IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 6, June 2016

Implementation of High Speed Multiplier using Fast Parallel Prefix Adder

Pramod Karale[1], R.D Daruwala[2]

Department of Electrical Engineering, Veermata Jijabai Technological Institute (VJTI), Mumbai, India[1, 2]

Abstract: With the rapid advancement in information technology, there is a huge demand for high speed processors. This has led to design of advanced processors capable of performing all mathematical computations at a faster speed. Multiplication is one such dominating computational technique which plays an important role as far as speed is concerned. In this paper, design and implementation of multiplier using Vedic mathematics named as Urdhva Triyakbhyam which incorporates Kogge Stone adder is presented. Adder is a parallel prefix derived from carry look-ahead adder which is the faster. This multiplier is implemented on Xilinx Board is shown to have exhibited high performance in terms of speed in the simulations.

Keywords: Urdhva Triyakbhyam, Kogge Stone adder, Xilinx Board.

I. INTRODUCTION

As the need of high speed processors is increasing, need to design fast multipliers is the key challenge required to be met. The speed of any processor is greatly depends on the computational speed of the functional blocks used in the system design. Multiplier is one of the key hardware blocks used in almost all the general purpose processors and the digital signal processors. Nowadays various multipliers are available like combinational multipliers, array multipliers, booth multiplier.

The two- bit multiplication M x N involves formation of N partial products of M bits each thereby summing the appropriately shifted partial products, subsequently to produce M+N bit result [1]. In this work, the high speed multiplier [1] is developed by Vedic Mathematics which is based on Vertical and Crosswise structure known as Urdhva Tiryakbhyam (UT) along with Kogge Stone adder.
Urdhva Tiryakbhyam Sutra is an alternate method to perform multiplication is referred from [1] which is applicable to all cases.

The architecture of multiplier includes the Kogge Stone adder. Peter M. Kogge and Harold S. Stone [6] developed the Kogge Stone adder .The Kogge Stone adder is a parallel prefix form of carry look-ahead adder. The time complexity for the carry signals generated is O (log n), thereby making it as a widely considered one because of its fast nature of performance [2]. This design is therefore commonly adopted in industry for achieving better performance and the same has been produced in this work as well.

The organization of the remaining research work is as follows: The organization of the remaining research work is as follows: Section II reviews the related works of various design implementations of multipliers. In Section III, design and implementation method are explained. Simulations and results are discussed in Section IV. Finally, conclusion is given in Section V.

II. RELATED WORKS

Previous work reveals that various multiplier designs and implementations have been proposed to enhance the speed of processors. Parallel multipliers were developed way back in 1960. The configuration of this multiplier [2] [3] [4] consist of rectangular array of identical combinational cells that generate and also sum the partial product bits. They are also known as Array multipliers. For unsigned N × N multiplication, an array multiplier defines two basic functions, partial product generation and summations which are combined. In this case, N^2 + N - 1 cells are connected to produce a multiplier (where N^2 contain an AND gate for partial product generation and N - 1 cells containing a full adder used for summing).N lower product bits are generated whereas the upper N bits of the product are formed by using a carry-propagate adder, which is a ripple carry adder.

Due to the regularity of their structures, array multipliers are the most frequently implemented structures. The literature also there exists another class of parallel multipliers which incorporates strategic application of counters or compressors [2] that has the capability to reduce a matrix of partial product bits to two words. These two words are then summed using a fast carry-propagate adder to generate the product. Hence this class of parallel multiplier is known as column compression multiplier. They are considered as the fastest multiplier as the delay is directly proportional to the logarithm of the multiplier and word length.

According to Thomas & Callaway et.al. [15] Column compression multipliers are more power efficient than array multipliers. In 1964, Wallace [14] introduced a scheme called as Wallace Tree which is a tree of carry save adders that perform summation of partial product bits for fast multiplication. Wallace's method was later improvised by Dadda [16] where a counter placement strategy was introduced that required fewer counters in the partial product reduction stage at the cost of a larger carry-

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 6, June 2016

propagate adder. For both methods, the total delay is proportional to the logarithm of the operand word-length. Other partial product reduction methods have been proposed since the work of Wallace and Dadda. While maintaining the fast speed of the Wallace and Dadda design the Windsor methods were later introduced which were based on strategic utilization of (3, 2) and (2, 2) counters aimed to improve area and layout.

In order to perform multiplication with 2's complement operands using array multiplier, booth algorithm [12] can be employed. This algorithm computes the partial products by examining two multiplicand bits at a time, but provides no advantage in terms of area reduction and delay. Better delays, though can be achieved by implementing a higher radix modified Booth algorithm [13]. Another method [17] which increases the maximum column height by two, handles 2's complement operands in an array multiplier. This may lead to an additional stage of partial product reduction, thereby increasing overall delays.

III. DESIGN AND IMPLEMENTATION

The algorithm for multiplication is adopted from Vedic Mathematics called as Urdhva Tiryakbhyam (UT). In the architecture of multiplier the partial products are added using Kogge Stone adder. The proposed architecture of 16 x 16 bit Vedic Multiplier is as shown below.

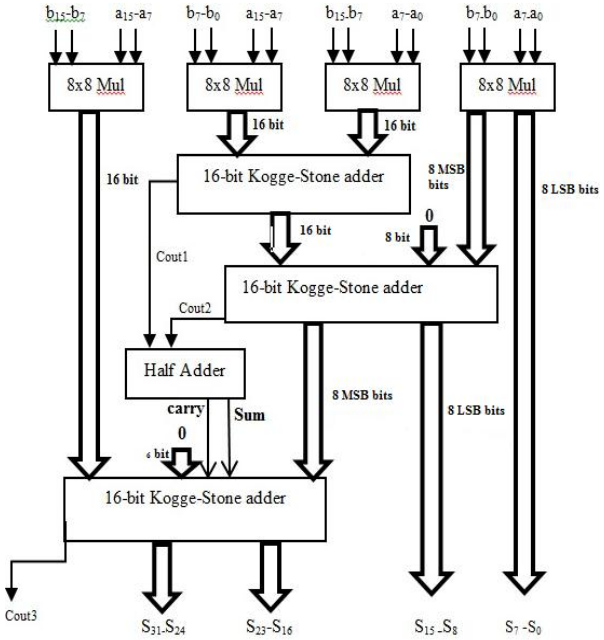

Fig.1: Architecture of Multiplier

The basic architecture is comprehended from the base paper [5] and modified to obtain expected output as well as speed.

The architecture design consists of simple and smaller blocks to synthesize a 16 x 16 bit multiplier. For example a 8 x 8 bit multiplier block can be synthesized by using four 4 x 4 multiplier blocks along with three 8-bit Kogge Stone adders, similarly a 4 x 4 bit multiplier can be

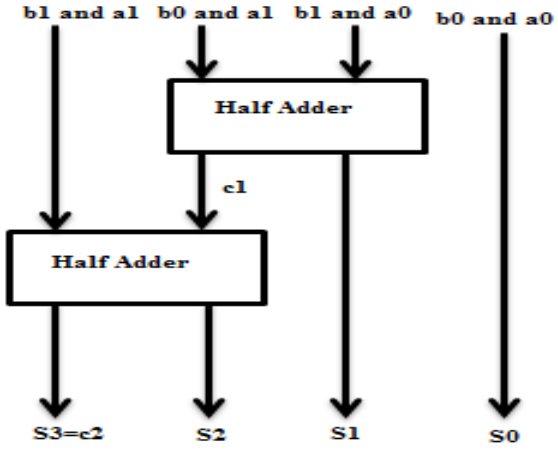

Fig.2: Basic 2 x 2 Vedic Multiplier Block

designed using four 2 x 2 multiplier along with three 4-bit Kogge Stone adders. Functionality of each block is verified using Xilinx simulation software.

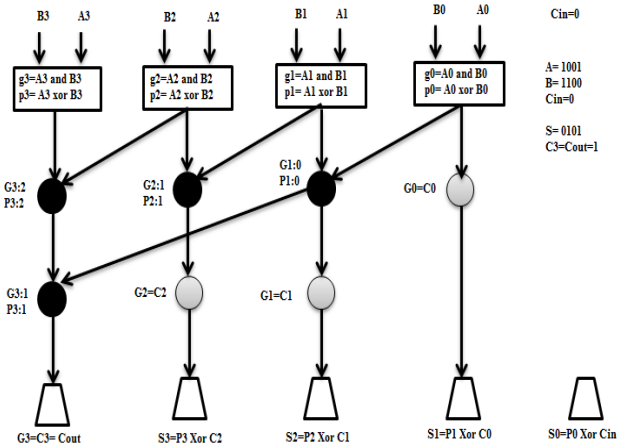

Fig.3: 4-bit Kogge Stone adder

The internal processing of Kogge Stone adder involves the following steps:

- **Preprocessing:** Generate (p_i ,g_i) from (A_i , B_i)

$$p_i => A_i \text{ xor } B_i \quad (1)$$

$$g_i => A_i \text{ and } B_i \quad (2)$$

- **Carry look ahead network:** Generate ($P_{i:pre}$, $G_{i:pre}$) from (g_i, p_i) and (g_{pre}, p_{pre})

$$P_{i:pre} => p_i \text{ and } p_{pre} \quad (3)$$

$$G_{i:pre} => g_i \text{ or } (p_i \text{ and } g_{pre}) \quad (4)$$

- **Post processing:** Generate Sum_i from(p_i ,$Carry_{i-1}$)

$$Sum_i = p_i \text{ xor } Carry_{i-1} \quad (5)$$

For the implementation of multiplier, the code is written in VHDL and simulated in Xilinx Software and later implemented on Xilinx Board.

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
Vol. 3, Issue 6, June 2016

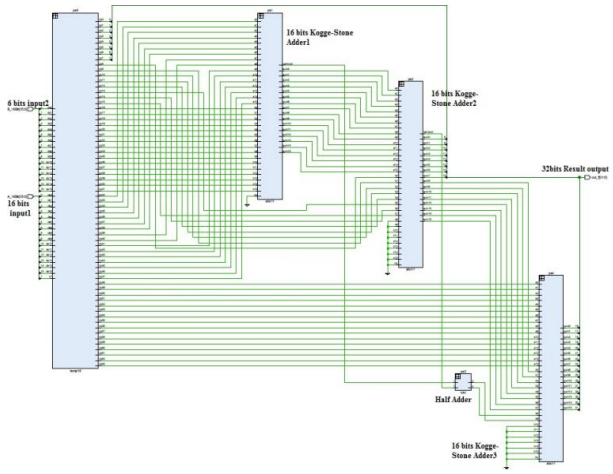

Fig.4: RTL Schematic

IV. SIMULATION AND RESULTS

A result has been drawn out from the simulation experiment performed with Kogge Stone adder in terms of computation time which is as shown in Table 1

Table 1: Comparison of delay times of different Kogge Stone adder.

Kogge Stone Adder (KSA)	Time(ns)
4-bit	1.179
8-bit	1.971
16-bit	3.457

Along with the Kogge Stone adder, the processing time of the above Multiplier has been measured which essentially give a clear indication of the high speed performance of Vedic multiplier using Kogge Stone adder which is as shown in Table 2

Table 2: Performance evaluation of Vedic multiplier

Vedic multiplier using_Kogge Stone adder	LUT'S Used	Total LUT'S Present	% area Utiliza -tion	Time (ns)
4-bit	27	53200	0.051	3.718
8-bit	140	53200	0.26	6.179
16-bit	715	53200	1.329	10.817

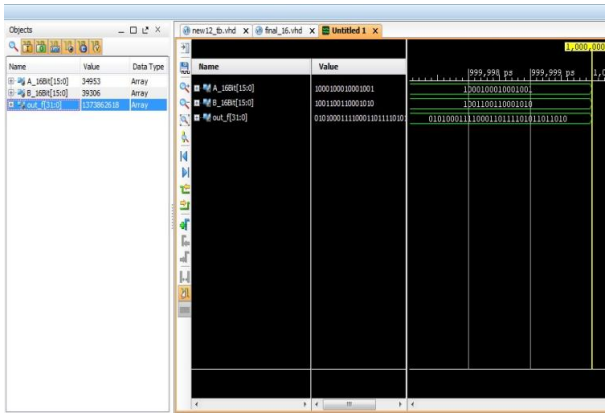

Fig.5: Simulation Result 16x16 Vedic Multiplier

The Simulation result for 16 bit multiplier is shown in Fig 5 in which input A =>34953 and input B =>39306 is taken and result out =>1373862618 is obtained.

V.CONCLUSION

In this paper high speed architecture for 16 x 16 bit multiplication is proposed by extracting the features of Vedic multiplier based on UT method and Kogge Stone adder. As Kogge-Stone adder is used as the main building block in the proposed multiplier architecture design, the total delay time and the execution time is considerably reduced, indicating a high speed performance of the multiplier and also able to reduce the area utilization factor drastically by using a specific Xilinx Simulation Software and Xilinx board.

REFERENCES

[1] R. Anjana, B. Abishna, M. S. Harshitta, E. Abhishek, V. Ravichandra and M. S. Suma, "Implementation of Vedic Multiplier using Kogge Stone Adder," IEEE International Conference on Embedded Systems, pp. 28-31, 2014.

[2] P. P. Zode and R. B. Deshmukh, "Fast Modular Multiplication using Parallel Prefix Adder," IEEE conference on Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN), pp. 1-4, 2014.

[3] N. Anand, G. Joseph, J. S. Raj and P. Jayakrishnan, "Implementation of adder structure with fast carry adder network for high speed processor," IEEE conference on Green Computing, Communication and Conservation of Energy (ICGCE), pp. 188-190, 2013.

[4] M. Khanand and K. Han, "An optimized network selection and handover triggering scheme for heterogeneous self-organized wireless networks," Mathematical Problems in Engineering, Article ID173068, pp. 11, 2014.

[5] M. Poornima, Shivaraj Kumar Patil, Shivukumar, K. P. Shridhar and H. Sanjay, "Implementation of Multiplier using Vedic Algorithm," MVJCE, Bangalore, May 2013.

[6] P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a General class of Recurrence equation," IEEE transactions on Computers, Vol.22 No.8, pp. 786-792, August 1973.

[7] P. Ramanathan, P. T. Vanathi, "A Novel Logarithmic Prefix Adder with Minimized Power Delay Product," Journal of Scientific & Industrial Research, Vol. 69, pp. 17-20, January 2010.

[8] R. Ladner and M. Fischer, "Parallel prefix computation," Journal of ACM. La Jolla, vol.27, no.4, pp. 831-838, October 1980.

[9] David Harris, "A Taxonomy of parallel prefix networks," Proceedings of the 37th Asilomar Conference on Signals, Systems and Computers Pacific Grove, California, pp. 2213-2217, November 2003.

[10] O. J. Bedrij, "Carry-Select Adder," IRE Transaction on Electronics Computers, vol.EC11, pp. 340-346, 2011.

[11] D. H. K. Hoe, C. Martinez, and J. Vundavalli, "Design and Characterisation of Parallel Prefix Adders using FPGAs," IEEE 43rd Southern Synopsium on System Theory, pp. 168-172, March 2011.

[12] D. Purushottam, Chidgupkar and M. T. Karad, "The Implementation of Vedic Algorithms in Digital Signal Processing," Global Journal of Engineering. Education, vol.8, No.2, pp. 153-157, 2004.

[13] A. D. Booth, "A signed binary multiplication Technique," Quarterly Journal of Mechanics and Applied Mathematics, vol.4, pp. 236-240, 1951.

[14] S. R. Kuang, J. P. Wang and C.-Y. Guo, "Modified booth multipliers with a regular partial product array," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 56, no. 5, pp. 404–408, May 2009.

[15] C. S. Wallace, "A suggestion for a fast multiplier," IEEE Transactio ns on Electronic Computers, vol. EC-13 ,no. 1, pp. 14–17, Feb. 1964.

[16] T. K. Callaway and E. E. Swatzlander, "Optimizing multipliers for WSI," International Conference on Wafer Scale Integration, pp. 85-94, 1973.

[17] L. Dadda, "Some schemes for parallel multipliers," Alta Frequenza, vol. 34, pp. 349-356, August 1965.

[18] C.R. Baugh and B. A. Wooley, "A two's complement parallel array multiplication algorithm," IEEE Transactions on Computers, vol. C-22, pp. 1045-1047, 1973.