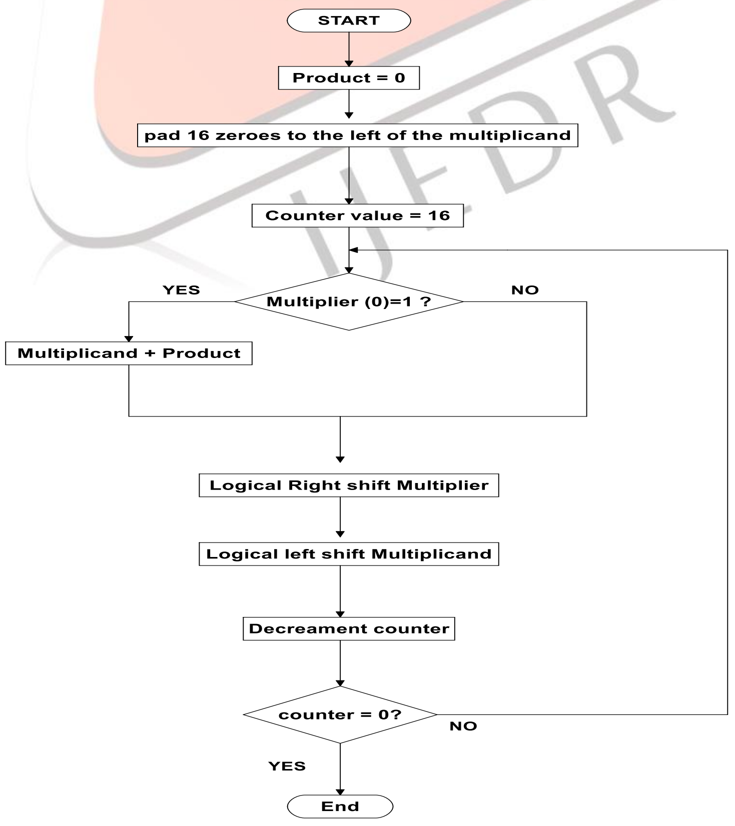


Fig 6. Array Multiplier

IV. Results

A. Sixteen Bit Array Multiplier:

1) Simulation

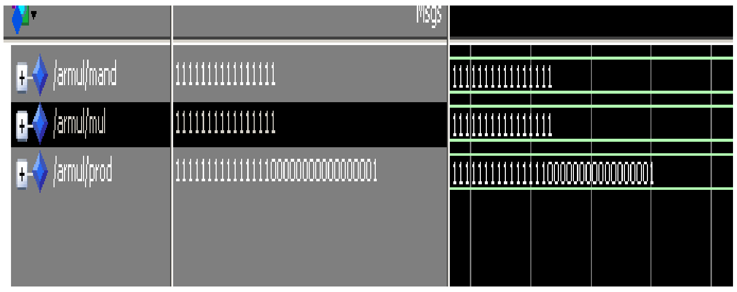
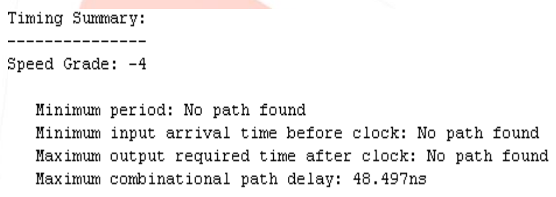


Fig 7. Simulation of 16x16 Array Multiplier

Description:
Mand: Multiplicand (16 bit data)
Mul: Multiplier (16 bit data)
Prod: Product (32 bit data

2) Timing report



```
Timing Summary:
---------------
Speed Grade: -4

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 48.497ns
```

Fig 8. Timing Report of 16x16 Array Multiplier

Observation: Maximum combinational path delay: 48.497ns

B. Sixteen bit multiplier using Urdhva Tiryagbhyam

1) Simulation:

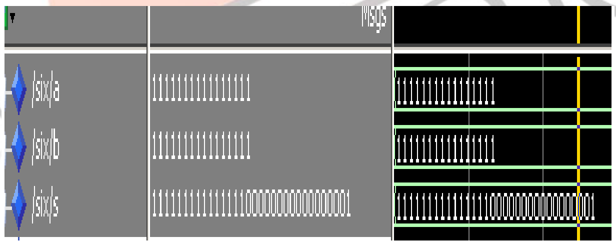
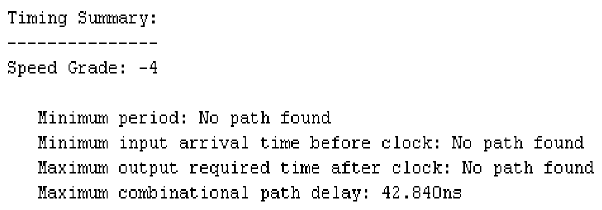


Fig 9. Simulation of 16x16 Urdhva Multiplier

Description:
a,b: 16-bit inputs (multiplicand and multiplier)
s: 32 bit product

2) Timing report



```
Timing Summary:
---------------
Speed Grade: -4

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 42.840ns
```

Fig 10. Timing Report of 16x16 Urdhva Multiplier

Observation: Maximum combinational path delay: 42.840ns

C. Sixteen bit Multiplier using "Nikhilam" sutra

1) Simulation :

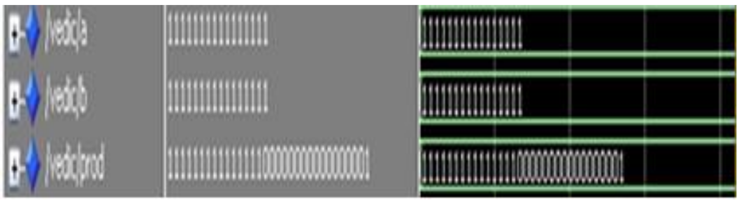
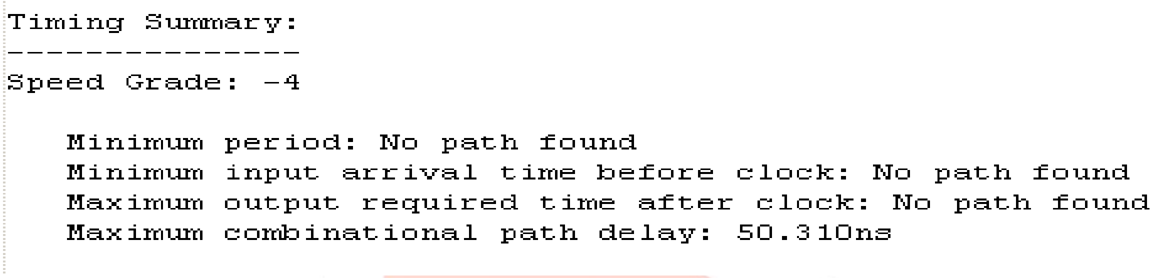


Fig 11. Simulation of 16x16 Nikhilam Multiplier

Description:
a,b: 16-bit inputs (multiplier and multiplicand)
prod: 31-bit output

2) Timing report:



```
Timing Summary:
---------------
Speed Grade: -4

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 50.310ns
```

Fig 12. Timing Report of 16x16 Nikhilam Multiplier

V. GRAPHICAL ANALYSIS OF TIME DELAY

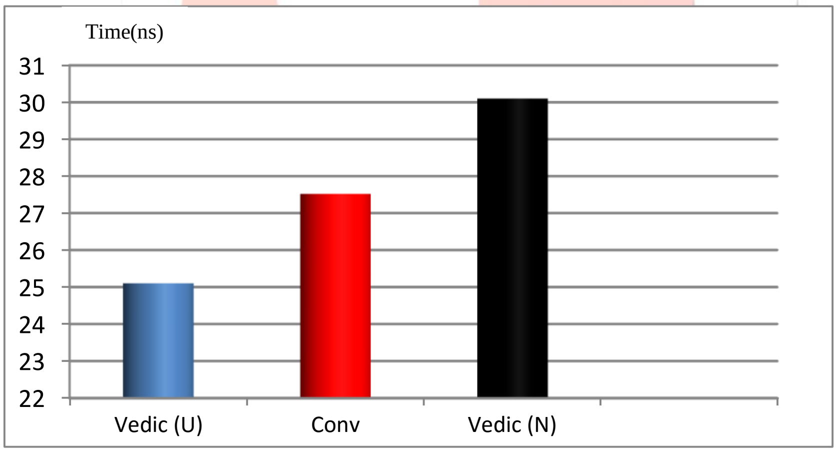


Fig 13. Graphical Comparison of time delays of 8x8 Multipliers

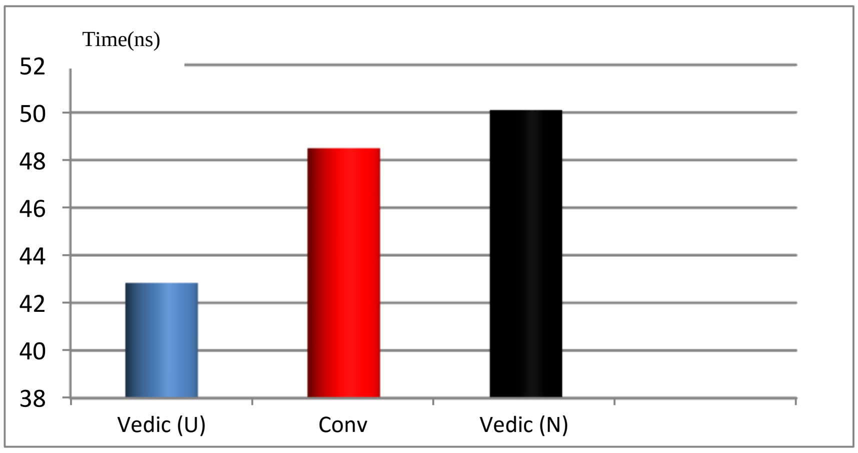


Fig 14. Graphical Comparison of time delays of 16x16 Multipliers

Comparing the Vedic and conventional multiplier, the percentage reduction in delay in an 8x8 multiplier is 9% and that in 16x16 multiplier is 14%. This shows that the percentage reduction in delay increases for higher bit numbers.

VI. CONCLUSIONS

Type	8x8 Multiplier	16x16 Multiplier
Conventional Multiplier	27.526 ns	48.497 ns
Urdhva Tiryaghbhyam	25.170 ns	42.840 ns
Nikhilam Navatashcharamam Dashatah	31.038 ns	50.310 ns

Table 1 Comparison of time delays of various Multiplier

Type	% slices used	% LUTs used	% IOBs used
Conventional Multiplier	7	7	45
Urdhva Tiryaghbhyam	10	9	45
Nikhilam Navatashcharamam Dashatah	13	11	45

Table 2 Area reports of various Multipliers

The proposed Vedic multiplier architecture shows speed improvements over the conventional multiplier architecture presented. The 16x16 vedic multiplier using the "urdhva tiryakbhyam" sutra is considerably faster than the conventional multiplier.Also, The 16x16 Vedic multiplier using " Nikhilam" is suitable and better than the rest in terms of speed when magnitude of both operands are more than half of their maximum values . This approach may be well suited for multiplication of numbers with more than 16 bit size. In many processors where delay is a major constraint, conventional multipliers can be replaced by high speed vedic multipliers for better and efficient high speed operation.

REFERENCES

Journal / Conference Papers

[1] Pushpalata Verma, K.K.Mehta "Implementation of an effecient multiplier based on vedic Mathematics using EDA tool" International Journal of Engineering and advanced technology(IJEAT) ISSN: 2249-8958,volume-1,Issue-5,June 2012.

[2] S.S.Kerur, Prakash Narchi,Jayashree C N, Harish M Kittur and Girish V.A "Implementation Of vedic multiplier for Digital Signal Processing",International Conference on VLSI ,Communication and Instrumentation (ICVCI) 2011, proceedings published by international computer applications (IJCA),pp.1-6.

[3] R. Sridevi, Anirudh Palakurthi, Akhila Sadhula, Hafsa Mahreen Design of a High Speed Multiplier (Ancient Vedic Mathematics Approach) International Journal of Engineering Research (ISSN : 2319-6890) Volume No.2, Issue No.3, pp : 183-186

Reference / Hand Books

[4] J. S. S. B. K. T. Maharaja, Vedic mathematics, Delhi: Motilal Banarsidass Publishers Private Limited.

[5] Stephen Brown and Zvonko Vranesic "Fundamentals of Digital logic Design with VHDL", Second edition, Tata McGraw-Hill publishing company limited .

Web

[6] Vedic Mathematics, www.vedicmaths.org

[7] Multipliers, www2.engr.arizona.edu.