International Journal of Engineering Research (ISSN : 2319-6890)
Volume No.2, Issue No.3, pp : 183-186 01 July 2013

# Design of a High Speed Multiplier (Ancient Vedic Mathematics Approach)

**R. Sridevi, Anirudh Palakurthi, Akhila Sadhula, Hafsa Mahreen**

Department of Electronics and Communication Engineering, KITS Warangal
Email : rsrideviprasad@yahoo.co.in,anirudh.palakurthi@yahoo.com,akila.sadula@gmail.com
hafsamahreen@gmail.com

## Abstract

***In this paper, an area efficient multiplier architecture is presented. The architecture is based on Ancient algorithms of the Vedas, propounded in the Vedic Mathematics scripture of Sri Bharati Krishna Tirthaji Maharaja. The multiplication algorithm used here is called Nikhilam Navatascaramam Dasatah. The multiplier based on the ancient technique is compared with the modern multiplier to highlight the speed and power superiority of the Vedic Multipliers.***

**Key Words:** Digital Multiplier, Nikhilam algorithm.

## 1. Introduction

The multiplier is one of the fundamental hardware blocks in many Digital Signal Processing systems for performing different operations like frequency domain filtering(FIR,IIR),frequency transformations(FFT), etc., Some of the important arithmetic functions implemented by the multiplier in the DSPs are Multiply and Accumulate (MAC), inner product. Not just in the DSP systems, the digital multiplier is an indispensable block in Digital Image Processing systems, and even in Microprocessor in its ALU.

The former microprocessors did not have a Multiplier block, instead of which they used multiply routines, for shifting and adding the partial results to produce the final product result. But with the enhanced levels of integration in the latest VLSI circuits day-by-day, the task of designing a multiplier block has began receiving immense devotion in the design of digital systems.

The multiplier, being the most significant block in many such digital systems, their speed and efficiency are primarily dependent upon the speed, area, throughput efficiency of the multipliers implemented in these systems. The other feature of the multiplier which has to be given quantitative concern in designing of the systems is Power Dissipation, viz. the multiplier is a source of high power dissipation.

Consequently, many algorithms have been suggested in different literatures aiming at improvising any one or more of the characteristics-speed, area, throughput, power of the digital multiplier. The Booth Multiplier, CSA array method, Wallace tree method, and the Booth recording multiplier are some of the important architectures proposed to improvise the digital multiplier.

In this paper, a high performance, high throughput and area efficient architecture of a multiplier for the Field Programmable Gate Array (FPGAs) is being proposed. The crucial aspect of this proposed architecture is that it is based on an Ancient Indian Vedic Mathematics. This paper gives information of "Nikhilam Sutra" which can increase the speed of multiplier by reducing the number of iterations. Vedic Mathematics also suggests one more formula for multiplication i.e. "Urdhva Tiryagbhyam" which is utilized for multiplication to improve the speed, area parameters of multipliers.

## 2. Conceptual Overview

### 2.1. Vedic Mathematics

Veda, by definition, is ‘knowledge’. The Vedic Math has a much ancient origin though attributed to the techniques rediscovered between 1911-1918, by Sri Bharati Krshna Tirthaji Maharaja. Vedic mathematics is the ancient system of mathematics, or, precisely, it is a distinct technique of calculations based on simple rules and principles with which any mathematical problem can be solved, whether it may be arithmetic, algebra, geometry, trigonometry or even calculus.

The Vedic mathematics is a coherent collective combination of 16 Sutras(Formulae) and 16 Sub-Sutras(the corollaries of the formulae). According to a theory, “The sutras of Vedic Mathematics are the software for the cosmic computer that runs this universe.”

The calibre of Vedic mathematics lies in the fact that it scales down the otherwise cumbersome-looking calculations in conventional mathematics to a very elementary one. This is so because the Vedic formulae are claimed to be established on the natural principles on which the human mind functions.

Vedic Mathematics holds two sutras (Urdhva Tiryagbhyam and Nikhilam Navatascaramam Dasatah) and one sub-sutra (Anurupyena) 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.

## 2.2. Nikhilam 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.

To multiply 92 and 89. Apply Nikhilam Sutra – “All from nine and last from ten” on both the numbers —

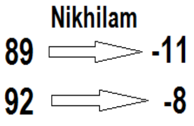


Figure 1

The arrows in Figure 2 indicate the operation of the Nikhilam Sutra being performed, viz. the subtraction of 10 from the last digit and 9’s from all the other digits starting with the leftmost digit.

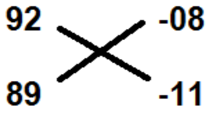
- Now we write this down side-by-side,

$$\begin{array}{cc} 92 & -08 \\ 89 & -11 \\ \hline \\ \hline \end{array}$$

- Multiply (-08) and (-11) to get ‘88’.

$$\begin{array}{ccc} 92 & -08 & \\ 89 & -11 & \times \\ \hline & 88 & \\ \hline \end{array}$$

- Now we cross-add. This is done by both “adding 92 and -11 to get 81”or “adding 89 and –08 to get 81.”



- Note that in both operations you get the same answer that is ‘81’ which is written below to get the solution.

$$\begin{array}{ccc} & 92 & -08 \\ \times & & \\ & 89 & -11 \\ \hline & 81 & 88 \\ \hline \end{array}$$

This technique works very well if the numbers to be multiplied are near a base. Upon little alteration, this also works very well for any pair of numbers.

After this illustration, we now discuss the operational principle of Nikhilam Sutra by taking the case of multiplication of two n–bit numbers x and y having complements $\overline{\boldsymbol{x}} = \boldsymbol{10^n - x}$ and $\overline{\boldsymbol{y}} = \boldsymbol{10^n - y}$ respectively.

The required product ‘*p*’ is defined as:

$$\mathbf{p = xy} \qquad \dots \qquad \mathbf{(1)}$$

which can be reframed by adding and subtracting $\boldsymbol{10^{2n} + 10^n(x + y)}$ to the right hand side as:

$$\boldsymbol{p = xy + 10^{2n} - 10^{2n} + 10^n(x + y) - 10^n(x + y)} \qquad \dots \qquad \mathbf{(2)}$$

The above terms can be clubbed as follows:

$$\begin{aligned} \boldsymbol{p} &= \boldsymbol{\{10^n(x + y) - 10^{2n}\}} \\ &\quad \boldsymbol{+ \{10^{2n} - 10^n(x + y) + xy\}} \\ &= \boldsymbol{10^n\{(x + y) - 10^n\} + \{(10^n - x)(10^n - y)\}} \\ &= \boldsymbol{10^n\{x - \overline{y}\} + \{\overline{xy}\} = 10^n\{y - \overline{x}\} + \{\overline{xy}\}} \end{aligned} \qquad \dots \qquad \mathbf{(3)}$$

From (3), the expressions of LHS and RHS can be deduced, which come out to be:

$$\boldsymbol{LHS = \{x - \overline{y}\} = \{y - \overline{x}\}} \qquad \dots \quad \mathbf{(4)}$$

$$\boldsymbol{RHS = \{\overline{xy}\}} \qquad \dots \quad \mathbf{(5)}$$

Hence the multiplication of two n- bit numbers is reduced to the multiplication of their complements. To take full advantage of this reduction, it should be ensured that the numbers obtained after taking the complements are lesser than the original numbers.

This condition is satisfied if both the original numbers are greater than 10n=2, i.e., x > 10n=2 and y > 10n=2.

This is the reason why it is said that the Nikhilam Sutra is efficacious in the multiplication of large numbers than the smaller ones.

## 3. Detailed Design (Proposed Architecture)

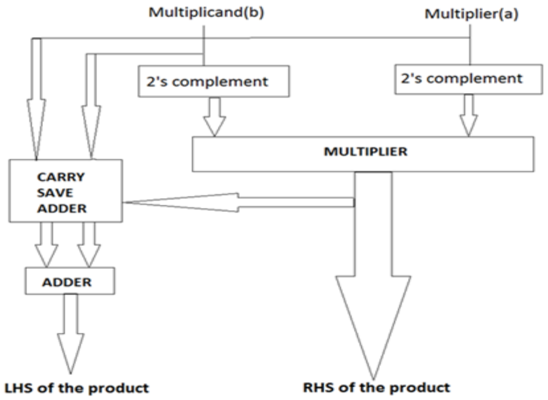


Figure 2

## 3.1. Top Module:

The block diagram for proposed multiplier is shown in the figure2. As we are using binary numbers in digital signal processing applications we have implemented for binary system. The multiplication can be done using the complementer, CSA adder and adder. The RHS of the product can be obtained by the multiplication of complimented outputs of multiplier and multiplicand and the LHS of the product can be obtained by addition using a CSA.
This can be used for the multiplication of any number of bits. In this paper, we have presented a 4x4 architecture applying the Nikhilam Algorithm. In the figure 2, the two inputs a and b represents the 4 bit multiplier and 4 bit multiplicand respectively.

## 3.2. Internal blocks:

### 3.2.1 2’s complementer:

The multiplicand and the multiplier are given as inputs to the two 2’s complementer blocks. The logic implementation of the 4 bit 2’s complementer is presented in figure3.

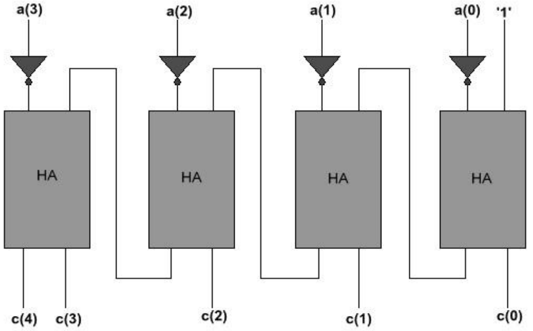


**Figure3**

In figure3, the “HA” represents a Half Adder block.

### 3.2.2 Multiplier:

Now, the complemented output of multiplier (-a) and the complemented multiplicand (-b) are then produced. These complemented outputs of multiplier and multiplicand are given as inputs to the multiplier block. The 4x4 multiplier architecture that we employed is based on calling a 2x2 multiplier so as to ease the multiplication procedure. This implementation is represented by the following figure 4.
Here, a and b are the 2-bit (or the 4-bit) multiplier and multiplicand respectively which are being multiplied to produce the final 4-bit(or the 8-bit) product vector.
The half of the LSB bits of multiplication output is taken as RHS product of the total multiplication.

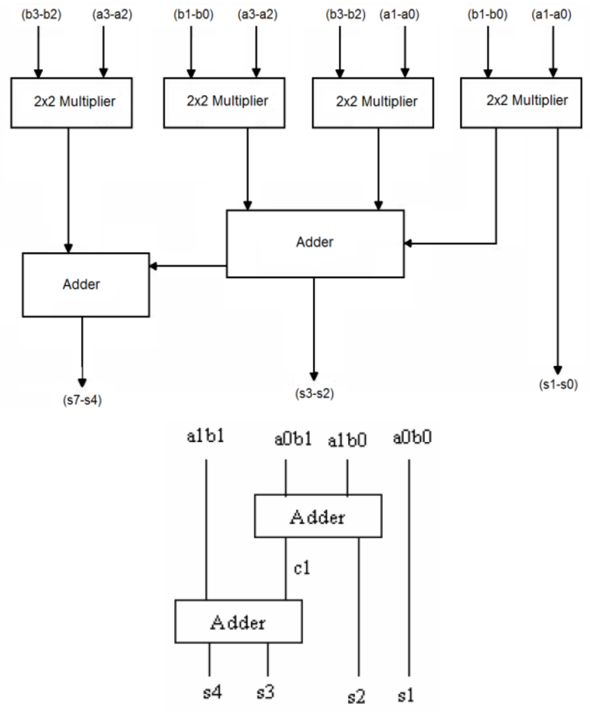
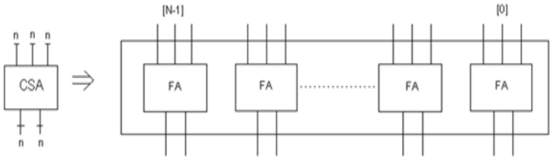


**Figure 4**

### 3.2.3 CSA:

The carry-save unit consists of *n* full adders, as shown in Figure5 each of which computes a single sum and carry bit based solely on the corresponding bits of the three input numbers.



n=4 in the evaluated multiplier of this paper

**Figure 5**

With three *n* - bit numbers $a_i$, $b_i$, and $c_i$ given to it, it produces a partial sum $ps_i$ and a shift-carry $sc_i$ according to the below equations:

$$ps_i = a_i \oplus b_i \oplus c_i$$

$$sc_i = (a_i \wedge b_i) \vee (a_i \wedge c_i) \vee (b_i \wedge c_i)$$

These $ps_i$ and $sc_i$ are then added using a conventional adder, to produce the sum of the three inputs.

The multiplier, multiplicand and the half of MSB bits are given as inputs to the CSA. The two outputs i.e., sum vector and carry vector obtained from the CSA adder are given to the inputs for the adder block. The output he we obtain is labeled the LHS of the required multiplication product.

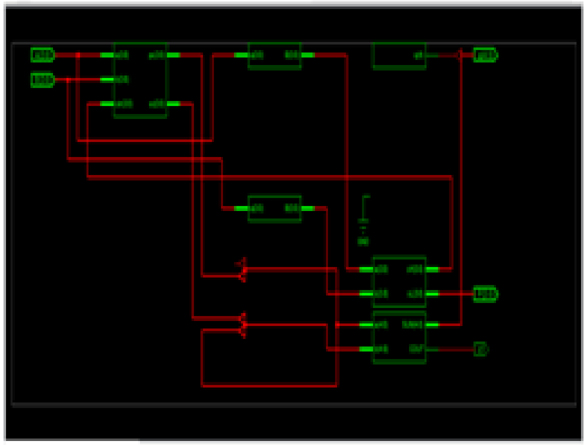
As we are using CSA, the delay will be reduced and the number of components gets reduced for addition mechanism. This is the main advantage of the multiplier based on Nikhilam over the multiplier based on conventional algorithms proposed in the aforementioned algorithms in 'I'.

## 4. Evaluation

In this section we evaluated the multiplier's performance employing Active HDL 7.2 and Xilinx 13.1i to code and synthesize respectively the module proposed.

### 4.1. Simulation Methodology

Xilinx 13.1i has been used to simulate the wave forms. The simulator carefully modeled the interconnections, the associated blocks and the propagation delays.



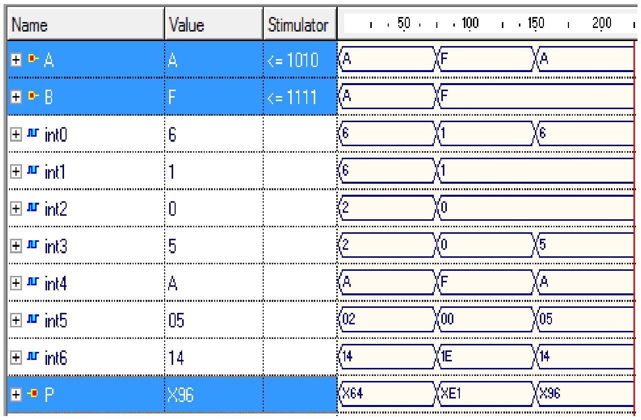
RTL Schematic

| | | |
|---|---|---|
| Number of Slices: | 27 out of 960 | 2% |
| Number of 4 input LUTs: | 47 out of 1920 | 2% |
| Number of IOs: | 17 | |
| Number of bonded IOBs: | 17 out of 83 | 20% |

Device utilization summary

### 4.2.Results:

In this section we show results for the Vedic multiplier based on Nikhilam Sutra and compare these with the conventional multiplier. Multiplier based on Nikhilam Algorithm utilizes smaller area and produces littler delay than the conventional multiplier. This reduction in the delay is attributed to the diminished consumption of area by the designed multiplier which possesses quite lesser number of internal blocks. Our results show that the Multiplier based on Nikhilam Algorithm is way more efficient than the conventional multiplier-be it in the area utilized or the delay associated.



| Name | Value | Stimulator | 50 · 100 · 150 · 200 |
|---|---|---|---|
| A | A | <= 1010 | A F A |
| B | F | <= 1111 | A F |
| int0 | 6 | | 6 1 6 |
| int1 | 1 | | 6 1 |
| int2 | 0 | | 2 0 |
| int3 | 5 | | 2 0 5 |
| int4 | A | | A F A |
| int5 | 05 | | 02 00 05 |
| int6 | 14 | | 14 1E 14 |
| P | X96 | | X64 XE1 X96 |

Simulation Results

## 5. Conclusion

The proposed Vedic multiplier architecture exhibits speed improvements. The 4x4 Vedic multiplier employing Nikhilam Sutra found to be better than 4x4 conventional multiplier 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.

## 6. References

i. *Charles. Roth Jr. "Digital Systems Design using VHDL," Thomson Brooks/Cole, 7th reprint, 2005.*

ii. *Jagadguru Swami Sri Bharati Krisna Tirthaji Maharaja, Vedic Mathematics: Sixteen Simple Mathematical Formulae from the Veda, Delhi (1965).*

iii. *H. Thapliyal and M. B. Shrinivas and H. Arbania, "Design and Analysis of a VLSI Based High Performance Low Power Parallel Square Architecture", Int. Conf. Algo. Math.Comp. Sc., Las Vegas, June 2005, pp. 72-76.*

iv. *P. D. Chidgup kar and M. T. Karad, "The Imp lementation of Vedic Algorithms in Digital Signal Processing", Global J. of Engg. Edu, vol.8, no.2, 2004.*

v. *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.*