Performance Comparison of Vedic Multiplier and Booth Multiplier

Anju

Abstract –The performance of the any processor will depend upon its power and delay. The power and delay should be less in order to get a effective processor. In processors the most commonly used architecture is multiplier. If the power and delay of the multiplier is reduced then the effective processor can be generated. In this paper Vedic Multiplier and Booth Multiplier are implemented on FPGA and comparative analysis is done. The Comparison of these Architectures are carried out to know the best architecture for multiplication w. r. t. power and delay characteristics. The designs are implemented using VHDL in Modelsim 10.1 b and synthesis is done in Xilinx 8.2i ISE.

Index Terms - Urdhva Tiryagbhyam, Vedic multiplier, Booth multiplier, Xilinx.

I. INTRODUCTION

Multiplication is an important fundamental function in an ALU. Since, multiplication dominates the execution time of most DSP algorithms; therefore high speed multiplier is much desired [1]-[4]. Currently, multiplication time is still the dominant factor in determining the instruction cycle time of a DSP chip. With an ever-increasing quest for greater computing power on battery operated mobile devices, design emphasis has shifted from optimizing conventional delay, time, area and size to minimizing power dissipation while still maintaining the high performance [5]-[7]. The low power and high speed VLSI can be implemented with different logic styles. The three important considerations for VLSI design are power, area and delay. There are many proposed logics for, low power dissipation and high speed and each logic style has its own advantages in terms of speed and power. The objective of good multiplier is to provide a physically compact high speed and low power consumption unit. We applied Urdhva Tiryabhyam sutra to binary multiplier using carry save adders.

Thus Vedic multiplier using Urdhava Tiryagbhyam sutra has less delay and thus they are treated as high speed multipliers as compared to Booth's algorithm using recorded multipliers.

II. BOOTH MULTIPLIER

The Booth recording multiplier is a multiplier which scans three bits at a time to reduce the number of partial products [8]. These three bits are the two bit from the present pair and a third bit from the high order bit of an adjacent lower order pair. After examining each triplet of bits, the triplets are converted by Booth logic into a set of five control signals used by the adder cells in the array to control the operations performed by the adder cells. Booth algorithm gives a procedure for multiplying binary integers in signed 2's complement representation on. It operates on the fact that strings of 0's in the multiplier require no addition but just shifting and a string of 1's in the multiplier from bit weight 2^k to weight 2^m can be treated as 2^{K+1}-2^m.

Manuscript received on June, 2013.

Ms. Anju, M. Tech Scholar, Electronics & Communication Engg., IMS Engineering College, NH – 24, Adhyatmik Nagar, Ghaziabad, U.P., India.

Fig 1 shows the hardware implementation for Booth's algorithm. It consists of an n-bit adder, control logic and four registers A, B, Q and Q_{-1}. Multiplier and Multiplicand are loaded into register Q and register B respectively Register A and Q_{-1} are initially set to 0.

The n-bit adder performs addition, inputs of adders comes from multiplicand and content of register A. In case of addition, Add/Sub line is 0, therefore, C_{in} = 0 and multiplicand is directly applied as a second input to the n-bit adder. In case of subtraction, Add/sub line is 1, therefore C_{in} = 1 and multiplicand is complemented and then applied to the n-bit adder. As a result, the 2's complement of the multiplicand is added to the contents of register A.

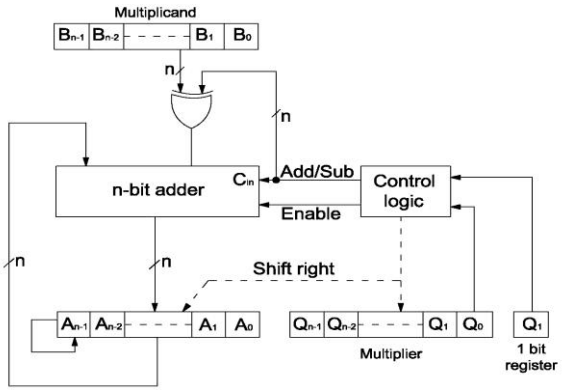


Fig. 1 Hardware implementation of signed binary multiplication for booth's algorithm

The control logic Scans bits Q_0 and Q_{-1} one at a time and generates the control signals to perform the corresponding function. If the two bits are same (1 1 or 0 0), then all the bits of A, Q and Q_{-1} registers are shifted to right I-bit without addition or subtraction (Add/Subtract Enable = 0). If the two bits differ, then the multiplicand is added to or subtracted from the A register, depending on the status of bits. After addition or subtraction right shift occurs such that the left most bit of A (A_{n-1}) is not only shifted into A_{n-2}, but also remains in A_{n-1}. This is required to preserve the sign of the number in A and Q.

The flow chart of Booth's algorithm for signed multiplication is shown in Fig 2.

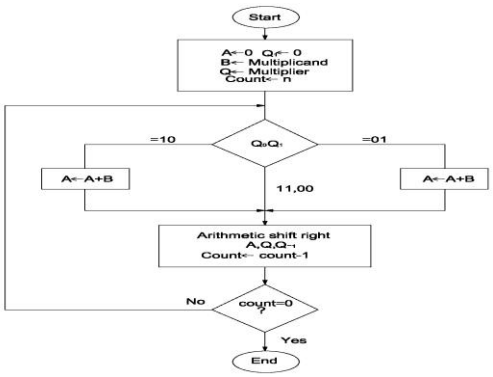
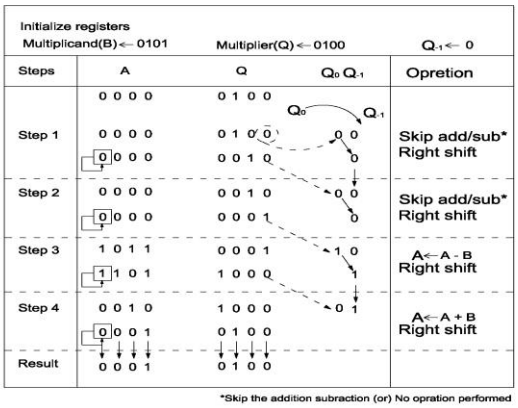


Fig. 2 Flow chart of Booth's algorithm for signed multiplication

Example-1 0101(5) x 0100(4)



Final Product 0101 x 0100 = 00010100

Here form the above example it can be proved that the addition/subtraction operation can be skipped if the successive bits in the multiplicand are same.

III. URDHAVA MULTIPLIER

The use of Vedic mathematics lies in the fact that it reduces the typical calculations in conventional mathematics to very simple ones. This is so because the Vedic formulae are claimed to be based on the natural principles on which the human mind works [9]-[10]. Vedic Mathematics is a methodology of arithmetic rules that allow more efficient speed implementation [11]-[12].

UrdhvaTiryakbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It means "Vertically and Crosswise" [13]-[16]. The digits on the two ends of the line are multiplied and the result is added with the previous carry. When there are more lines in one step, all the results are added to the previous carry. The least significant digit of the number thus obtained acts as one of the result digits and the rest act as the carry for the next step. Initially the carry is taken to be as zero. The line diagram for multiplication of two 4-bit numbers is as shown in Fig 3.

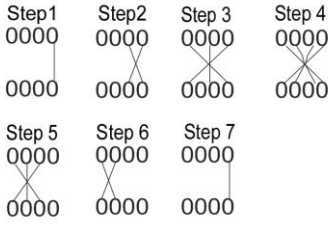


Fig. 3 Line diagram for multiplication of two 4-Bit Number

Now we will extend this Sutra to binary number system. For the multiplication algorithm, let us consider the multiplication of two 8 bit binary numbers $A_7A_6A_5A_4A_3A_2A_1A_0$ and $B_7B_6B_5B_4B_3B_2B_1B_0$. As the result of this multiplication would be more than 8 bits, we express it as $...R_7R_6R_5R_4R_3R_2R_1R_0$. As in the last case, the digits on the both sides of the line are multiplied and added with the carry from the previous step. This generates one of the bits of the result and a carry. This carry is added in the next step and hence the process goes on. If more than one lines are there in one step, all the results are added to the previous carry. In each step, least significant bit acts as the result bit and all the other bits act as carry. For example, if in some intermediate step, we will get 011, then I will act as result bit and 01 as the carry. Thus we will get the following expressions [12]-[13].

$R_0=A_0B_0$
$C_1R_1=A_0B_1+A_1B_0$
$C_2R_2=C_1+A_0B_2+A_2B_0+A_1B_1$
$C_3R_3=C_2+A_3B_0+A_0B_3+A_1B_2+A_2B_1$
$C_4R_4=C_3+A_4B_0+A_0B_4+A_3B_1+A_1B_3+A_2B_2$
$C_5R_5=C_4+A_5B_0+A_0B_5+A_4B_1+A_1B_4+A_3B_2+A_2B_3$
$C_6R_6=C_5+A_6B_0+A_0B_6+A_5B_1+A_1B_5+A_4B_2+A_2B_4+A_3B_3$
$C_7R_7=C_6+A_7B_0+A_0B_7+A_6B_1+A_1B_6+A_5B_2+A_2B_5+A_4B_3+A_3B_4$
$C_8R_8=C_7+A_7B_1+A_1B_7+A_6B_2+A_2B_6+A_5B_3+A_3B_5+A_4B_4$
$C_9R_9=C_8+A_7B_2+A_2B_7+A_6B_3+A_3B_6+A_5B_4+A_4B_5$
$C_{10}R_{10}=C_9+A_7B_3+A_3B_7+A_6B_4+A_4B_6+A_5B_5$
$C_{11}R_{11}=C_{10}+A_7B_4+A_4B_7+A_6B_5+A_5B_6$
$C_{12}R_{12}=C_{11}+A_7B_5+A_5B_7+A_6B_6$
$C_{13}R_{13}=C_{12}+A_7B_6+A_6B_7$
$C_{14}R_{14}=C_{13}+A_7B_7$

$C_{14}R_{14}R_{13}R_{12}R_{11}R_{10}R_9R_8R_7R_6R_5R_4R_3R_2R_1R_0$ being the final product [14]. Hence this is the general mathematical formula applicable to all cases of multiplication.

IV. FPGA IMPLEMENTATION AND EXPERIMENTAL RESULTS

In this work, 8x8 bit Vedic multiplier and 8x8 bit Booth multiplier are designed in VHDL (very High speed Integrated Circuits Hardware Description Language).

Logic simulation was done in Modelsim 10.1 b. Table 1 & 2 and Figs 4(a) & 5(b), show the synthesis and simulation results of 8x8 bit Booth multiplier and 8x8 bit Vedic multiplier respectively.

When multiplicand, booth multiplier_8 bit/mplicand_in = 11111111(225decimal) and multiplier, booth_multiplier_8 bit/mplier_in=11111111 (225 decimal), we get the result 16-bit output booth multiplier_8 bit/product = 111111100 000001 (65025.)

For 8x8 bit Vedic multiplier when input multiplicand = 10111111(191 decimal) and multiplier = 11111111 (225 decimal). Product of 1011111 (191) x 11111111 (255) =1011111001000001 (48705)

TABLE 1: Synthesis Result of 8x8 bit Vedic multiplier

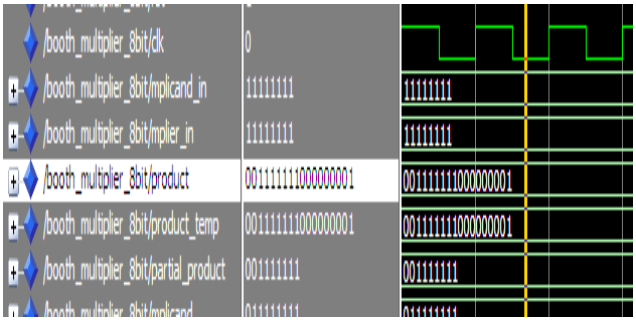
Device utilization Summary (estimated values)			
Logic utilization	**used**	**Available**	**Utilization**
Number of Slices	174	192	90%
Number of slice Flip Flop	133	384	34%
Number of 4 input LUTs ;	326	384	84%
Number of Bounded 10 Bs;	67	90	74%
Number of GCLKs	1	4	25%

(a)

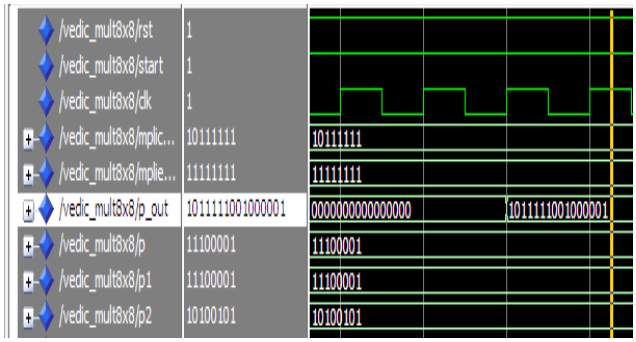
TABLE 2: Synthesis Result of 8x8 bit Booth multiplier

Device utilization Summary (estimated values)			
Logic utilization	**used**	**Available**	**Utilization**
Number of Slices	49	192	25%
Number of slice Flip Flop	48	384	12%
Number of 4 input LUTs ;	84	384	21%
Number of Bounded 10 Bs;	34	90	37%
Number of GCLKs	1	4	25%

(b)



(a)



(b)

Fig. 4 Simulation results of 8x8 bit Vedic and Booth Multiplier

V. EXPERIMENTAL RESULT

Table 3 compares the result of Area, Delay speed and power of Booth and Urdhava Tiryakbhyam multiplier. The result shows that Urdhava Tiryakbhyam multiplier is the best multiplier compare to Booth multiplier in terms of Area, delay and power consumptions (Fig. 5)

TABLE 3: Result for 8x8 Multiplier

FPGA Device Package XC2550 TQ144	**Area in LUT's**	**Delay (ns)**	**Speed (MH_z)**	**Memory (Kb)**	**Power (mW)**
Booth Multiplier	9	11.176	89.477	156440	11.30
Urdhava Multiplier	86	8.460	118.203	166744	9.29

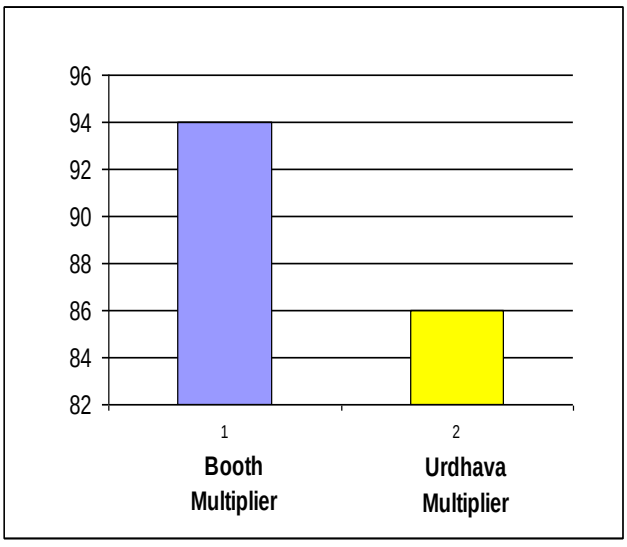


Fig. 5a : Area Comparison for (8x8)

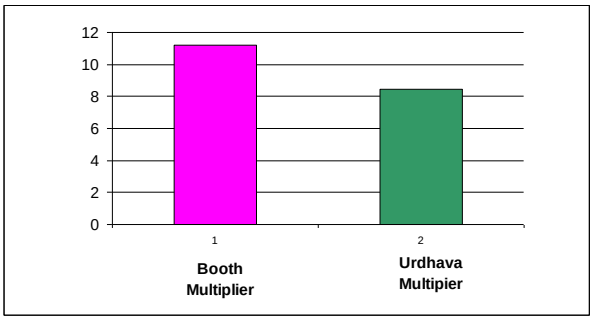


Fig. 5b Delay Comparison for (8x8)

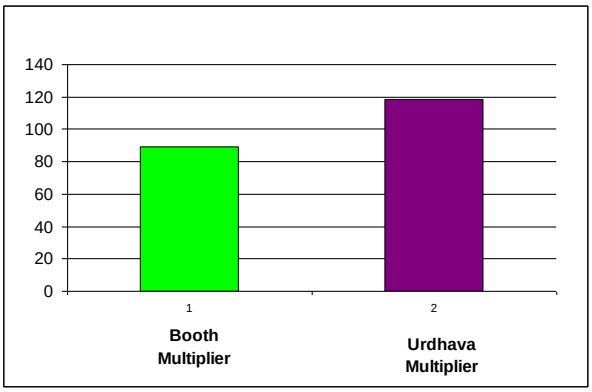


Fig. 5c Speed Comparison for (8x8)

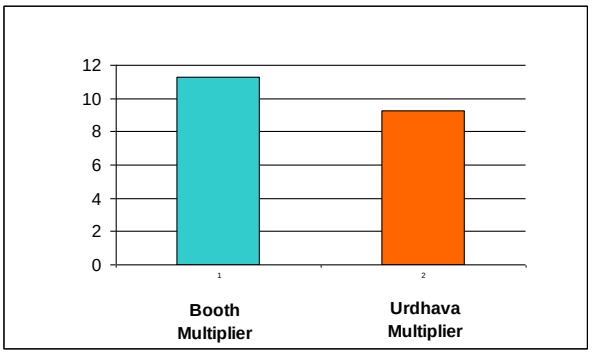


Fig. 5d Power Comparison for (8x8)

VI. CONCLUSION

It can be concluded that Urdhava Multiplier is superior in all respect like speed, delay, area, complexity and power consumption. Hence Urdhava Tiryakbham is a better multiplier compared to Booth Multiplier. The power of Vedic Mathematics can be explored to implement high performance multiplier in different VLSI application.

REFERENCES

1. Ravindra P Rajput, M. N Shanmukha Swamy, *"High Seed Modified Booth Encoder multiplier for signed and unsigned numbers"*, 14th Internaitonal Conference on Modeling and simulation, 2012 IEEE, pp. 649-654.
2. Rashmi K. Lamte and Prof. Bhasker, *"Speedy Convolution using Vedic Mathematics"*, International Journal of Recent Trends in Engineering and Technology, Vol-05, No-01, March 2011.
3. Jagadeshwar Rao M, Sanjay Dubey, *"A high speed and Area Efficient Booth Recoded Wallace tree Multiplier for fast Arithmetic Circuits,"* Asia Pacific Conference on postgraduate Research in Microelectronics & Electronics (PPIM EASIA) 2012.
4. Ch. Harish Kumar *"Implementation and Analysis of Power, Area and Delay of Array Urdhava, Nikhilam Vedic Multipliers, "* Internatinal Journal of Scientific and Research Publications, Volume 3, Issue 1 January 2013, ISSN 2250 – 3153.

5. Nidhi Mittal, Abhijeet Kumar, "*Hardware Implementation of FFT using vertically and crosswise Algorithm"*, International Journal of Computer Application (0975-8887), Volume-35- No-1, December 2011.
6. L. Sriraman, T.N. Prabakar." *Design and Implementation of two variable Multiplier using KCM and Vedic Mathematics*, 1st International Conference on Recent Advances in Information Technology 2012 IEEE.
7. H. Thapliyal, M. B. Srinivas and H. R. Arabnia , *"Design And Analysis of a VLSI Based High Performance Low Power Parallel square Architecture"*, in Proc. Int. Conf. Also. Math. Compo. Sc., Las Vegas, June 2005, pp. 72-76.
8. Vinoth, C. Bhaaskaran, V.S.K. Brindha, B. Sakthikumarna, S. Kavinilavu, V. Bhaskar, B. Kanagaasabapathy, M. and Sharath, B. "*A novel low power and high speed Wallace tree multiplier for RISC Processer," 3rd International Conference on Electronics Computer Technology (ICECT), 2011, Vol-1,April 2010,pp. 8-10* .
9. Chen Ping-hua and Zhao Juan, "*High-Speed Parallel 32x32-bit Multiplier Using a Radix-16 Booth Encoder", Third International Symposium on intelligent Information Technology Application Wrokshops, 2009.* IITAW 09, pp.406-409, 21-22 Nov. 2009.
10. Swami Bharati Krshna Tirthaji, *Vedic Mathematics*. Delhi: Motilal Banarsidass Publishers, 1965.
11. Asmita Haveliya *"A Novel Design for High Speed Multiplier .for Digital Signal Processing Applications (Ancient Indian Vedic mathematics approach)"* International Journal of Technology And Engineering System(IJTES):Jan - March 2011- Vo12 .No. l.
12. Harpreet Singh Dhillon and Abhijit Mitra *"A Digital Multiplier Architecture using Urdhava Tiryakbhyam Sutra of Vedic Mathematics"* IEEE Conference Proceedings,2008.
13. Parth Mehta, Dhanashri Gawali *"Conventional versus Vedic mathematical method for Hardware implementation of a multiplier"* 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies.
14. Himanshu Thapliyal, S. Kotiyal and M.B. Srinivas, "*Design and Analysis of a Novel Parallel Square and Cube Architecture Based on Ancient Indian Vedic Mathematics"*, Proceedings on 48th IEEE International Midwest Symp-osium on Circuits and Systems (MWSCAS 2005)
15. Shamim Akhtar, "*VHDL Implementation of Fast NxN multiplier Base on Vedic Mathematics*," Jaypee Institute of Information Technology University, Noida, 2011307 U.P, India, 2007 IEEE.
16. Pushpalata Verma, K.K. Mehta, *"Implementation of an efficient multiplier based on Vedic Mathematics using EDA Tool"*, International Journal of Engineering and Advance Technology (IJEAT) ISSN : 2249-8958, volume-1, Issue -5, June 2012.

Ms. Anju is M. Tech student at IMS Engineering College, Ghaziabad, affiliated to Mahamaya Technical University. She is also associated to Shri Ram Swaroop Memorial Engineering College, Lucknow