International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 07 | July 2021 www.irjet.net p-ISSN: 2395-0072

"Design and Implementation of an Efficient Modified Vedic Multiplier Incorporating Fast Adder and Its Applications"

Chaya V[1], Sudha H[2]

[1]*PG student, Dept. of Electronics and Communication Engineering, Bangalore Institute of Technology, Karnataka, India*
[2]*Associate Professor, Dept. of Electronics and Communication Engineering, Bangalore Institute of Technology, Karnataka, India*

Abstract - *Multiplication is an important fundamental function in any arithmetic operations. A Multiplier is one of the essential hardware blocks in the processor design. Vedic mathematics is a system of mathematics, where the name is given to the ancient Indian mathematic technique which help to solve the typical mathematical operations in an easy and faster way. Vedic math based multiplier is a high speed multiplier and adder is one of the main component used in the multiplication technique. In this work, a modified Vedic multiplier with high speed that incorporates the fast parallel prefix adder is designed and implemented based on Urdhva Tiryagbhyam (UT) sutra. The performance metrics of proposed Vedic multiplier are achieved and compared with existing Vedic multiplier and conventional multiplier. This Vedic multiplier is designed in Verilog language and simulation is done in Xilinx ISE 14.7. Further, the designed multiplier is applied to image processing applications.*

Key Words: **Vedic multiplier, Urdhva Tiryagbhyam sutra (UT), Parallel prefix adder (PPA)**

1. INTRODUCTION

While using any of the electronic devices, we may face a situation where the device hangs up or stops working for a moment. The main reason behind it is processor speed which is the primary feature of an arithmetic logic unit which has an impact on the power of a processor. It is well-known that the multiplier unit forms an integral part of processor design. In many of the mathematical operations, the execution time and speed are the major factors. So, due to this the fast multiplication process becomes an essential factor in any of the arithmetic blocks. Thus, the requirement to design the fast multiplier has been increased for a long time.

In this work, Vedic multiplier is designed using fast parallel prefix adder. By using parallel prefix adder, the summation of partial product in the multiplier can be increased and due to this the overall performance of the multiplier can be improved. Thus, the main reason of this work is to enhance the speed of the multiplication operation and to reduce the area by using the proposed Vedic Multiplier and further implement it in any of image processing applications.

In recent times, Vedic Mathematics is a one of the system of ancient mathematics which is becoming more popular and a speedier method to compute and analyze any complex or simpler arithmetic calculations. It was discovered by Indian mathematician Shri Bharathi Krishna Tirthaji and first published in 1965. The origin of Vedic mathematics were derived from the Sanskrit word 'Veda' which means 'Knowledge'. Vedic Mathematics contains list of 16 sutras to solve mathematical calculations in easy and faster way which is applicable to all branches of mathematics. The mathematical operations using this Vedic Method require less hardware, and this can also increase the computational speed of processor.

Among all the 16 sutras Urdhva Tiryagbhyam is the most popular sutra used for the design of Vedic multipliers. Urdhva Tiryagbyham algorithm follows the "vertical and crosswise" multiplication which includes the parallel multiplication process and involves in the fast multiplication operation by greatly reducing the number of partial products.

They have found enormous use in many applications like image processing, DSP, MAC, ASIC design etc. Hence, the designed Vedic multiplier is necessary in the fields of medical image processing, x-ray imaging, signal processing, robotics etc. to save time and area.

2. METHODOLOGY

In a simple way multiplication, the two integers are considered namely multiplicand and multiplier which involves the process of adding a multiplicand integer to itself a number of times specified by another multiplier integer. Firstly, a simple Vedic multiplier is designed using the Urdhva Tiryagbhyam Sutra which has the following three concepts namely,

1. By following the vertical and diagonal (crosswise) multiplication, the partial products are generated.
2. By using parallel prefix adder, the reduction of partial products takes place and the obtained carry will be given to next adder.
3. Final addition.

The above 3 steps has to be repeated continuously to all bits to obtain final product. Commonly, this technique is used to perform multiplication process of any given decimal

numbers by using vertical and crosswise method and this procedure is used for 2*2 and 3*3 digits multiplication which is seen in the below Fig- 1 and Fig-2.

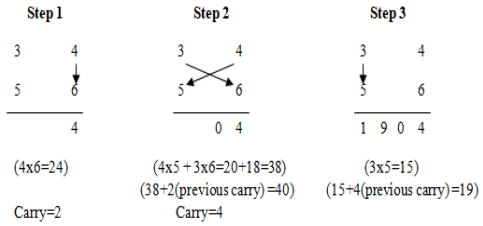

Fig -1: Example of 2x2 bits multiplication using UT Sutra

In the above Fig -1, the multiplication of 2 numbers, 34 and 56 is shown by following the method of Urdhva Tiryagbhyam Sutra.

Step 1: Initially 4 & 6 are multiplied and in the obtained value (24), the least significant digit is written as the least significant digit of final product (4) and the other number (2) from the obtained value is given as a carry for the next step.

Step 2: Then cross multiplication of 3 & 6 and 4 & 5 is performed and the product terms are added and also previous carry(2) from the previous stage is added to this product terms and from this obtained value (40), the next bit of final product is generated. The least significant digit (0) of the obtained value is written in the final product and the most significant digit (4) is considered as the carry for the next stage.

Step 3: At final stage, 3 & 5 are multiplied vertically and added with previous carry (4) from the previous stage and the obtained value (19) is taken as most significant digit of the final result.

Similarly by following the above procedure for the multiplication process of 3* 3 bit is performed and the example of 3*3 bit is shown in the below Fig -2.

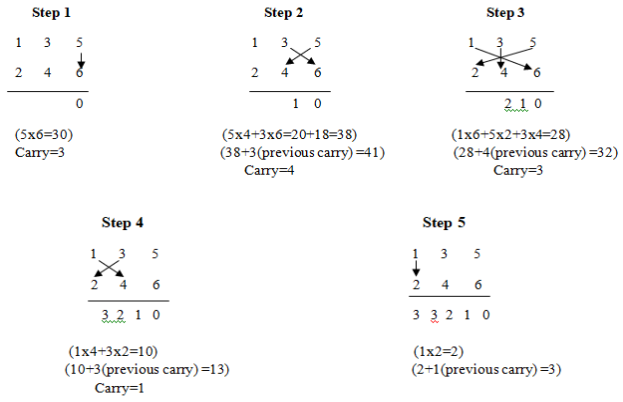

Fig -2: 3X3 multiplication using UT Sutra

The above Fig -2 describes the steps of multiplying the 3x3 bits, 135 with 246 using Urdhva Tiryagbhyam Sutra.

In this multiplication process, the digits which are vertically lined up are multiplied and added with each other and the carry obtained from the previous stages is added to the obtained value. From this obtained value, each digit of the final product is generated. The sum digit and a carry digit which is added to the next stage are generated and it is not propagated further at final step. In the complete multiplication process only least significant digit of the obtained value is considered as result bit and the remaining bit is taken as carry for next stage. At last step, the obtained value is considered as the most significant digit of the final product.

Similarly, the higher order multiplication operation for any number of bits such as 4x4, 8×8, 16×16, and 32x32 can be done using the same procedure of Urdhva Tiryagbhyam Sutra.

2.1 Parallel Prefix Adder (PPA):

Ling introduced the modified parallel prefix adder for prefix addition which is faster than that of the conventional adder. This modified parallel prefix adder which was proposed is used to calculate the prefixes formed and can be able to save the significant hardware consumption.

Parallel Prefix adder performs the parallel addition which involves the 3 step process to compute the output namely,

1. Pre- processing step
2. Carry generation step
3. Post- processing step

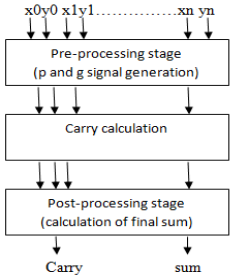

Fig 3- Mechanism of Parallel Prefix adder

1. Pre-processing step

In this step, the Generate (gi) signal and Propagate (pi) signal is computed for all the bits of the adder by using the below equations (1) and (2).

$$pi = ai \wedge bi \quad \ldots\ldots (1)$$
$$gi = ai \,\&\, bi \quad \ldots\ldots (2)$$

2. Carry generation step

Here, rather than the normal carry (ci), the pseudo carry (hi) is calculated initially, which leads for the saving of a logic element. By following the below equation (3), the pseudo carry can be computed.

$$hi = gi + gi{-}1 + pi{-}1.gi{-}2 + pi{-}1.pi{-}2.gi{-}3 + pi{-}1.pi{-}2.pi{-}3.gi{-}4 \ldots\ldots + pi{-}1.pi{-}2 \ldots\ldots p1.g0 \ldots\ldots (3)$$

Next, the real carry (Ci) is calculated by using the below equation (4),

Ci =hi.pi …… (4)

However, the carry calculation by following the above method leads to the greater area consumption and the power leakage of the adder. Due to this, the ling has modified the equations to achieve the better performance of the adder. Here, the propagate and generate signals acts as the intermediate signals and those are given by the below (5) and (6) equations, where i > k > j

P(i:k) = P(i:j) . P(j−1:k)…… (5)
G(i:k) = G(i:j)+ G(j-1:k) . P(i:k)…… (6)

Next, by considering the pseudo carry in terms of intermediate signals (G(i:k), P(i:k)), the real carry (Ci) is calculated by following the below equation (7),

Ci = (Gi:k + Pi−1:k−1.Gk−1:j+1).pi…… (7)

By following this modified equation in every single sum bit calculation saves one logic gate which in turn reduces the area consumption and makes the calculation of the sum easier.

3. Post- processing step

In this final step, the final addition is computed by using the below equation (8),

Si = pi ^ Ci …… (8)
Ci+1 = (pi.C0) + gi

By using this modified PPA, the rapidity of carry propagation process increases which results in the faster summation process of the partial products. This leads to the better performance of a designed Vedic multiplier.

3. PROPOSED VEDIC MULTIPLIER

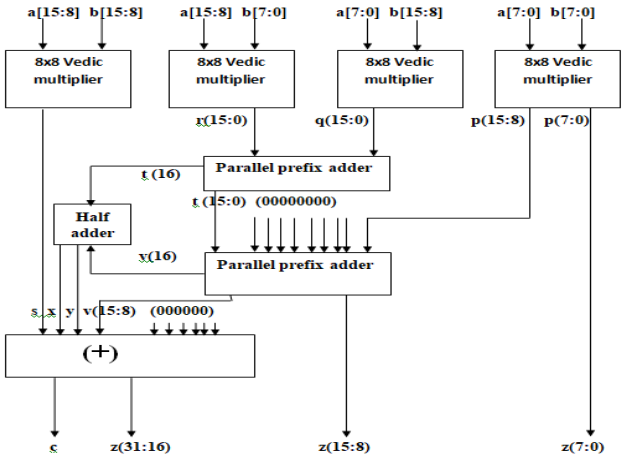

Fig -4: 16x16 bit Vedic multiplier architecture using PPA

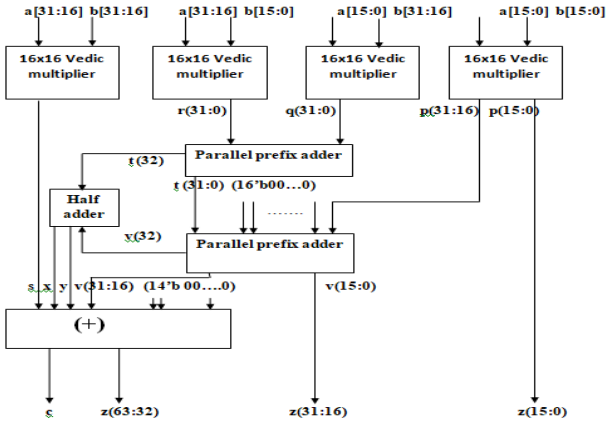

Fig -5: 32x32 bit Vedic multiplier architecture using PPA

The proposed block diagrams represents the design of the 16x16 bit and 32x32 bit Vedic multiplier(VM) consisting of parallel prefix adder by using the Urdhva Tiryagbhyam(UT) Sutra and these are seen in the above Fig -4 and 5 respectively. The design initially starts with the implementation of 2 x 2 bit Vedic multiplier using the above mentioned method which is shown in Fig -1. Vedic Multiplier block is then instantiated for further bits like 4 x 4 bit, 8 x 8 bit and so on. By using the designed 8 bit Vedic multiplier, the 16 bit Vedic multiplier is designed. And again by using this designed 16 bit Vedic multiplier, the design of 32 bit Vedic multiplier is done. Parallel Prefix Adder (PPA) is used in the design for the addition purpose which leads to the reduction of partial products. In the overall design of 32x32 bit Vedic multiplier, the two-16 bit PPAs and an half adders were used in the design for the overall multiplication process.

Initially, the four 8x8 bit VM are used to design 16x16 bit VM. The lower 8 bits [p(7:0)] of right most 8 bit VM output are taken directly as the output of the final product of the 16x16 bit VM[z(7:0)]. The next two Vedic multiplier's output [q(15:0) & r(15:0)] is given to the parallel prefix adder for the summation of the obtained partial product arrays. And the output from this PPA [t(15:0)] is then given as an input to the other PPA. And the other input for this PPA is obtained from the remaining upper 8 bits [p(15:8)] from the right most VM. These upper 8 bits are being padded with 0's to make it as 16 bit. The carry generated from both the PPA's [t (16) & v (16)] are given to half adder. The 8 LSB bits [v (7:0)] of 2nd PPA's output are considered as the output of the final product of the 16 x 16 bit VM [z (15:8)]. Further, the output[s (15:0)] of the left most VM, the remaining upper 8 bits [v(15:8)] of the 2nd PPA and the sum(x) and carry(y) bit of the half adder and the 6 bits of zero are being padded with these to make it 16 bits, and overall bits are given to an adder as input for the final addition purpose. The 16 bit output from this adder are taken directly as the output of the final product of the 16 x 16 bit VM[z(31:16)] and the MSB is considered as the carry bit (c).

Similarly, the 32x32 bit Vedic multiplier is designed by following the same steps used to design the 16x16 bit Vedic multiplier.

4. SIMULATION RESULTS OF DESIGNED VEDIC MULTIPLIER

The simulation results of 32 x 32, 16 x 16 and 8 x 8 bits are shown in the below figure 6, 7 and 8 respectively. The simulation is performed in Xilinx ISE 14.7 software tool. In these waveforms the multiplier, multiplicand (inputs), partial products and final product (output) are shown.

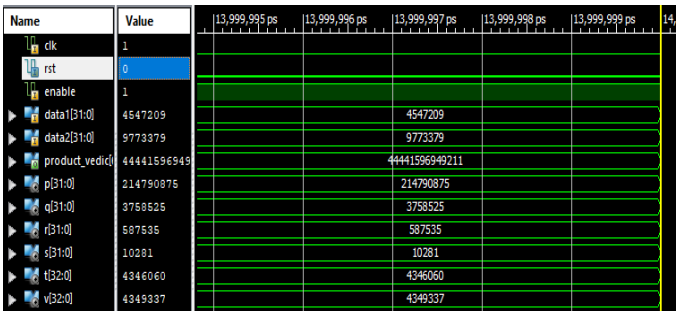

Fig -6: Output of 32 x 32 bit designed Vedic multiplier

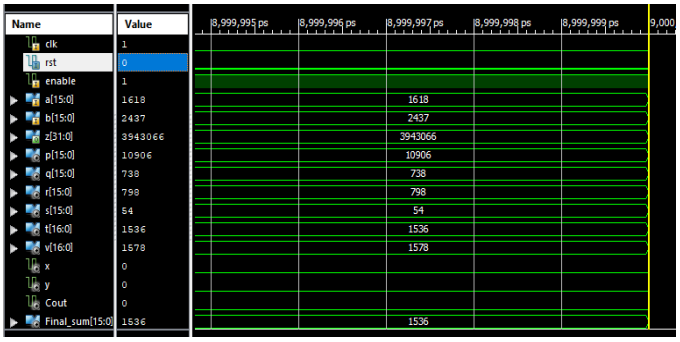

Fig -7: Output of 16 x 16 bit designed Vedic multiplier

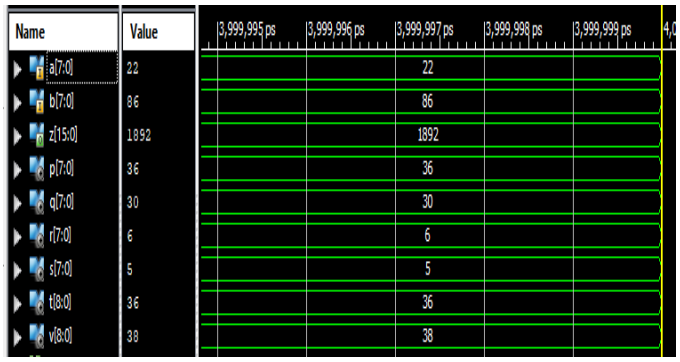

Fig -8: Output of designed 8 x 8 bit Vedic multiplier

The performance parameters achieved by the Proposed Vedic multipliers are given in the below Table -1.

Table -1: Performance parameters

Proposed Vedic multiplier	Delay (ns)	Frequency (MHz)	4 input LUTs/Slices	FPGA Technology
8 x 8 VM	3.232	309.406	187/102	Spartan 3E
16 x 16 VM	3.644	274.424	766/418	Spartan 3E
32 x 32 VM	8.266	120.970	3182/1740	Spartan 3E

Comparison of performance parameters like delay and LUTs of Proposed Vedic multiplier with existing Vedic multiplier and conventional booth multiplier are shown in the below Table- 2.

Table –2: Comparison of performance metrics of different multipliers

Multiplier type	Delay (ns)			LUTS		
	8x8	16x16	32x32	8x8	16x16	32x32
Proposed Vedic multiplier	3.232	3.644	8.266	187	766	3182
Existing Vedic multiplier	24.186	43.714	68.859	201	903	3802
Booth multiplier	36.640	61.251	17.473	274	473	4002

From the above performance and comparison table, the proposed Vedic multiplier is best suitable multiplier to use in any of the applications like image processing, digital signal processing, ASIC Design, MAC etc.

5. APPLICATIONS

The designed Vedic multiplier is used in image processing applications like RGB to gray scale conversion, image segmentation, image inversion and image watermarking. The above designed Vedic algorithm is implemented in these above mentioned applications to enhance the quality of the image and to obtain the results of SNR and PSNR. The main idea behind using this designed Vedic multiplier is to make the operations on images in a faster way. The input image given is shown in Fig -9.

Fig -9: Input image

A. RGB to gray scale conversion

A grayscale image is a type of image which consists of the colors in the shades of black and white. This image is usually 8-bit image having 256 combinations of gray shades with each pixel.

Initially in the given image, the color pixel is described by the intensities of red, green, and blue. One of the methods used here to convert the RGB to gray is luminosity method in which it averages the values of RGB, but it gives a weighted average to green color, since it is more sensitive to human eye than other colors. And, the formula used in this method to obtain the output is 0.21 R + 0.72 G + 0.07 B.

Fig -10: Output image of Grayscale

B. Image color segmentation

The collection of various pixels is present in any of the given input image based on the background or foreground images. The set of pixels that have similar feature are grouped together by using image segmentation. One of the methods of image segmentation is Thresholding.

In this thresholding method, the image pixels get partitioned based on the given intensity level and divides the image into two parts namely foreground and background. Depending upon the value of pixels with reference to the threshold value, the foreground images and background images can be differentiated. Here, by using this method, the foreground images are retained over to that of background images.

Fig -11: Output of Image color segmentation

C. Color Image inversion

Color image inversion is the simplest and easiest technique in image processing. In this technique, black and white image inversion takes place where the white (light) areas are inverted to black (dark), and black regions are inverted to white regions. And more distinct shades are inverted by the complementary colors.

Fig -12: Output image of Color inversion

D. Image watermarking

Image watermarking is the technique of embedding one image into the other source image by the user based on image feature detection. To obtain the image watermark, one image needs to be embedded with the other image by using an algorithm.

Fig -13(a): Source image **Fig. 13(b):** Image to be hidden

Fig -13(c): Output of image water marking

The Signal to Noise Ratio (SNR) and Peak Signal to Noise Ratio (PSNR) of the output images are given in the below Table -3. Based on peak PSNR values, the image quality can be measured.

Table -3

Application	SNR value	PSNR value
RGB to grayscale conversion	22.0146	22.7033
Image segmentation	16.8054	19.6736
Image inversion	13.9450	17.8480
Image watermarking	15.7624	18.7994

6. CONCLUSION

An efficient Vedic multiplier incorporating parallel prefix adder is designed and is implemented using "Urdhva Tiryagbhyam" sutra. This proposed multiplier provides better solution which further increases the performance of the system in terms of speed and reduces the delay, area utilization and complexity. The delay and area of 8x8, 16x16, 32x32 bits designed Vedic multiplier is compared with existing Vedic and conventional booth multipliers. Further the proposed Vedic multiplier is applied to the image processing applications.

REFERENCES

[1] Preyesh Dalmia, Vikas, Abhinav Parashar, Akshi Tomar and Dr. Neeta Pandey, "Novel high speed Vedic Multiplier proposal including Adder based on Quaternary Signed Digit number system" 2380-6923/17 © 2018 IEEE DOI 10.1109/VLSID.2018.78.

[2] Ram, G. Challa, D. Sudha Rani, R. Balasaikesava, and K. Bala Sindhuri, "VLSI architecture for delay efficient 32-bit multiplier using vedic mathematic sutras," 2016, IEEE.

[3] Shamim Akhter, Vikas Saini and Jasmine Saini, "Analysis of Vedic Multiplier using Various Adder Topologies", 2017, IEEE

[4] Dr. G.S. Sunitha and Rakesh H, "Performance Comparison of conventional multiplier with vedic multiplier using ISE simulator," 2018, IEEE.

[5] Josmin Thomas, R.Pushpangadan, Jineshs, "Comparative Study of Performance Vedic Multiplier on The Basis of Adders Used," 2015, IEEE.

[6] K. Deergha Rao, PP.V. Muralikrishna, and Ch. Gangadhar, "FPGA Implementation of 32 Bit Complex Floating Point Multiplier Using Vedic Real Multipliers with Minimum Path Delay," 978-1-5386-5002-8/18/$31.00 ©2018 IEEE,

[7] Rohan Pinto and Kumara Shama, "Low Power Modified Shift-Add Multiplier Design using Parallel Prefix Adder," IEEE, 2018

[8] Gadakh, S. N and A. S. Khade, "FPGA implementation of high speed vedic multiplier," International Conference and Workshop on Electronics and Telecommunication Engineering, 2016.

[9] M. Akila, C. Gowribala and S. Maflin Shaby, "Implementation of High Speed Vedic Multiplier using Modified Adder," 2017.

[10] Shobha Sharma, Amita Dev and Akanksha Kant, "An Efficient Implementation of High Speed Vedic Multiplier Using Compressors for Image Processing Applications," International Journal of Computer and Information Engineering, 2016.

[11] Akanksha Kant and Shobha Sharma, "Applications of Vedic multiplier designs –A review," 978-1-4673-7231-2/15/$31.00 ©2015, IEEE.

[12] Raj Kumari and Rajesh Mehra, "Power and Delay Analysis of CMOS Multipliers using Vedic Algorithm," IEEE, ICPEICES-2016.