DOI: 10.21917/ijme.2016.0034

VLSI IMPLEMENTATION OF HIGH SPEED AREA EFFICIENT ARITHMETIC UNIT USING VEDIC MATHEMATICS

K.N. Vijeyakumar[1], S. Kalaiselvi[2] and K. Saranya[3]

[1,3]*Department of Electrical and Electronics Engineering, Dr. Mahalingam College of Engineering and Technology, India*

E-mail: [1]vijey.tn@gmail.com, [3]miles2gosaran@gmail.com

[2]*Department of Electronics and Communication Engineering, Dr. Mahalingam College of Engineering and Technology, India*

E-mail: [2]kalaisundaram1990@gmail.com

Abstract:

High speed Arithmetic Units (AUs) are widely used in architectures used in signal and image processing applications. AUs involve multi-functions and have multiplier as the critical element. In this paper, we present design and implementation of high speed and area efficient AU using Vedic algorithm. The work uses a simple "vertical and crosswise sutra" of Vedic mathematics to produce low complexity Partial Product (PP) generation unit in multiplier which reduces critical delay. Implementation results using TSMC 180 nm CMOS process with CADENCE Encounter Digital Implementation of the proposed AU revealed delay and Area-Delay Product (ADP) reductions of 13.7% and 19.2% respectively compared to prior recent approaches.

Keywords:

Vedic Multiplier, Urdhva Triyagbhyam Sutra, Arithmetic Unit, High Speed Multiplier

1. INTRODUCTION

With the enrichment of new technology in the field of VLSI design to enhance the ability of processors for handling complex and challenging processes has resulted in the integration of a number of processor cores into one chip. Still the load on the processor is not less in generic system. This load is reduced by supplementing the main processor with co-processors, which are designed to work upon specific type of functions like numeric computation, signal processing, graphics etc. [1]. The main functional unit of these processing architectures is arithmetic unit. It is a fundamental building block of the Central Processing Unit (CPU) and performs arithmetic operation viz., addition, subtraction, multiply and accumulation. The speed of AU greatly limits the speed of the processing architecture and it depends mainly on the multiplier used, so there is a great demand for high speed and low power multiplier.

In addition the increased complexity of various applications, demands not only faster multiplier chips but also smarter and efficient multiplying algorithms that can be implemented in the chips. Two most common multiplication algorithms followed in the digital hardware are viz., array and booth multiplication algorithms. The drawback of these two algorithms is large propagation delay and increased hardware complexity [2]. So the use of these multipliers in AU designs increases either hardware complexity or propagation delay.

The rest of the paper is organized as follows. Section 2 presents an insight into the various related works on AU design. Section 3 presents an overview of the Vedic algorithms used for fast arithmetic operations. Section 4 evaluates the logic depth involved in conventional and UTS-Vedic multipliers. The details of the proposed AU are discussed in section 5. Section 6 presents the simulation results of the proposed AU. Final section 7 briefs the conclusion of the proposed work.

2. RELATED WORKS

To circumvent, a number of approaches and algorithms are proposed for design of arithmetic and multiplier units. Rong Lin [3] proposed a novel approach for inner product processor design. The proposed approach can be easily reconfigured for computing inner products of high bit width input arrays. Besides the use of parallel counters leads to, few control bits for computations and requires few instruction cycles for reconfiguration. Rong Lin's approach performs better in terms of area and power reductions with improved speed compared to prior similar approaches.

Wen-Chang Yeh and Chein-Wei Jen [4] proposed a novel high speed multiplication algorithm using modified Booth technique. The use of MBE algorithm and conditional adders at the final stages of PP reduction in the proposed design reduces number of PP rows to half and show significant delay reduction respectively. However the use of MBE technique increases hardware and routing complexity.

Lan-Da Van and Chih-Chyau Yang [5] developed a novel methodology for design of area-efficient Fixed-Width (FW) multipliers. The proposed design uses efficient truncation algorithms to produce low-error area-efficient FW designs. The implementation results with UMC 0.18-um CMOS process, reveals better performance of the proposed design compared to prior arts. However these designs cannot be used in precision computing architectures.

Higinio Mora-Mora et al. [6] in a novel approach proposed a multiplier unit with reduced PP generation. The design uses Look-Up Tables (LUT) and counter elements. The advantage of the proposed technique is better improvement in speed and reduced resource utilization, when implemented on FPGA series boards. Jin-Hao Tu and Lan-Da Van [7] proposed a design for pipelined reconfigurable FW multiplier. The inclusion of the gated clock in the proposed design reduces switching and power dissipation.

Chalamalasetti et al. [8] proposed a novel coarse grain reconfigurable MORA architecture. The feature of the proposed architecture is that it achieves high throughput, speed and low power. Sotiris Xydis et al. [9] proposed a design methodology for reconfigurable AU. The insertion of flexibility into the computational resources of custom AU is done to provide operation re-configurability. The design when implemented on FPGA demonstrates low resource utilization, delay, power efficiency and reconfiguration compared to previous approaches mentioned in the literature.

Shanthala and Kulkarni [10] proposed a Block Enabling Technique (BET) for the design of low power Multiply and Accumulate (MAC) Unit. The novelty of the proposed technique is that during each multiply and addition, the blocks in the pipeline are enabled only when the data is available from the previous stage. This in turn enables few blocks in the pipeline thus saving power. The proposed approach is implemented for an 8 bit granularity using Cadence Virtuoso design environment.

Khader Mohammed et al. [11] in a novel approach introduced Parameterized Digital Electronic Arithmetic (PDEA) for design of AU. Simulation results revealed that the proposed approach demonstrated better accuracy and speed. Implementation results on Xilinx FPGA Saprtan-3E revealed fewer logic cell utilization and low power dissipation. Manoranjan Pradhan et al. [12] proposed a novel MAC unit using vedic algorithm suitable for high performance processing elements. The proposed design when implemented on Virtex Spartan 6, demonstrated better speed improvement.

Harish Kumar and Hemanth Kumar [13] proposed a technique for design of multiplier unit using Vedic algorithm. The proposed technique when implemented with Xilinx 10.1 ISE, provided reduced delay and power consumption compared to previous approaches. In a novel multiplier design for AUs, Virendra B. Magar [14] utilized Urdhva Triyabhayam Sutra (UTS) of Vedic algorithm, which enables parallel generation of partial products. The proposed multiplier on implementation in Xilinx Spartan 3 FPGA board revealed improved speed grade and area, when compared to array and booth multipliers.

Purohit et al. [15] proposed two new high performance AUs for floating point applications. The novelty of the proposed methodology is that, it can be expanded for higher bit widths of input operand with ease. An implementation of proposed design on FPGA revealed reduced resource utilization compared to prior arts. In this paper we present the architecture of reconfigurable integer data path which can perform addition, subtraction, multiplication and accumulation operations. The proposed data path performs fast multiplication by subdividing inputs into two clusters and performing four $n/2 \times n/2$ multiplications in parallel. The multiplier is configured and input multiplexed to perform the addition and subtraction operations whereas the accumulation operation needs an extra adder. The proposed architecture has uniform critical path across the operations and easily extendable for higher bit widths. Eight bit version of the proposed architecture is implemented with TSMC 180-nm technology process.

3. BASICS OF VEDIC MATHEMATICS

Vedic mathematics is the name given to the ancient Indian system of mathematics mainly used for fast arithmetic operations, discovered in early twentieth century. Vedic mathematics is based on sixteen principles or word-formulae which are termed as Sutras, or aphorisms, or algorithms and their upa-sutras or corollaries derived from these Sutras [16].

The sutras find extensive application in solving mathematical problems viz., algebra, arithmetic, geometry or trigonometry with ease. In the proposed approach we use Urdhva Tiryakbhyam Sutra (UTS) of vedic mathematics meaning Vertical & Crosswise for fast multiplication to reduce critical delay in high performance processing units [17].

3.1 URDHVATIRYAGBHYAM SUTRA

Urdhva Tiryakbhyam Sutra (meaning vertically and crosswise) is a general multiplication formula applicable to all cases of multiplication. The parallel generation of sub-operand PPs and PP reduction based on the UTS for a 4 × 4 multiplier is shown in Fig.1 [18]. The multiplier uses four 2 × 2 multipliers to generate PPs which are then compressed to produce final product of a 4 × 4 multiplication.

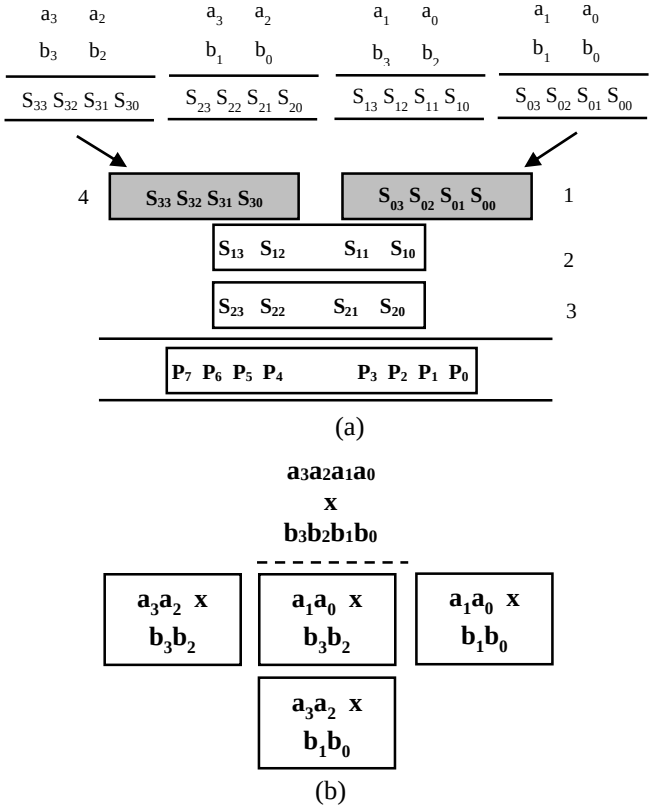

Fig.1. 4 × 4 Vedic multiplication (a). Block diagram, (b). PP Generation and Reduction

4. AREA AND DELAY EVALUATION OF CONVENTIONAL & VEDIC MULTIPLIERS

Area and delay evaluation of the vedic and conventional multiplier designs are done based on their equivalent NAND implementation. We assume the delay of NAND gate to be 1 unit and its area count to be 1. The gates which are in parallel perform operation simultaneously and we use one gate delay in these cases for calculation of delay of circuit/element. Based on the above approach, the delay and area of the individual elements that make up the proposed design are shown in Table.1.

The area count and worst case delay of the Vedic and conventional multiplier designs are calculated as follows. The worst case delay is found by counting the number of NAND the gates in the critical path and the area is evaluated by counting total number of NAND gates that make up the circuit and is shown in Table.2. From the evaluation results it is found that the Vedic multiplier has fewer area and delay by 5 and 1 NAND gates. So we used Vedic multiplier in the design of proposed arithmetic unit discussed in section 5.

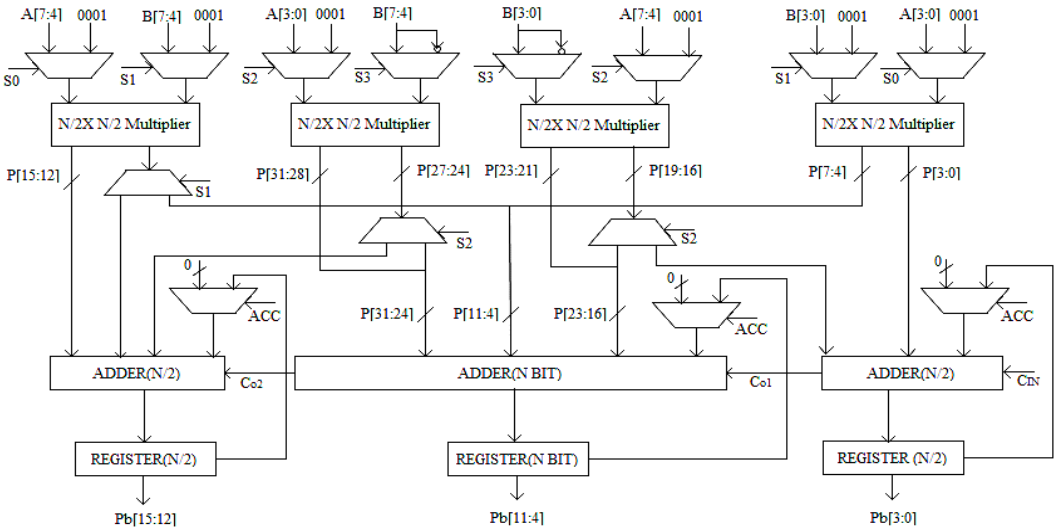

Fig.2. Proposed Arithmetic Unit

Table.1. Area and Worst Case Delay of Basic Building Blocks

Element	Area	Worst case delay
Full Adder	Sum:8 + Carry:4 =12	3
Half Adder	Sum:5 + Carry:4 =7	2
AND Gate	2	2

Table.2. Comparison of Area and Delay for Conventional and Vedic Multipliers

Basic Blocks	Conventional Multiplier		Vedic Multiplier	
	Area	Delay	Area	Delay
Full Adder	60	15	48	12
Half Adder	28	8	35	10
AND Gate	32	2	32	2
Total	120	25	115	24

5. PROPOSED ARITHMETIC UNIT

The proposed AU shown in Fig.2 accepts two *n*-bit operands *A* and *B*. The input operands are split into *A*[*N*-1:*N*/2] & *A*[*N*/2-1:0] and *B*[*N*-1:*N*/2] & *B*[*N*/2-1:0]. The input multiplexers directed by control inputs S_3 - S_0 sends the input sub-operands to the respective *n*/2 × *n*/2 multiplier units to produce intermediate products which are then compressed by the adder stages to produce final product. In addition the proposed AU performs addition, subtraction and accumulation operations on the inputs. Note that for accumulation operations the multiplier outputs are send back to adder stages.

5.1 ADDITION/SUBTRACTION

In case of addition operation, the multiplier on the left performs *A*[3:0] × 1, second one performs *B*[3:0] × 1, third multiplier performs *B*[7:4] × 1 while the rightmost multiplier perform *A*[7:4] × 1. The de-multiplexer controlled through S_2 directs the PPs to the next stage adder to compute the result of *A* + *B*. Note that for subtraction operation, the multiplier on the left performs *A*[3:0] × 1, while the right multiplier performs *A*[7:4] × 1 whereas the second and third multiplier performs the complement of *B*[3:0] × 1 and *B*[7:4] × 1. The de-multiplexer controlled through S_1 and S_2 directs the PPs to the next stage adder to compute the result of *A* - *B*. Also note that for subtract operations (see Fig.2) signal C_{IN} is set to 1, thus enabling 2s complement addition on inputs to realise subtract operation without additional hardware.

5.2 MULTIPLICATION & ACCUMULATION

Multiply-Accumulate operation is a common step that computes the product of two numbers and adds the product to accumulator [19]. To perform accumulation operation, the results of the addition are sent back to the data path through multiplexers controlled by signal ACC in the proposed AU.

6. RESULTS AND DISCUSSION

The proposed arithmetic unit was described using structural VHDL and synthesized using CADENCE Encounter RTL Compiler with TSMC 180nm technology file. The simulation output of the proposed arithmetic unit is shown in Fig.3. To validate different operations of AU the respective control inputs are triggered for a pair of inputs and is shown in Fig.3. We used Chalamalasetti et al. [8] and Purohit et al. [15] designs for comparison. The designs used for comparison are also designed using structural VHDL and synthesized using CADENCE Encounter RTL Compiler with the same technology file, and the synthesis results including hardware area (Area), worst case delay (Delay) and power dissipation (Total power) are shown in Table.3.

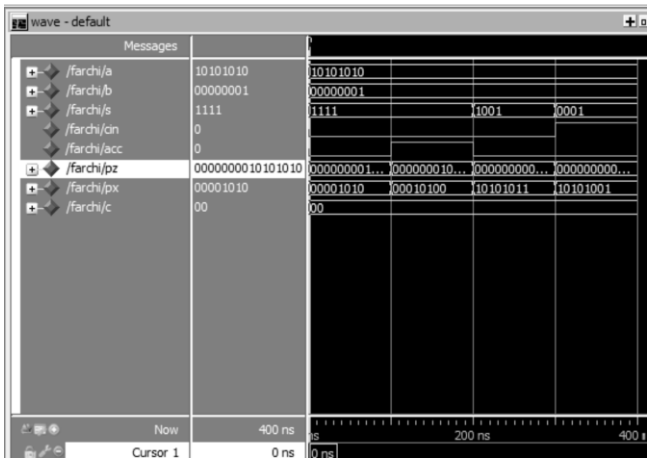

Fig.3. Simulation output of proposed AU

Table.3. Comparison of Area, Power, Delay of Proposed AU and Previous Approaches

Arithmetic Unit	Area (μm^2)	Delay (ns)	Power (µW)	ADP	PDP (Joules)
Ref[8]	1061.12	5.293	17.615	5616.5	93.23
Ref[15]	930.924	4.812	16.161	4479.6	77.67
Proposed	870.390	4.154	19.182	3615.6	79.67

From the synthesis results, it is seen that the proposed arithmetic unit demonstrates delay reduction of 21.5% and 13.7% compared to Chalamalasetti et al. [8] and Purohit et al. [15] arithmetic units respectively. This is due to parallel PP generation and reduced critical path of the Vedic multiplier used in the proposed AU. Note that the proposed arithmetic unit demonstrates an area reduction of 17.9% and 6.5% compared to Chalamalasetti et al. [8] and Purohit et al. [15] designs respectively, thanks to Vedic algorithm which reduces number of gates and number of full adders required for PP generation and compression respectively.

Also note that the better delay and area reductions of the proposed AU realizes better ADP saving, with the ADP of our proposed design reducing by 35.6% and 19.2% compared to the designs in [8] and [15] respectively. In addition we have extracted the physical layout of the proposed AU using CADENCE Encounter digital implementation and is shown in Fig.4.

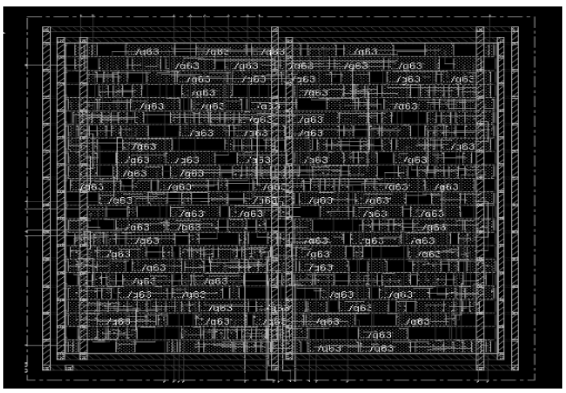

Fig.4. Physical Layout of the Proposed AU

7. CONCLUSION

The proposed work developed an approach for design of high speed and area efficient AU suitable for high performance processing architectures. Extensive comparisons using synthesis results shows that the proposed AU out performed previous designs in terms of critical delay and area. This suggests the suitability of the proposed AU for high speed portable VLSI implementation. Note that the structuring of our proposed AU with equal logic depth maintains parallelism and uniform delay across all the outputs. In addition the design of the blocks and multiplexers are such that the design can be reconfigured for higher bitwidths of input operand with ease.

REFERENCES

[1] M. Ramalatha, K. D. Dayalan, P. Dharani and S.D. Priya, "High Speed Energy Efficient ALU Design using Vedic Multiplication Techniques", *Proceedings of International Conference on Advances in Computational Tools for Engineering Applications*, pp. 600-603, 2009.

[2] Swapnil Manohar Mehkarkar and Snehal J. Banarase, "Implementation of High Speed FIR Filter based on Ancient Vedic Multiplication Technique", *International Journal of Emerging Technology and Advanced Engineering*, Vol. 4, No. 5, pp. 763-770, 2014.

[3] Rong Lin, "Reconfigurable Parallel Inner Product Processor Architectures", *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 9, No. 2, pp. 261-272, 2001.

[4] Wen-Chang Yeh and Chein-Wei Jen, "High-Speed Booth Encoded Parallel Multiplier Design", *IEEE Transactions on Computers*, Vol. 49, No. 7, pp. 692-701, 2000.

[5] Lan-Da Van and Chih-Chyau Yang, "Generalized Low-Error Area-Efficient Fixed-Width Multipliers", *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 52, No. 8, pp. 1608-1619, 2005.

[6] Higinio Mora-Mora, Jerónimo Mora-Pascual, José Luis Sánchez-Romero and Juan Manuel García-Chamizo, "Partial Product Reduction by using Look-Up Tables for M×N Multiplier", *Integration, The VLSI Journal*, Vol. 41, No. 4, pp. 557-571, 2008.

[7] Jin-Hao Tu and Lan-Da Van, "Power-Efficient Pipelined Reconfigurable Fixed-Width Baugh-Wooley Multipliers", *IEEE Transactions on Computers*, Vol. 58, No. 10, pp. 1346-1355, 2009.

[8] S.R. Chalamalasetti, S.S. Purohit, M. Margala and W. Vanderbauwhede, "MORA - An Architecture and Programming Model for a Resource Efficient Coarse Grained Reconfigurable Processor", *Proceedings of NASA/ESA Conference on Adaptive Hardware and Systems*, pp. 389-396, 2009.

[9] Sotiris Xydis, George Economakos and Kiamal Pekmestzi, "Designing Coarse-Grain Reconfigurable Architectures by Inlining Flexibility into Custom Arithmetic Datapaths", *Integration, The VLSI Journal*, Vol. 42, No. 4, pp. 486-503, 2009.

[10] S. Shanthala and S.Y. Kulkarni, "VLSI Design and Implementation of Low Power MAC Unit with Block Enabling Technique", *European Journal of Scientific Research*, Vol. 30, No. 4, pp. 620-630, 2009.

[11] Khader Mohammad, Sos Agaian and Fred Hudson, "Implementation of Digital Electronic Arithmetics and its Application in Image Processing", *Computers and Electrical Engineering*, Vol. 36, No. 3, pp. 424-434, 2010.

[12] Manoranjan Pradhan, Rutuparna Panda and Sushanta Kumar Sahu, "MAC Implementation using Vedic Multiplication Algorithm", *International Journal of Computer Applications*, Vol. 21, No. 7, pp. 26-28, 2011.

[13] Harish Kumar and A.R. Hemanth Kumar, "Design and Implementation of Vedic Multiplier using Compressors", *International Journal of Engineering Research & Technology*, Vol. 4, No. 6, pp. 230-233, 2015.

[14] Virendra B. Magar, "Intelligent and Superior Vedic Multiplier for FPGA Based Arithmetic Circuits", *International Journal of Soft Computing and Engineering*, Vol. 3, No. 3, pp. 31-36, 2013.

[15] S.S. Purohit, S.R. Chalamalasetti, M. Margala and W.A. Vanderbauwhede, "Design and Evaluation of High-Performance Processing Elements for Reconfigurable Systems", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 21, No. 10, pp. 1915-1927, 2013.

[16] http://www.vedamu.org/Vedic Mathematics.pdf

[17] S. Leonard Gibson Moses and M. Thilagar, "VLSI Implementation of High Speed DSP algorithms using Vedic Mathematics", *Singaporean Journal Scientific Research*, Vol. 3, No. 1, pp. 138-140, 2010.

[18] Nishant G. Deshpande and Rashmi Mahajan, "Ancient Indian Vedic Mathematics based 32-Bit Multiplier Design for High Speed and Low Power Processors", *International Journal of Computer Applications*, Vol. 95, No. 24, pp.19-22, 2014.

[19] Irshad Khan, Sunil Shah and Vinod Kapse, "Quick Review on Multiplication Algorithm for Enhancing Efficiency of MAC Unit", *Proceedings of International Conference of Advance Research and Innovation*, pp. 159-162, 2015.