# FPGA Implementation of Simple and High Speed Vedic Multiplier

Shilpi Thawait[1], Jagveer Verma[2]
[1](M.Tech Scholar, Department of Electronics & Telecommunication, Chouksey Engineering College / CSVTU Bhilai(C.G.), India)
[2](Assistant Professor, Department of Electronics & Telecommunication, Chouksey Engineering College / CSVTU Bhilai(C.G.), India)

***Abstract***

*Multiplier is the basic and the key element used in many Digital Signal Processing applications, arithmetic operations, in image processing applications such as FFT (fast fourier transform), convolution, correlation etc. This paper presents a simplified and efficient method of multiplication using vedic mathematics. Vedic mathematics is the name given to the ancient Indian system of mathematics and is based on mental calculations. Urdhva Triyakbhyam sutra of vedic multiplication is found to be the most efficient sutra of vedic multiplication among its all 16 sutras. The aim of this paper is to implement a simpler and high speed vedic multiplier by using Urdhva Triyakbhyam sutra efficiently.*

*Synthesis has been done using Xilinx ISE 9.2i simulator using VHDL language.*

**Keywords -** *CSA(Carry Save Adder), ISE (Integrated Software Environment), LUT (Look Up Table), UT(Urdhva-triyakbhyam), Vedic Multiplier, VHDL(Very High Speed Integrated Circuit Hardware Description Language), Xilinx.*

## I. INTRODUCTION

The multiplier is one of the most important block used in all the processors now a days. A variety of computer arithmetic techniques are used to realize a digital multiplier. Most of the techniques involve calculating the partial products and then summing them together. Most of the application requiring Digital Signal Processing requires multiplier. Hence speed , complexity and power consumption of the multiplier are of much important parameters to be considered. This paper presents a well organized and systematic design of multiplier with high speed and performance.

The architecture of the multiplier presented here is based on the Vertical and Crosswise algorithm that is the Urdhva Triyakbhyam sutra of multiplication of vedic mathematics.

## II. VEDIC FORMULAE

The word 'Vedic' has been derived from the word 'veda' which means the store-house of all knowledge. There are mainly 16 sutras in vedic mathematics dealing with various branches of mathematics like arithmetic, algebra, geometry etc.

These 16 sutras of vedic mathematics along with their brief descriptions 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 the 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.[1]

Study of these formulae is a field of diverse study. The proposed multiplier uses Urdhva-Triyakbhyam sutra only. Hence detail description of only Urdhva-Triyakbhyam sutra is given below in this paper and detail description of all the other formulae are beyond the scope of this paper.

### *A. Urdhva Tiryakbhyam (UT) sutra*

This sutra literally means vertically and crosswise. This sutra is one of the best known sutra of Vedic Sutras that provides an effective algorithm applicable to all multiplication cases. In addition, the sutra yields faster operations by generating partial product and sum in a single iteration step.

Two-digit multiplication based on this method is shown in Fig.2.1. To explain using an example, consider the multiplication of two digits, 23 and 52.

The first step is the multiplication (3x2), which becomes the LSB bit. Then, the addition of operation (2x2) and (5x3) is carried out. One digit answer is placed left to LSB and the carry is forwarded to the next stage. Next step is multiplication of 2x5 and adding the product with the previous carry. This procedure can be extended for 4x4 bit multiplication as shown in Fig.2.2 and can be further extended to perform NxN bit multiplication.[2]

In Fig.2.2, vertical and slanted lines represent the AND operation i.e. partial product generation between operands while horizontal lines represent summing of partial products.

Steps to be followed for the multiplication process for 4x4 multiplier are shown in Fig.2.2.

Considering ABC as multiplicand and DEF as the multiplier, the steps of multiplication are descriptive in the fig.2.2.[3]

### B. 2x2 Multiplier

Applying Urdhva-Triyakbhyam sutra let us take two inputs each of 2 bits say A1A0 and B1B0. Output can be of 4 bits. Result of multiplication is obtained after getting partial product and performing addition.

```
                    A1 A0

               x B1 B0

   ___________________________

               A1B0 A0B0

        A1B1 A0B1

   ___________________________

     CY    S2    S1    S0
```

In equation form, Urdhva-Triyakbhyam sutra can be given as below:

Consider two 4-bit binary numbers a3a2a1a0 and b3b2b1b0. The partial products (P7P6P5P4P3P2P1P0) generated are given by the following equations:

i. P0= a0b0
ii. P1= a0b1 + a1b0
iii. P2 = a0b2 + a1b1 + a2b0+ P1
iv. P3= a0b3 + a1b2 + a2b1 + a3b0+ P2
v. P4 = a1b3 + a2b2 + a3b1 + P3
vi. P5 = a1b2 + a2b1 + P4
vii. P6 = a3b3 + P5
viii. P7 = carry of P6 [4]

In the fig.2.3 S0 is the vertical product of bit A0 and B0. S1 is the addition of crosswise bit multiplication that means addition of A1xB0 and B1xA0. S2 is again the sum of vertical multiplication of bit A1 and B1 with the carry generated during S1 calculation, if any. CY is the carry generated during S2 calculation , if any.

As shown in Fig.2.3 half adders are used to add the partial products generated.

### C. 4x4 Multiplier, 8x8 Multiplier, 16x16 Multiplier

For getting high speed performance 4x4 multiplier can be designed using 2x2 and CSA (Carry Save Adder) as shown in fig.2.4.

### D. CSA(Carry Save Adder)

Carry Save Adders have been used in above architectures for adding the partial products generated by the multiplication of bits. CSA basically consists of n-full adders and a Ripple Carry Adder but here in proposed architectures it consists of full adders as well as half adders as per the need and a Ripple Carry Adder. Each full and half adder computes a single sum and carry bit based solely on the corresponding bits of the inputs which in number is 3 for full adders and is 2 for half adders. For given the three n-bit numbers a,b,c as input it produces a partial sum bit, PS and a shift carry bit, SC as :

$PS_i = a_i$ XOR $b_i$ XOR $c_i$

$SC_i =$ ($a_i$ AND $b_i$) OR ($b_i$ AND $c_i$) OR ($c_i$ AND $a_i$)

The actual sum can be computed by the following steps :

1. Shifting the carry sequence SC left by one bit position.
2. Appending a ‘0’ as the most significant bit of the partial sum sequence PS.
3. Using ripple carry adder add these two sequence together and produce the resulting n+1 bit value.[6]

This process can be continued indefinitely, adding an input for each stage of half and full adders without any intermediate carry propagation. CSA is being used here in proposed architecture because number of stages it uses for calculation of sum and hence propagation delay are fixed regardless of the number of bits. Hence it speeds up the carry propagation and provides faster operation and has paid a very important role in increasing the speed of operation of proposed multiplier.

## III. SIMULATION RESULT

The 16x16 multiplier has been designed in VHDL and its functionality is verified for various possible inputs by generating test bench. For fast operation adder should be fast so here concept of Carry Save Adder has been used. Simulation result of 8x8 multiplier is shown in fig.4.1.

The VHDL program of 8x8 multiplier has been used for designing 16x16 multiplier. Simulation result of 16x16 Vedic Multiplier is shown in fig.4.2.

## IV. SYNTHESIS RESULT

| Multiplier | No. of slices used | No. of 4 input LUTs used | No. of IOBs used |
|---|---|---|---|
| 8x8 | 112 | 196 | 32 |
| 16x16 | 456 | 799 | 64 |

Synthesis has been done using Xilinx ISE 9.2i software. Target device for the synthesis is c3s50-5-pq208.

## V. FIGURES

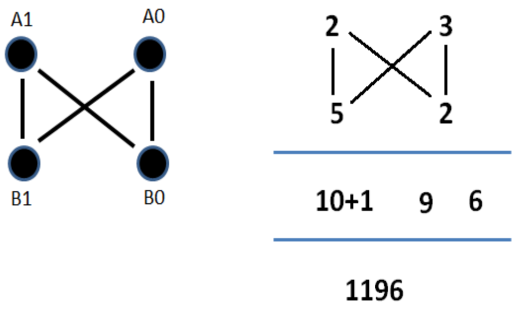

Figure2.1 Two digit multiplication using 'UT' sutra[2]

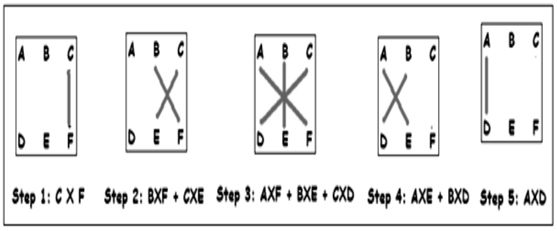

Figure2.2 Four bit multiplication using 'UT' sutra[3]

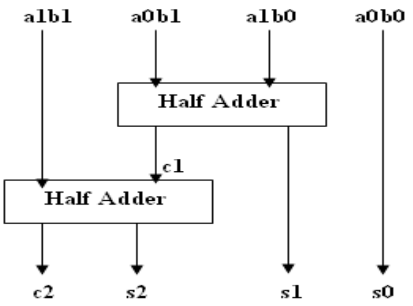

Figure2.3 Architecture of 2x2 vedic multiplier[5]

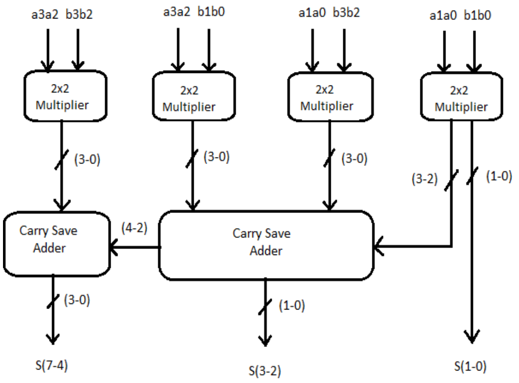

Figure2.4 Architecture of 4x4 vedic multiplier

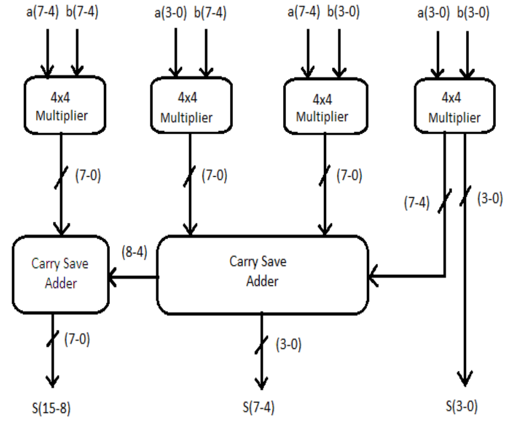

Figure2.5 Architecture of 8x8 vedic multiplier

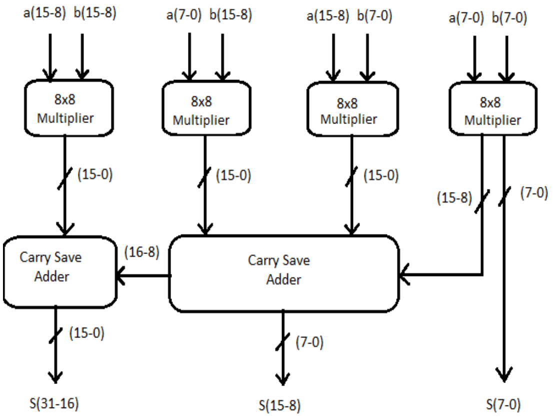

Figure2.6 Architecture of 16x16 vedic multiplier

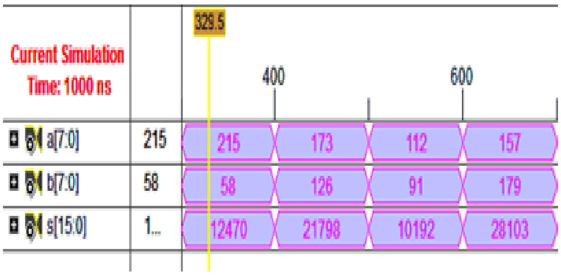

Figure4.1 Simulation Result of 8x8 Vedic Multiplier

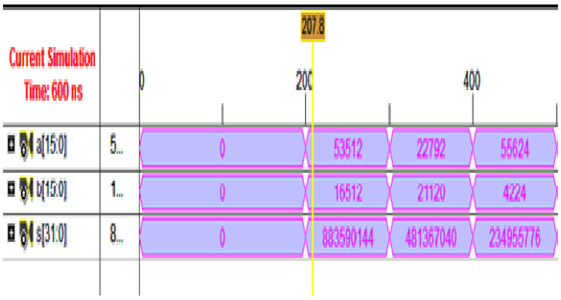

Figure4.2 Simulation Result of 16x16 Vedic Multiplier

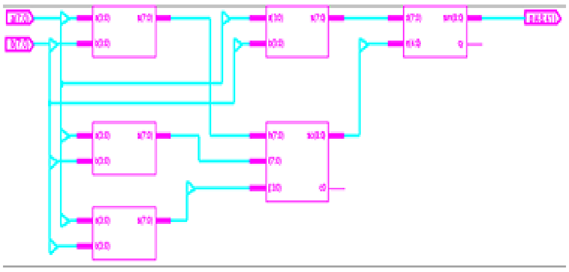

Figure5.1 RTL view of 8x8 Vedic Multiplier

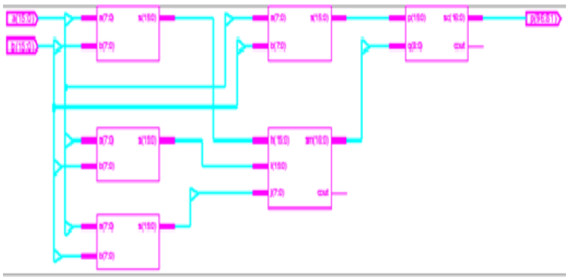

Figure5.2 RTL view of 16x16 Vedic Multiplier

## VI. CONCLUSION

In this paper a simpler concept with a simpler architecture of Vedic Multiplier has been presented. This provides hierarchical design for multiplier. So the modularity gets increased and the complexity gets reduced. The design is based on Vedic Mathematics and uses concept of CSA that means modified form of the same has been used here.

## 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, 2008.

[2] Hardik Sangani, Tanay M. Modi, V.S. Kanchana Bhaaskaran' "Low Power Vedic Multiplier Using Energy Recovery Logic" , International Conference on Advances in Computing,Communications and Informatics (ICACCI),2014.

[3] Sudeep.M.C, Sharath Bimba.M, Mahendra Vucha ,"Design and FPGA Implementation of High Speed Vedic Multiplier", International Journal of Computer Applications (0975 – 8887) , Volume 90 – No 16, March 2014.

[4] Premananda B.S., Samarth S. Pai, Shashank B., Shashank S. Bhat, "Design and Implementation of 8-Bit Vedic Multiplier", IJAREEIE, vol 2, issue 12, December 2013.

[5] Pushpalata Verma, K. K. Mehta, "Implementation of an Efficient 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.

[6] Wikipedia, http://en.wikipedia.org/wiki/Carry-save_adder, DOB- 29/04/2015.