International Journal of Research in Advent Technology, Special Issue, RAECE-2K19
E-ISSN: 2321-9637
Available online at www.ijrat.org

Design and Implementation of 64 Bit Vedic Multiplier Using Adders (Verilog HDL)

[1]P.S.Sreenivasa Reddy,[2] B.Suneetha
[1,2]Associate Professor,
Nalla Narasimha Reddy Group of Institutions,
ps_reddy1@rediffmail.com,Suneetha.b@nnrg.edu.in

Abstract: This Paper describes the design of high speed Vedic multiplier that uses the techniques of Vedic mathematics based on 16 sutras (algorithms) to improve the performance. In this work the efficiency of Urdhva-Tiryagbhyam (vertical and crosswise) Vedic method for multiplication which is different from the process of normal multiplication is presented. Urdhva-Tiryagbhyam is the most efficient algorithm that gives minimum delay for multiplication for all types of numbers irrespective of their size. Vedic multiplier is coded in Verilog HDL and stimulated and synthesized by using XILINX software 14.7v ISE on Spartan 3E kit. Further the design of Vedic multiplier is design with Hybrid adder and compared with the proposed multiplier in terms of delay, power and device utilization.

Keywords - Vedic mathematics, Vedic multiplier, Urdhva-Tiryagbhyam, Ripple Carry Adder (RCA), Binary to Excess Code Converter (BEC), Half Adder (HA), Full Adder(FA), Carry Select Adder (CSLA), parallel Prefix adders, Brent Kung adder.

1. INTRODUCTION

Multiplication operation is intensively used in digital signal processing applications, so there is a need for high speed multiplier. This paper presents a systematic design methodology for fast and area efficient multiplier based on Vedic mathematics [1]. The Multiplier Architecture is based on the Vertical and Crosswise algorithm of ancient Indian Vedic Mathematics. The adder block used in Vedic multiplier is the main source of delay in overall multiplication operation. The organization of this paper is as follows. Section-II presents logic for Vedic Algorithm based multiplication operation. Section-III deals with different adder topologies. Section –IV Explain The parallel prefix adders. Section -V about the Brent kung adder. Section -VI Simulation results followed by conclusion and references.

For example, if A=0000001010101101, then it is divided as

00000010, and 10101101.Similarly if B= 0010001110101011, then it can be divided as 00100011 and 10101011. Representing the divided parts of A as AM and AL. Similarly for the input B, it is divided in two parts as BM and BL. A and B can be represented as AMAL and BMBL. Multiplication operation between A and B can be represented as:

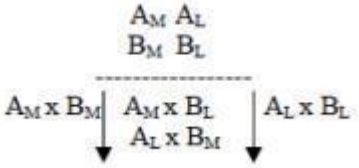

Figure1: Cross Multiplication Representations

These partial products can be generated and added as per structure proposed in. The block diagram for 16x16 Multiplication based on Vedic technique as proposed in [1] is shown in Figure 2. The first stage is four sets of 8x8 Multiplier which is also based on Vedic technique

2. VEDIC ALGORITHM BASED MULTIPLICATION

The generalized structure of Vedic multiplier is proposed in [1]. The basic module is 2x2 Multiplier. For N x N multiplication, divide the multiplicand and multiplier into two parts, consisting of (N to N/2-1) bits and (N/2 to 1) bits.

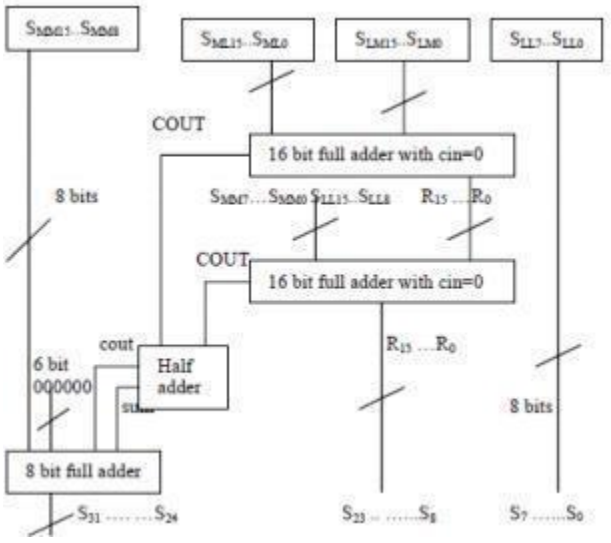

Figure 2: Block Diagram for 16-bit Vedic Multiplication

As can be seen that adder is used in the intermediate computation, hence fast and area efficient design will enhance the overall performance of multiplication. In next section, different adder topologies are discussed which are used to design Vedic multiplier for different input bit length.

3. DIFFERENT TOPOLOGIES FOR ADDERS

In literature different architectures of adders namely Ripple Carry Adder (RCA) [2, 3], CSA [2, 4-11], CBL [2, 7] and SQRT-CSA, Modified SQRT-CSA [6] is available. A brief discussion on above adders is given below for clarity. RCA is basic parallel adder where a chain of adders are cascaded with carry rippled from one stage to another. In CSA, each stage consists of two ripple carry adders and a set of multiplexers. Based on previous carry input, the current sum and carry for the next stage is computed. As the bit length of input data to the adder increases, the number of stages of CSA also increases. In linear CSA, equal number of input bits is used in all the stages.

Later SQRT-CSA was proposed where the progressively increasing input bit size was used for different stages. It was found that delay distribution in SQRTCSA was much better than that of linear CSA. Later Modified SQRT-CSA was proposed in which CSA with Cin =1 was replaced by BEC (Binary to Excess-1 Converter) [5, 6].

We have used std_cell based approach where synthesis tool use predesigned optimized cell for the realization. Next section deals with the analysis of Vedic multiplier based on different adder topologies.

4. PARALLEL PREFIX ADDERS

Parallel prefix adders are used to speed up the binary additions as they are very flexible. The structure of Carry Look Ahead Adder (CLA) is used to obtain parallel prefix adders. Tree structures are used to increase the speed of arithmetic operation. Parallel prefix adders are used for high performance arithmetic circuits in industries as they increase the speed of operation. The construction of parallel prefix adder involves three stages:

i. Pre- processing stage

ii. Carry generation network

iii. Post-processing stage

i. Pre-Possessing Stage

Generate and propagate signals to each pair of inputs A and B are computed in this stage.

These signals are given by the following equations:

$$Pi=Ai \text{ xor } Bi \qquad (4.1)$$

$$Gi=Ai \text{ and } Bi \qquad (4.2)$$

ii. Carry generation network

In this stage, we compute carries equivalent to each bit. Implementation of these operations is carried out in parallel. After the computation of carries in parallel they are segmented into smaller pieces. Carry propagate and generate are used as intermediate signals which are given by the logic equations 4.3&4.4

$$CPi:j = Pi:k+1 \text{ and } Pk:j \qquad (4.3)$$

$$CGi:j = Gi:k+1 \text{ or } (Pi:k+1 \text{ and } Gk:j) \qquad (4.4)$$

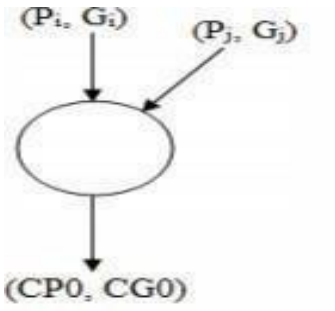

Fig. 3: Carry

Network iii. Post processing stage

This is the concluding step to compute the summation of input bits. It is common for all the adders and the sum bits are computed by logic equations 4.5&4.6:

$$Ci-1= (Pi \text{ and } Cin) \text{ or } Gi \qquad (4.5)$$

$$Si=Pi \text{ xor } Ci-1 \qquad (4.6)$$

5. BRENT-KUNG ADDER

Brent-Kung adder is a very well-known logarithmic adder architecture that gives an optimal number of stages from input to all outputs but with asymmetric loading on all intermediate stages. It is one of the parallel prefix adders Parallel prefix adders are unique class of adders that are based on the use of generate and propagate signals. The cost and wiring complexity is less in Brent kung adders.

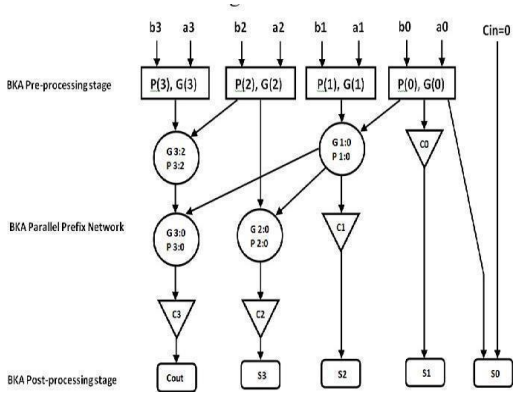

Fig. 4: 4 Bit Brent Kung Adder

6. SIMULATION RESULTS

All the synthesis and simulation consequences are achieved the usage of Verilog HDL. The synthesis and simulation are completed on Xilinx ISE 14.7. The simulation outcomes are shown beneath figures. The corresponding simulation consequences of the 64bit Vedic multiplier with hybrid adder are shown in below.

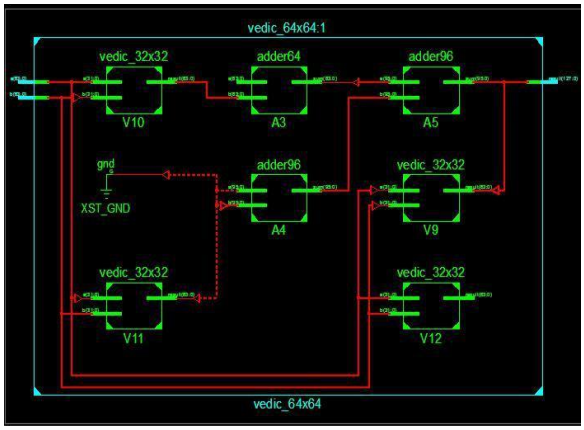

Fig 5. RTL Schematic of 64 Bit Vedic Multiplier

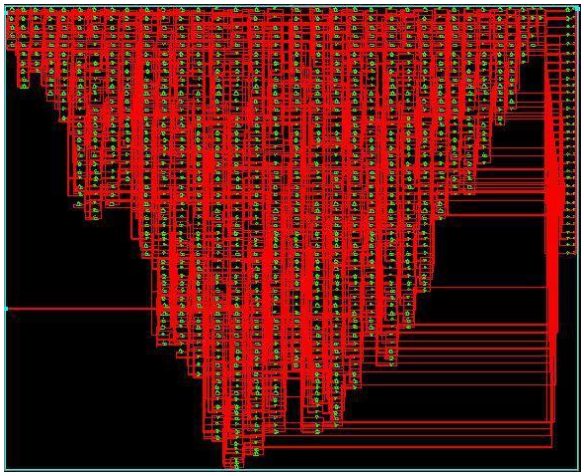

Fig 6. Technology schematic

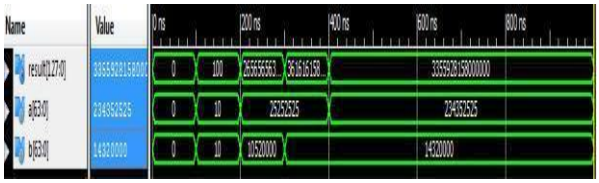

Fig 7. Simulation result

```
Device utilization summary:
---------------------------

Selected Device : 3s1200efg320-4

 Number of Slices:                     5553  out of   8672    64%
 Number of 4 input LUTs:              10130  out of  17344    58%
 Number of IOs:                         256
 Number of bonded IOBs:                 256  out of    250   102% (*)
```

Fig 8. Device utilize summary

```
Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 15881979452305582 / 128
-------------------------------------------------------------------------
Delay:               47.081ns (Levels of Logic = 148)
  Source:            a<1> (PAD)
  Destination:       result<127> (PAD)
```

Fig 9. Timing details

Fig 10. Power consumption

7. CONCLUSION

From the above results it is clear that for 8bit, memory utilized for Vedic multiplier using Hybrid (138728KB) is less when compared to Vedic multiplier with normal serial adders (198568KB).Similarly in the case of 16bit, memory utilized for Vedic multiplier using hybrid adder (139624KB) is less when compared to normal Vedic multiplier with normal serial adder (208268KB). By comparing the values of both Vedic multiplier with hybrid adder and serial adder it is clear that the delay for Vedic multiplier with hybrid adder is much less when compared with Vedic multiplier with normal serial adder. As we increase number of bits delay can be reduced by using Vedic multiplier with hybrid adder than Vedic multiplier with serial adder.

REFERENCES

[1] S. Akhter, "VHDL Implementation of Fast NxN Multiplier Based on Vedic Mathematic", Proc. of 18th European Conference on Circuit Theory and Design (ECCTD), pp. 472-475, Aug. 2007.

[2] Vikas K. Saini, Shamim Akhter, and Tanuj Chauhan, "Implementation, Test Pattern Generation, and Comparative Analysis of Different Adder Circuits", VLSI Design, Vol. 2016, Article ID 1260879, 8 pages, 2016. doi:10.1155/2016/1260879.

[3] N. H. E. Weste, D. Harris, and A. Banerjee, "CMOS VLSI Design: A Circuits and Systems Perspective", Pearson Education, 3rd Edition, 2005.

[4] Basant Kumar Mohanty and Sujit Kumar Patel "Area-Delay-Power Efficient Carry-Select Adder" ,IEEE Transaction On Circuits and Systems-II Express Briefs, Vol-61, No.6, June 2014.

[5] O. J. Bedrij, "Carry-select adder," IEEE IRE Transactions on Electronic Computers, Vol: EC – 11, pp. 340–344, 1962.

[6] S.Akhter, S. Chaturvedi, and K.Pardhasardi, "CMOS Implementation of Efficient 16-Bit Square Root Carry-Select Adder",2nd International Conference on Signal Processing and Integrated Networks (SPIN), India, Noida, pp. 891 – 896, Feb 2015.

[7] B. Ramkumar, and H. M. Kittur, "Low-Power and Area-Efficient Carry Select Adder", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 20, No. 2, pp. 371-375, Feb. 2012.

[8] G.Singh , "Design of Low Area and Low Power Modified 32-BIT Square Root Carry Select Adder", International Journal of Engineering Research and General Science Vol 2, No. 4, pp-422-431, 2014.

[9] I-Chyn Wey, Cheng-Chen Ho, Yi-Sheng Lin, and Chien-Chang PengAn "Area-Efficient Carry Select Adder Design by Sharing the Common Boolean Logic Term",International Multiconference of Engineers and Computer Scientists, Vol 2, March 2012.

[10] Yajuan He, Chip-Hong Chang and Jiangmin Gu "An Area Efficient 64- bit Square Root Carry-select Adder for Low Power Applications", IEEE International Symposium on Circuits and Systems", Vol.4, pp.4082 – 4085, May 2005.

[11] V. Kokilavani, K. Preethi, and P. Balasubramanian "FPGA-Based Synthesis of High-Speed Hybrid Carry Select Adders", Advances in Electronics, Volume 2015, Article ID 713843, 13 pages doi: /10.1155/2015/713843.