ISSN: 0975-8585

Research Journal of Pharmaceutical, Biological and Chemical Sciences

Optimizing Area of Vedic Multiplier using Brent-Kung Adder.

V Anand, and V Vijayakumar*.

Department of Electronics and Communication Engineering, Sathyabama University, Chennai-600119, India.

ABSTRACT

Designing multiplier is one of the interesting and challenging job for satisfies the user requirement as per demand. In various digital systems, power consumption and throughput decides the performance. Vedic multiplier is a most suitable system for faster manipulation and optimized circuit. The inspiration behind the present work is to utilize Vedic multiplier and enhance the operation of multiplication in efficient way in terms of area. Besides, using Urdhva Tiryabhyam, Brent Kung algorithm (BKA) along with Vedic math sutra describe the partial products addition and can also related with the properties of existing algorithms. The BKA assist to parallel generation of partial products and faster carry generation, leading to better area. The Vedic Multiplier code is written in Verilog HDL and synthesized on Xilinx Spartan 3E using Xilinx ISE 9.1i. The propagation delay of the proposed architecture is found to be 22.018ns.

Keywords: VM-Vedic Multiplier, BKA- Brent kung Adder, KSA-Kogge Stone Adder, RCA-Ripple Carry adder, CLA-Carry look-ahead Adder, CSA-Carry Save Adder

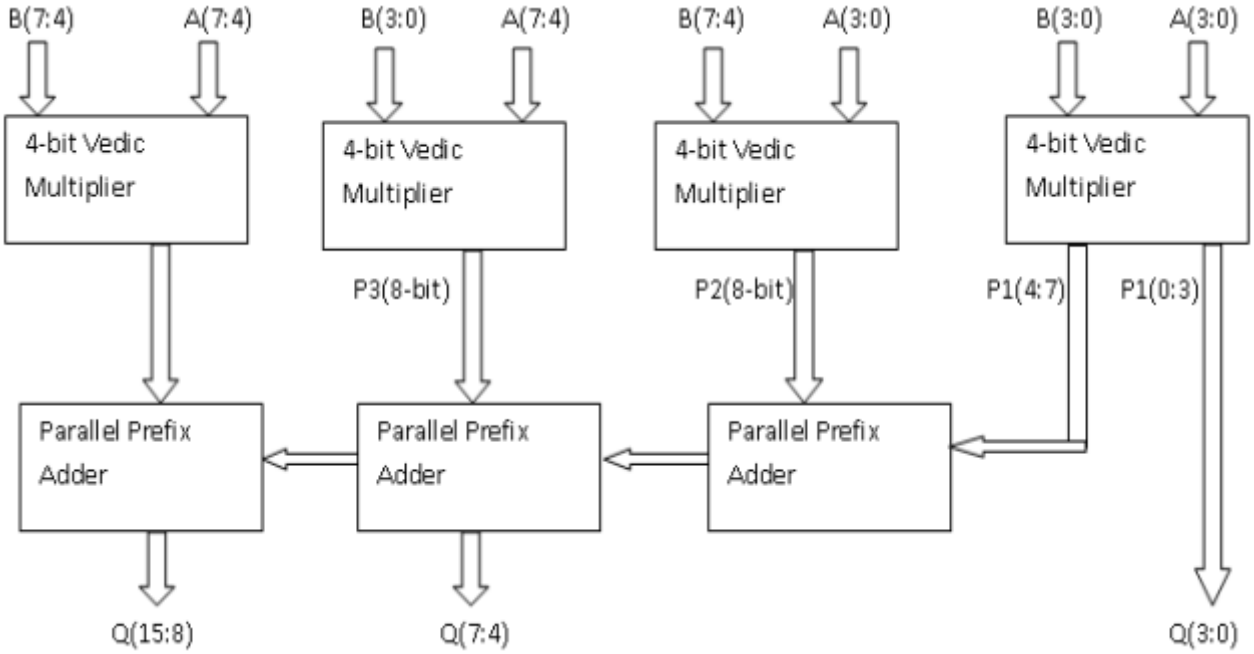
Corresponding author

ISSN: 0975-8585

INTRODUCTION

Fast multiplications and adders are ever required in various digital signal processing systems. Multiplication operations can also be needed for the other complex bases operation such as Fast Fourier Transforms, Discrete Fourier Transform, and convolution. Thus, faster clock frequency is a mandatory arithmetic unit for doing faster operation [1]. Speed and power dissipation of the system is also significant parameters of the system. Vedic multiplier is a fabulous method for less power consumption and fast operation with respect to the size of bit. The proposed Vedic multiplier is implemented from antiquated Indian mathematics where it is represented in Atharva Veda. Shankaracharaya Bharati Krishna Teerthaji Maharaja who is his sanctity Holiness Jagadguru gave a comprised and easiest form of mathematical explanation for different applications. Totally, he constructed sixteen formulae along with sixteen sub formulae when after his wide range of research in Atharva Veda [2] [3]. Figure 1 refers a basic schematic diagram of Vedic Multiplier.

Fig 1. Schematic diagram of Vedic Multiplier.



There are numerous adders efficiently performing the operation in various types of multiplier. Especially, many studies are concentrated in area, power consumption and time [4, 5] which are the primary concern in the present work. Carry Look-Ahead Adder (CLA) consume more power and required large area with respect to size of the bit [6]. Thus, the speed limitation is encountered with respect to bit size. In contrast, the carry-selected adders (CSA) can enhance the speed performance. In CSA the sum will received at output, irrespective of receive carry-in. Therefore, it takes very less time rather than other methods to manipulate the sum [7].

Multipliers forms the key blocks of a Digital Signal processor. Multiplication is the key aspect, whereby improvement in computational speed and area of multiplication decreases the processing time and area of Digital Signal Processors. Convolution transforms, Fast Fourier transforms and various other forms of transforms are making use of multiplier blocks as key component.

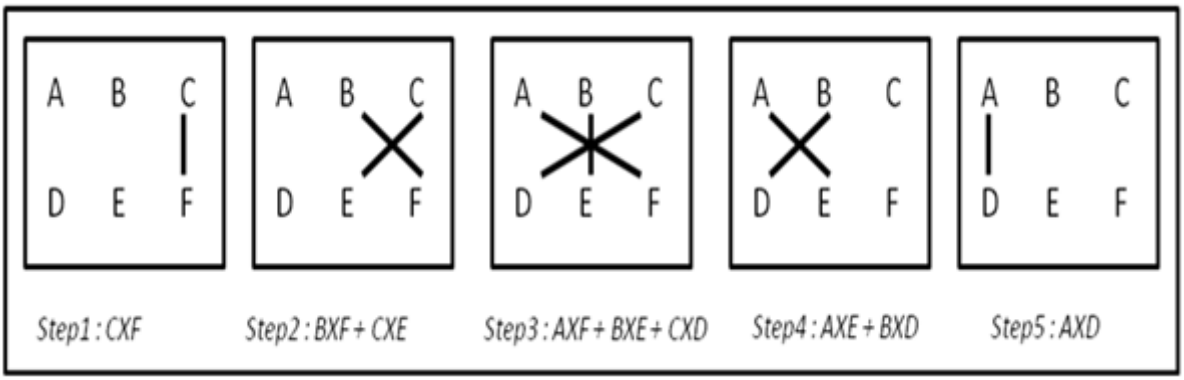
An area efficient method for multiplication based on ancient Indian Vedic mathematics is studied in this paper. Urdhva Tiryagbhyam is efficient [8] among various methods available in Vedic multiplications. It is a generic multiplication formula applicable to all multiplication cases. For addition of partial products in the multiplier Brunt Kung algorithm is used and realized. The code is written in Verilog HDL [9] and synthesized using Xilinx ISE 9.1i.

ISSN: 0975-8585

METHODS

Urdhva Tiryagbhyam

Fig 2. Multiplication steps for Urdhva Tiryagbhya



Consider the upper row ABC as multiplicand and lower row DEF as the multiplier. The multiplication steps are descriptive in the Figure 2. For better understanding the examples are solved below. The intermediate carry generated is getting appended to the very next bit. Figure 3 a and b represents the decimal and binary multiplication illustration of Vedic Technique.

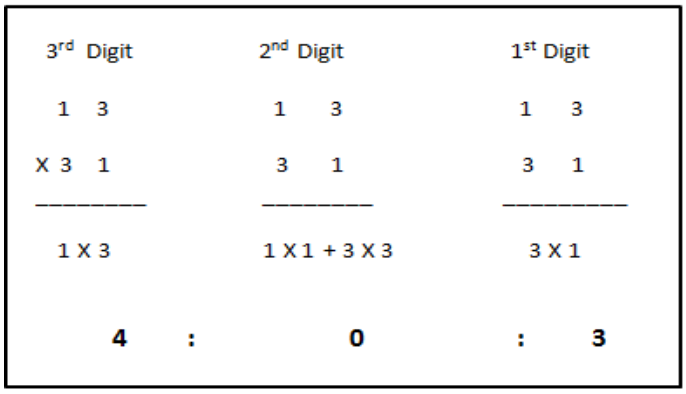


Fig 3 a. Decimal Multiplication Illustration of Vedic technique.

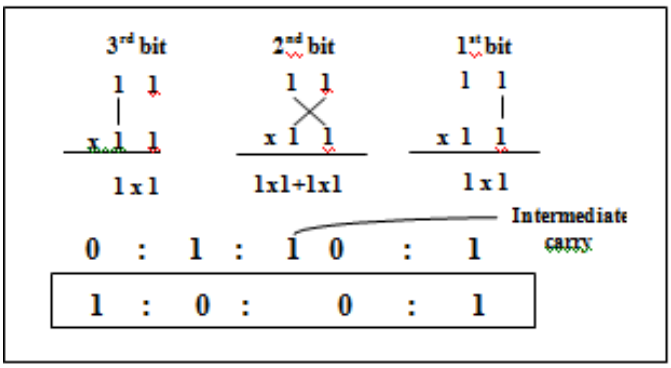


Fig 3 b. Binary Multiplication Illustration of Vedic technique.

ISSN: 0975-8585

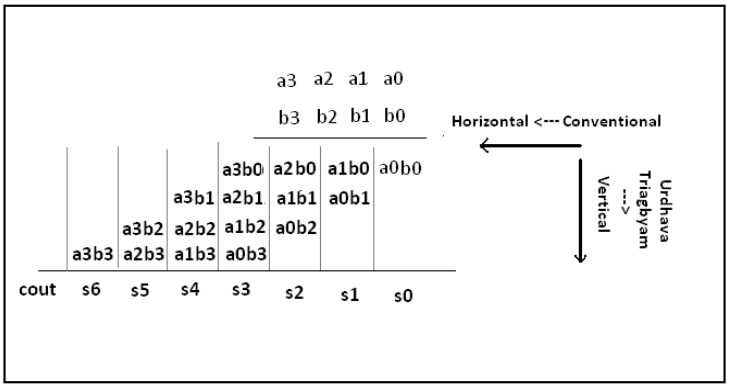


Fig 4. Difference between Vedic technique and conventional Multiplication.

Brent-Kung ALGORTIHM

Brunt Kung algorithm was developed by R P Brent and H T Kung and published in Journal of the Association of Computing machinery in 1978[10]. It generates carry in O(log2 N) time and is used in the industry for area efficient arithmetic circuits considering it to be the adder with reduced area. The area is improved using BKA at the cost of slightly increased delay as shown in Figure 5.

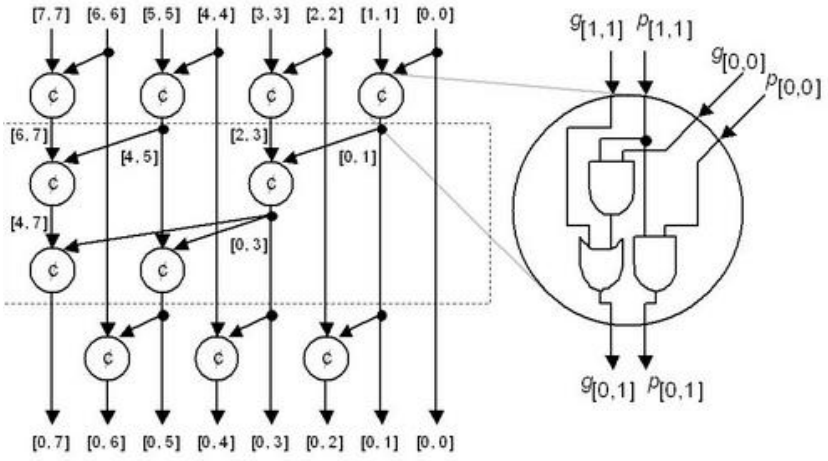


Fig 5. bit Brent Kung Adder network

This is an attempt to apprehend the functioning of BKA in three distinct steps:

Preprocessing

The Preprocessing step involves computation of generate and propagate signals corresponding to each pair of bits in A and B. The logic equations of these signals are given below.

$$P(i) = A(i) \oplus B(i)$$

$$G(i) = A(i) \cdot B(i)$$

ISSN: 0975-8585

Carry look ahead network

This block is responsible for improvement in area. The computations of carries corresponding to each bit involved are carried out in this step. The group propagate and generate are used as intermediate signals given by the logic equations below.

$$Pi{:}j = P(i{:}k+1) \cdot P(k{:}j)$$

$$Gi{:}j = G(i{:}k+1) + (P(i{:}k+1) \cdot G(k{:}j))$$

Post processing

The sum bits are computed in the post processing step using the logic given below.

$$S(i) = P(i) + C(i\text{-}1)$$

Proposed multiplier

The Proposed Vedic multiplier architectures of 2x2, 4x4, and 8x8 bit modules are shown in Figure 6 a-c respectively. The basic architecture was comprehended from the reference paper [11] and modified to obtain the right output as well as reduce area. The major change adopted in the architecture is that we have used Brent Kung algorithm to add partial products rather than KSA, and CSA.

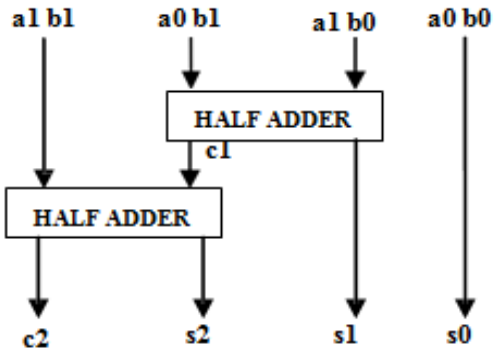


Fig 6a. 2 X 2 Vedic Multiplier Architecture

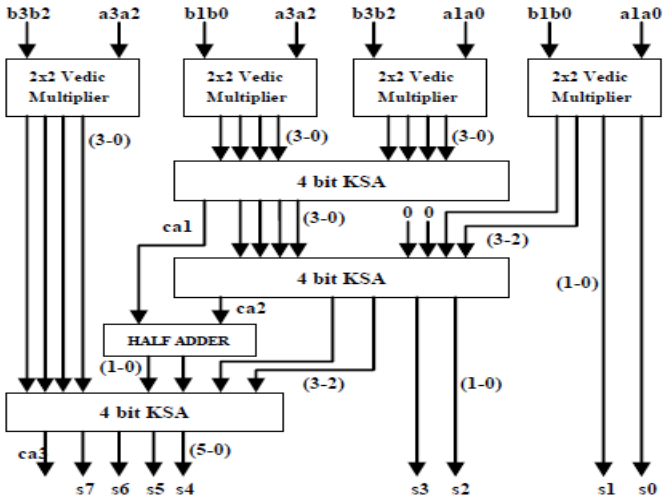


Fig 6b. 4 X 4

ISSN: 0975-8585

Vedic Multiplier Architecture

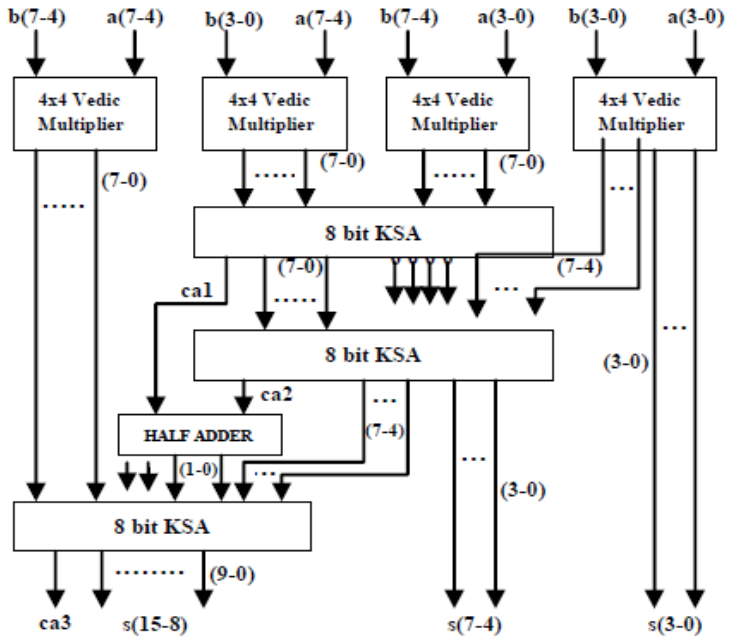


Fig 6c. 8 X 8 Vedic Multiplier Architecture

RESULTS AND SIMULATIONS

The 8x8 Vedic multiplier Verilog code was synthesized using Xilinx ISE 9.1i and simulated on FPGA device xc3s100-5tq144 of SPARTAN 3E Family. Device utilization summary as tabulated in Table 1. As a result, out of 1920, 188 numbers of 4 inputs LUTs is selected with the utilization percentage of 9%. In logic distribution, 99 numbers of occupied slices are utilized. Moreover, number of slices containing related and unrelated logic is 99 and 0 respectively. Totally 33 numbers of bonded IOBs are used for 108 available IOBs. In the proposed multiplier 1143 equivalent gate count and additional 1584 JTAG gate for IOBs utilized for design.

Table 1. Device Utilization Summary of proposed Vedic multiplier device.

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	188	1,920	9%	
Logic Distribution				
Number of occupied Slices	99	960	10%	
Number of Slices containing only related logic	99	99	100%	
Number of Slices containing unrelated logic	0	99	0%	
Total Number of 4 input LUTs	188	1,920	9%	
Number of bonded IOBs	33	108	30%	
Total equivalent gate count for design	1,143			
Additional JTAG gate count for IOBs	1,584			

ISSN: 0975-8585

The simulated output of device utilization summary and timing delay as shown in Figure 7.

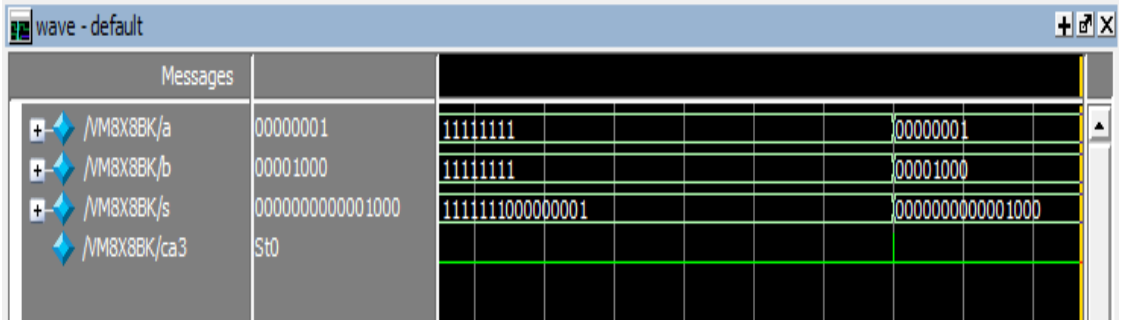


Fig 7. Simulation outputs and Time Details

The results of area and delays between 8x8 modified Vedic multipliers using BKA executed on xc3s100-5tq144 and VM using KSA are validated with existing device are shown in Table.2. It is infers that the area of the proposed device is comparatively low with existing one. 188 LUT and 99 slices are utilized in the proposed work. In the case of existing device built with 224 LUT and 120 Slices which are not optimum for multiplier. Since more power is needed for more area. Delay time for both proposed and existing device has received almost same value with 22.018 and 21.417 ns respectively. It is a good agreement with proposed work, the optimized delay time is observed as same as the existing one with enhanced area. Therefore, VM8x8BKA is a prominent device for manipulating multiplication operation.

Table 2. Comparison of Area and Delay time observed by 8 X 8 VM using KSA and BKA adders

Device	*Xc3s100e-4tq144*	*Xc3s100e-4tq144*
PROGRAM	***LUT & Slices***	***Delay(nS)***
VM8x8KSA[1]	224 &120	21.417
VM8x8BKA	188 & 99	22.018

CONCLUSION

In conclusion, the proposed technique of multiplication using the Vedic technique UrdhvaTiragbyam and Brunt kung algorithm have used less area when compared to available techniques in literature. The delay is found to be 22.018nS when the proposed techniques is simulated for 8X8 multiplication on SPARTAN 3E. The improvement in Area can be achieved for higher bit size when BKA algorithm is adopted. Other pipelining and parallel processing techniques can also be utilized to improve speed. This work will increase awareness about Vedic mathematics techniques in the engineering field and provides area efficient Multipliers for DSP Processors.

REFERENCES

[1] Rakshith Saligram et al., Proceedings of 2013 IEEE Conference on Information and Communication Technologies, 2013; 809-814.
[2] Jagadguru Swami Sri Bharath, Krsna Tirathji, “Vedic Mathematics or Sixteen Simple Sutras From The Vedas”, Motilal Banarsidas, Varanasi (India), 1986.
[3] Ganesh Kumar G et al., International Journal of Scientific and Research Publications, 2012; 2:1-5.
[4] Narasimharao Y et al., JATIT, 2014; 67: 617-625.
[5] Narasimharao Y et al., IJCET, 2014; 4: 1908-1912.
[6] Hoe, DHK. et al., IEEE, 43rd Southeastern Symposium on System, 2011: 168-172.
[7] Desiree JV, International Journal of Scientific & Engineering Research, 2013; 4: 2221-2227.
[8] Ramchandran S, Pande K S, International Journal Of Computational Engineering Research, 2012; 2: 697-703.

[9] Samir Palnitkar. "Verilog HDL, A guide to Digital Design and Synthesis." 2nd edition. Dorling Kindersley and Pearson Education, Inc. 2008.
[10] PushpalataVerma, Mehta K K, International Journal of Engineering and Advanced Technology, 2012; 1: 75-79.
[11] Sudeep M C, Sharath Bimba M,Mahendra Vucha, International Journal of Computer Applications, 2014; 90: 16.