Copyright © 2019 American Scientific Publishers
All rights reserved
Printed in the United States of America

Journal of
Low Power Electronics
Vol. 15, 302–308, 2019

A High-Performance Signed-Unsigned Multiplier Using Vedic Mathematics

Ganesh Kumar Ganjikunta*, Sibghatullah I. Khan, and M. Mahaboob Basha

Dept. of ECE, Sreenidhi Institute of Science & Technology, Hyderabad 501301, India

(Received: 30 March 2019; Accepted: 8 July 2019)

A high speed $N \times N$ bit multiplier architecture that supports signed and unsigned multiplication operations is proposed in this paper. This architecture incorporates the modified two's complement circuits and also $N \times N$ bit unsigned multiplier circuit. This unsigned multiplier circuit is based on decomposing the multiplier circuit into smaller-precision independent multipliers using Vedic Mathematics. These individual multipliers generate the partial products in parallel for high speed operation, which are combined by using high speed adders and parallel adder to generate the product output. The proposed architecture has regular-shape for the partial product tree that makes easy to implement. Finally, this multiplier architecture is implemented in UMC 65 nm technology for $N = 8$, 16 and 32 bits. The synthesis results shows that the proposed multiplier architecture improves in terms of speed and also reduces power-delay product (PDP), compared to the architectures in the literature.

Keywords: Vedic Mathematics, Signed-Unsigned Multiplier, Power-Delay Product, Vedic Multiplier.

IP: 5.10.31.210 On: Wed, 18 Dec 2019 09:25:34
Copyright: American Scientific Publishers
Delivered by Ingenta

1. INTRODUCTION

Multiplication is one of the basic arithmetic operation utilized in most of the signal processing applications. As per Oberman,[1] most of the instructions are multiplications in a scientific program. Therefore, it is greatly imperative to develop high speed and energy efficient multipliers for digital signal processing (DSP) and multimedia applications.[2]

Many high performance multiplier algorithms and architectures have been designed and explored by the researchers in the past few years. Among them, array multipliers[3] are often used which are based on shift and add method with regular structure. Further, so as to decrease the delay of multipliers, the partial products of the multiplier are summed up by a Wallace tree method.[4–6] However, the layout of Wallace tree method is irregular.

In Refs. [7] and [8], new multiplier designs are proposed based on the Vedic Mathematics approach. In these, a divide and conquer technique is used to obtain the final results. To further improve the performance of these designs, the partial products of the multiplier are summed up by carry save adder and carry-lookahead adder.[9–13] In Ref. [14], the adders are replaced with compressors and named the design as compressor based Vedic multiplier (CVM).

The major advancements in multipliers design is reducing the number of partial products in the Booth's and modified Booth's algorithms. So far, certain DSP applications require multipliers that can support either signed or unsigned binary numbers. However, there is a need for design that can support both signed and unsigned operations. In Ref. [15], a unified signed-unsigned architecture based on tree structure is proposed for redundant binary arithmetic. Further in Ref. [16], a signed-unsigned Modified Booth Encoding (SUMBE) multiplier is proposed for multiplier. However, the main requirements for high performance DSP systems are the speed and energy efficiency.

In this paper, we propose a high speed $N \times N$ bit multiplier architecture that can support signed-unsigned operation. A modified two's complement converter circuit is used to convert the negative binary input of either multiplicand or multiplier into positive. These positive inputs are given to a high-speed unsigned multiplier. Finally at the last stage, one more two's complement converter circuit is employed to the final product for negative number inputs.

The remainder of this paper is organized as follows. Section 2 presents the $N \times N$ bit unsigned multiplier using Urdhva–Tiryagbhyam principle. Section 3 describes the proposed $N \times N$ bit signed-unsigned multiplier architecture that comprises of a modified two's complement converter circuit and an $N \times N$ bit unsigned Vedic multiplier. In Section 4, for $N = 8$, 16 and 32 bits, the implementation

*Author to whom correspondence should be addressed.
Email: ganeshkumarg@sreenidhi.edu.in

1546-1998/2019/15/302/007 doi:10.1166/jolpe.2019.1616

and evaluation of the proposed unsigned multiplier and signed-unsigned architecture with other multiplier designs is presented. Finally, Section 5 presents the concluding remark of this work.

2. VEDIC MULTIPLIER USING URDHVA-TIRYAKBHYAM

The name Vedic Mathematics is inferred from the ancient system of Indian Mathematics. This is a mathematical elaboration of formulae from the Vedas that contain sixteen sutras. These Mathematical formulae deals with various derivatives of mathematics like arithmetic, analytical geometry, algebra, calculus, trigonometry etc. Among these sixteen sutras, Urdhva–tiryagbhyam (vertically and cross-wise) is one of the important formula that can be useful for multiplication. Therefore, this sutra is considered for the implementation of proposed signed-unsigned multiplier.

In the following subsections, at first we discuss the basic building block of multiplier which is a 2×2 bit multiplier. Then, we discuss the design of a 4×4 binary multiplier using Urdhva–tiryagbhyam principle with 2×2 bit multiplier. This method of multiplication is extended to an 8×8 bit multiplier. To improve the speed of the design, a carry select adder is employed at the final stage to add the partial products of the multiplier. Finally, we discuss the extension of this method in order to obtain an $N \times N$ bit unsigned multiplier.

IP: 5.10.31.210 On: Wed, 18 Dec 2019 19:25:34
Copyright: American Scientific Publishers
Delivered by Ingenta

2.1. 2 × 2 Bit Multiplier

The multiplication algorithm is illustrated by considering two binary numbers each of 2-bits as A_1A_0 and B_1B_0. The output of this multiplication product is represented as $Z_3Z_2Z_1Z_0$. The multiplication result is obtained by generating partial products using AND gates and adding these partial products using half adders. A 2×2 binary multiplier is shown in Figure 1. This can be used to build the higher bit multipliers.

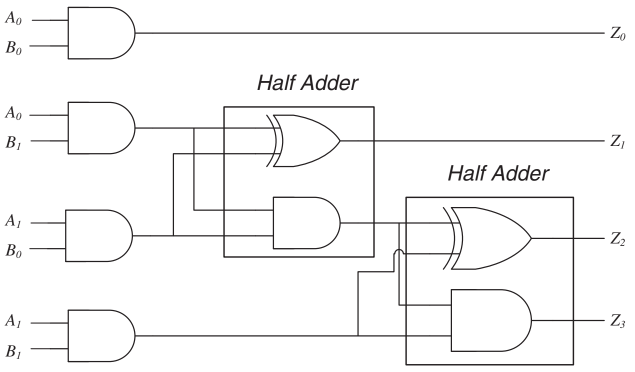

Fig. 1. 2 × 2 binary multiplier.

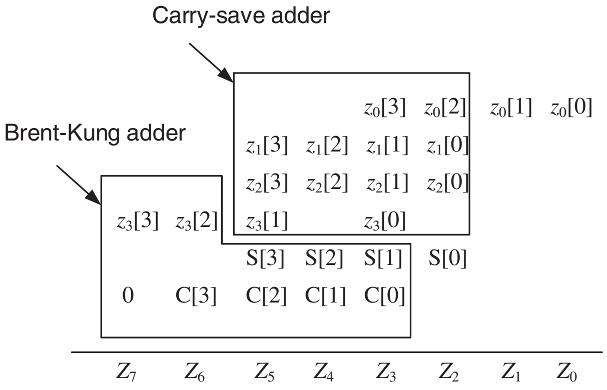

Fig. 2. Addition of partial products using Carry-save and Brent-Kung adders.

2.2. 4 × 4 Bit Multiplier

A multiplier with multiplicand and multiplier as A and B of operand size of $N = 4$ bits is considered. These operands are divided equally into two parts of $n = N/2$ bits each.

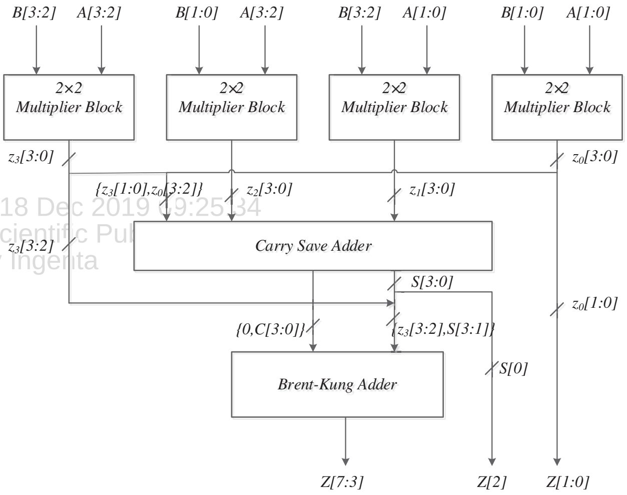

Fig. 3. Block diagram of 4 × 4 binary multiplier.

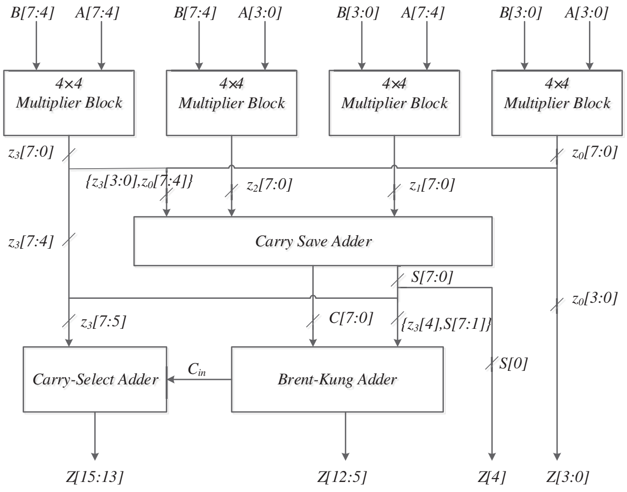

Fig. 4. Block diagram of 8 × 8 bit binary multiplier.

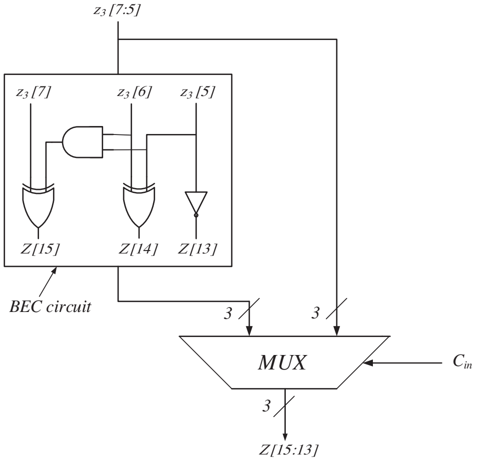

Fig. 5. Carry-select adder with BEC circuit.

Here, $A_H = A_3A_2$, $B_H = B_3B_2$ are their respective two most significant bits (MSBs), whereas $A_L = A_1A_0$, $B_L = B_1B_0$ are their respective two least significant bits (LSBs).

According to Urdhva–tiryagbhyam, the multiplication of A and B can be computed as follows:

$$Z = A \cdot B = 2^{2n}(A_H B_H) + 2^n(A_H B_L + A_L B_H) + (A_L B_L) \tag{1}$$

At first, vertical multiplication of A_L and B_L is executed that produces the partial products as $z_0[3{:}0]$. Next, crosswise multiplication of two 2×2 bit multipliers is performed with input A_H and B_L and A_L and B_H. This generates the partial products as $z_1[3{:}0]$ and $z_2[3{:}0]$. Finally, vertical multiplication of A_H and B_H generates $z_3[3{:}0]$ partial products. The summation of these generated partial products is considered as shown in Figure 2.

The 4×4 bit multiplier using four 2×2 multiplier blocks, fast adder and parallel adder which can reduces the delay is shown in Figure 3. The generated partial product bits $z_0[0]$ and $z_0[1]$ are considered as the final product bits $Z[1{:}0]$. A carry save adder is employed to add the partial products $z_1[3{:}0]$, $z_2[3{:}0]$ and $\{z_3[1:0], z_0[3:2]\}$. This generates the sum outputs as $S[3{:}0]$ and carry outputs as $C[3{:}0]$. The sum output $S[0]$ is considered as the final product bit $Z[2]$. Finally, the sum output bits $S[3{:}1]$, carry bits $C[3{:}0]$ and the partial product bits $z_3[3{:}2]$, are summed up by using Brent-Kung adder to obtain the final product terms $Z[7{:}3]$ of the 4×4 bit multiplier.

2.3. 8 × 8 Bit Multiplier

In the similar way of 4×4 bit multiplication, the multiplicand and multiplier bits of 8×8 bit multiplier are decomposed as $A = A_H A_L$ and $B = B_H B_L$. The higher and lower bits of A and B are considered as $A_H = A_7A_6A_5A_4$, $A_L = A_3A_2A_1A_0$, $B_H = B_7B_6B_5B_4$ and $B_L = B_3B_2B_1B_0$. These decomposed inputs are given to four 4×4 multiplier blocks that generates partial products in parallel and it is shown in Figure 4. The four LSB partial products output ($z_0[3{:}0]$) of the first bit 4×4 multiplier block is directly given as output to final product bits $Z[3{:}0]$. A carry save adder is employed to sum the partial product bits $z_1[7{:}0]$, $z_2[7{:}0]$ and $\{z_3[3:0], z_0[7:4]\}$. This generates the sum output as $S[7{:}0]$ and carry output as $C[7{:}0]$. The sum output $S[0]$ is given directly to the final product output $Z[4]$ bit. Next, the carry save adder outputs sum and carry bits

IP: 5.10.31.210 On: Wed, 18 Dec 2019 09:25:34
Copyright: American Scientific Publishers
Delivered by Ingenta

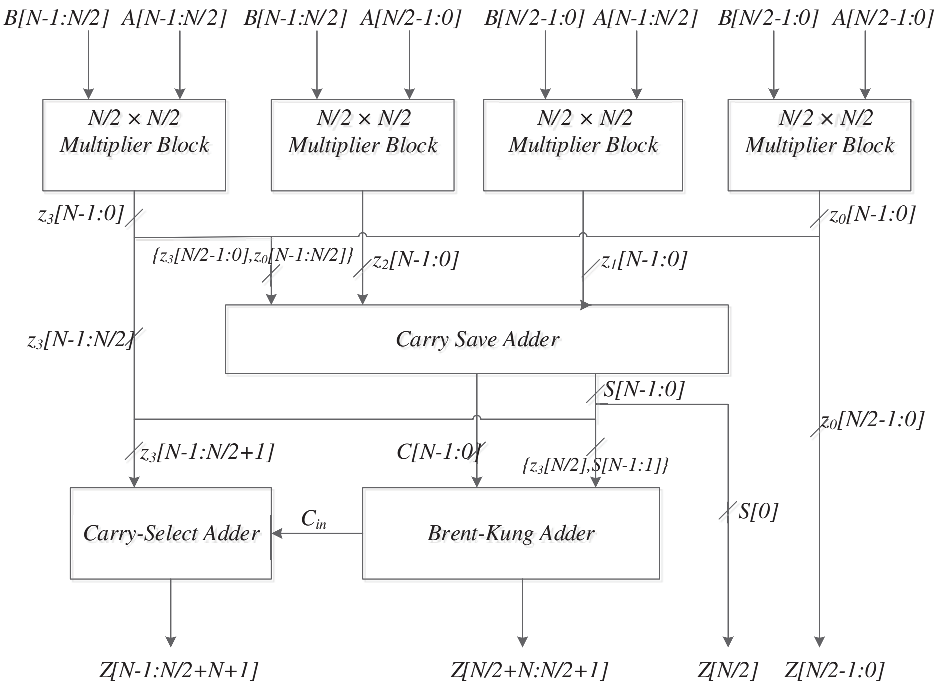

Fig. 6. Block diagram of $N \times N$ bit unsigned multiplier.

are summed up using Brent-Kung adder additionally with the partial product bit $z_3[4]$. This can generates the final output terms $Z[12{:}5]$ and a carry bit C_{in}. The C_{in} bit and the $z_3[7{:}5]$ are given to a high speed carry select adder that gives the final product outputs $Z[15{:}13]$. The internal structure of the high speed carry select adder is discussed below.

The carry select adder that includes a Binary to Excess-1 Converter (BEC) circuit and a multiplexer (MUX) is shown in Figure 5. The BEC circuit adds one to the input $z_3[7{:}5]$ and the output is feeds as one of the input to the multiplexer. The other input to the multiplexer is $z_3[7{:}5]$. If the carry bit $C_{\text{in}} = 0$, then $z_3[7{:}5]$ becomes the final product terms $Z[15{:}13]$. If the input carry bit $C_{\text{in}} = 1$ then, the output of the multiplexer is the addition of the partial product output $p_3[7{:}5]$ with the BEC circuit that gives the final product $Z[15{:}13]$.

2.4. $N \times N$ Bit Multiplier

The procedure discussed in the earlier subsection can be generalized to multiplication of two binary numbers of arbitrary sizes. To multiply $N \times N$ bit numbers, initially the operands A and B are decomposed equally into two $N/2$-bits. Then the decomposed operands are further decomposed until the bit length is equal to 2. At first 2×2 bit multiplier is designed and then 4×4 bit multiplier is realized using four parallel 2×2 bit multiplier blocks and fast adders. This process is extended to $N \times N$ bit multiplier using four parallel $N/2 \times N/2$ bit multipliers along with fast adders (carry save and carry select adders) and parallel adder (Brent-Kung adder) as shown in Figure 6.

3. $N \times N$ SIGNED-UNSIGNED MULTIPLIER ARCHITECTURE

The design of unsigned multiplier using Vedic approach is discussed in Section 2. In some of the DSP systems require both signed and unsigned multipliers. Therefore, in this section a signed multiplier using the proposed unsigned multiplier structure is discussed. This multiplier supports various modes of operation, namely signed-signed, unsigned-unsigned, unsigned-signed or signed-unsigned multiplication. In the case of unsigned multiplication, all N-bits are represented as the magnitude of the operand. In the case of signed multiplication, the MSB bits of operand represent the sign of the input and remaining, $N-1$ bits are represented as the magnitude of the operand.

The proposed $N \times N$ bit signed-unsigned multiplier architecture is shown in Figure 7. Both the N-bit input operands are passes through two N-bit two's complement converter logic that converts input into its two's complement form. For a negative number, the converter gives the positive magnitude of the operand. The actual N-bit input and the two's complemented N-bit inputs are fed to a 2:1 multiplexer array. The multiplexer selects one of them using the control unit. This unit contain an AND gate with

IP: 3.10.31.210 On: Wed, 18 Dec 2019 09:23:34
Copyright: American Scientific Publishers
Delivered by Ingenta

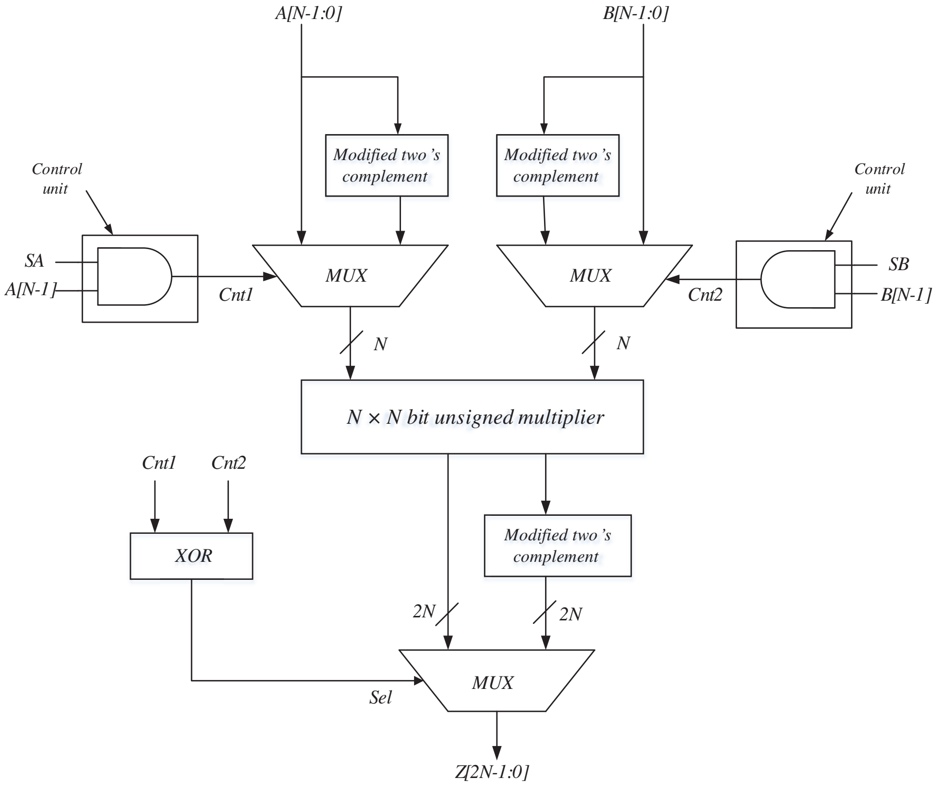
A[N-1:0]
B[N-1:0]
Modified two 's complement
Modified two 's complement
Control unit
Control unit
SA
A[N-1]
Cnt1
MUX
MUX
Cnt2
SB
B[N-1]
N
N
N × N bit unsigned multiplier
Cnt1
Cnt2
XOR
Modified two 's complement
2N
2N
Sel
MUX
Z[2N-1:0]

Fig. 7. Block diagram of the proposed $N \times N$ bit signed-unsigned multiplier.

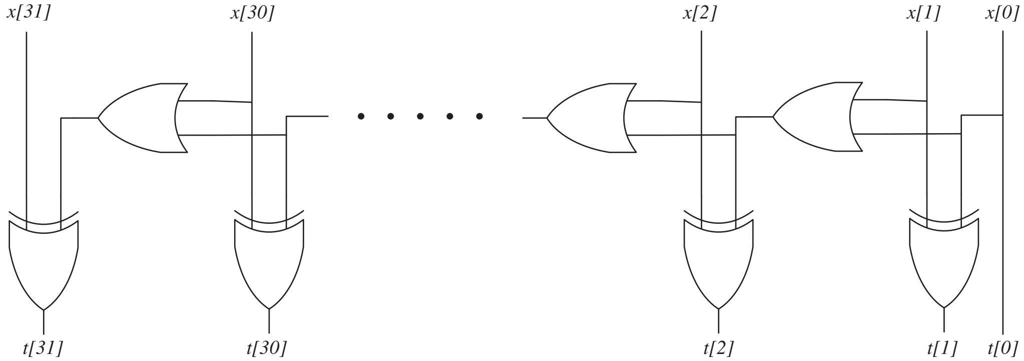

Fig. 8. Modified two's complement circuit for 32-bit input.

the input sign control signals SA, SB and MSB bits of the operands. This control unit produces the output signals *Cnt*1 and *Cnt*2 as shown in Figure 7.

The selection of the multiplication operation on signed or unsigned operands is depends on the control unit. If the control signals SA and SB are 0, then the architecture performs unsigned multiplication. If the control signal SA and SB are 1, it is considered as signed multiplication. If the control signals SA = 0 and SB = 1, then it performs unsigned-signed multiplication. If the control signals SA = 1 and SB = 0, then it performs signed-unsigned multiplication. The MUX select the original value, if the input binary value is positive number. If the input binary value is negative number, the MUX select the output from the two's complement converter logic, which is a positive value with the same magnitude. At this stage both the multiplier and multiplicand input is converted into positive and these are given as input to the $N \times N$ bit unsigned multiplier. This unsigned multiplier is employed as shown in Figure 6.

IP: 3.10.31.210 On: Wed, 18 Dec 2019 09:25:34
Copyright: American Scientific Publishers
Delivered by Ingenta

The $N \times N$ bit unsigned multiplier output is given to another 2*N*-bit two's complement converter circuit. The unsigned multiplier output as well as the two's complemented unsigned multiplier output are fed to a final 2:1 multiplexer. The select bit to this final MUX is exclusive-or (XOR) of the control unit output signals Cnt1 and Cnt2. If both of the original inputs are either positive or negative, then the final multiplier output is positive which is taken directly from unsigned multiplier output, and if one of the input is positive while the other is negative, then the output is negative which is taken from two's complement output. Therefore, the final MUX can appropriately select which 2*N*-bit value to the output and then this output is the final output of the signed-unsigned multiplier.

The conventional method of two's complement takes the complement of a binary number and summed up by 1 to the complemented number. However, it requires large area and also propagation delay is more. Therefore, a modified two's complement converter circuit is designed using XOR and OR gates in the proposed architecture. In order to explain this, a modified two's complement converter circuit with 32-bit is considered. This circuit takes the inputs $x[31{:}0]$ and produces the two's complement outputs $t[31{:}0]$ using XOR and OR gates as shown in Figure 8.

The usage of the fast adders, parallel adder and modified two's complement converter circuit improves the performance of the proposed architecture. Therefore, the proposed signed-unsigned multiplier can be useful for faster signal processing and multimedia applications. However, the proposed multiplier requires large area for computation.

4. EXPERIMENTAL RESULTS

The unsigned multiplier and the signed-unsigned multiplier discussed in the last two sections are implemented for operand size of 8, 16 and 32. The RTL level Verilog HDL code is written for the proposed multipliers of different operand size. The synthesis of the designs is done by using Synopsys Design Compiler with the UMC 65 nm CMOS library. Further, the Cadence Encounter Digital Implementation System is employed for placement and routing. The delays are estimated after extracting

Table I. Comparison of various unsigned multiplier.

N	Multiplier	Delay (ns)	$D(-)$ (%)	Area (um^2)	Power (mW)	Power-delay product (pJ)	PDP(−) (%)
8	Jaina[11]	1.39	–	723.83	0.350	0.486	–
	Noll[4]	1.31	5.8	576.63	0.346	0.453	6.8
	Huddar[14]	1.25	10.0	728.92	0.352	0.440	9.4
	Fadavi[17]	1.17	15.8	581.11	0.348	0.407	16.2
	Proposed	1.04	25.1	749.12	0.358	0.372	23.4
16	Jaina[11]	2.87	–	3007.35	1.420	4.075	–
	Noll[4]	2.68	6.6	1776.95	1.297	3.475	14.7
	Huddar[14]	1.82	36.5	3104.54	1.832	3.334	18.1
	Fadavi[17]	1.80	37.3	2003.11	1.821	3.227	20.8
	Proposed	1.57	45.3	3331.52	1.948	3.058	24.9
32	Jaina[11]	4.68	–	10428.32	7.113	33.288	–
	Noll[4]	3.92	16.2	7764.24	7.019	27.154	18.4
	Huddar[14]	2.93	37.3	14242.12	8.720	25.549	23.2
	Fadavi[17]	2.83	39.5	8948.48	8.631	24.425	26.6
	Proposed	2.52	46.1	14431.52	9.054	22.816	31.4

Table II. Experimental results of signed-unsigned multipliers.

N	Multiplier	Delay (ns)	$D(-)$ (%)	Area (um^2)	Power (mW)	Power-delay Product (pJ)	PDP(−) (%)
8	Rajput[16]	1.64	–	843.84	0.44	0.721	–
	Proposed	1.32	19.5	946.88	0.48	0.633	12.2
16	Rajput[16]	2.68	–	2642.52	2.53	6.780	–
	Proposed	2.22	17.1	3870.08	2.71	6.016	11.2
32	Rajput[16]	4.72	–	10421.72	12.43	58.764	–
	Proposed	4.08	13.5	15121.08	13.10	53.474	9.0

the RC from the placed and routed netlists. The average power-consumption of the designs is estimated by adopting the Synopsys Primetime.

For comparison, four other recent related and important multipliers proposed by Noll,[4] Jaina,[11] Huddar[14] and Fadavi[17] are also implemented in the same platform of the proposed implementation. The delay, area, power consumption and power-delay product of the proposed unsigned multiplier along with the multipliers proposed by others are summarized in Table I. Moreover, $D(-)$, and $PDP(-)$ in Table I denotes the delay and PDP decrements when compared with the Jaina.[11] The multiplier by Jaina[11] with maximum delay is placed in the first row. Then all other multiplier results are listed in the Table I with decreasing delay. The delay comparison in Table I shows that the proposed unsigned multiplier has minimum delay in comparison to all other multipliers. This proposed multiplier has delay advantage of almost 25%, 45% and 46% for 8, 16 and 32 bit multipliers respectively, in comparison to Jaina.[11] As the proposed circuit uses parallel multiplier blocks and fast adders, the hardware requirement and power consumption of the proposed circuit is more when compare to other designs. But, in terms of PDP the proposed architecture gives advantage of almost 23%, 25% and 31% for 8, 16 and 32 bit multiplier respectively in comparison to Jaina.[11]

Furthermore, we have also implemented and synthesized the recent implementation of signed-unsigned multiplier by Rajput[16] and the proposed signed-unsigned multiplier for $N = 8$, 16 and 32 bits. In the proposed signed-unsigned multiplier, a modified two's complement converter circuit is employed along with the unsigned multiplier[18] to improve the speed of the design. The implementation results are shown in Table II. From this Table II, one can be observed that the proposed signed-unsigned multiplier for 32×32 bit offer 13.5% and 9% decrement over the Rajput[16] multiplier in terms of delay and PDP, respectively. From these results, one can say that the proposed design performance efficiently in terms of speed and power-delay product.

5. CONCLUSION

In this paper, we have proposed an $N \times N$ bit high speed signed-unsigned multiplier architecture using Vedic Mathematics. This proposed signed-unsigned multiplier comprises modified two's complement circuits and an $N \times N$ bit unsigned multiplier circuit. The unsigned multiplier generates the partial products in parallel, which makes the structure more regular, easy to implement and results in high speed partial product generation. The speed of the design is further enhanced by using fast adders and parallel adder, to accumulate the partial products. From the synthesis results, the proposed unsigned multiplier shows up to 46% delay and 31% PDP advantage over one of the previous reported multiplier for operand size of 32-bit. The signed multiplier using the proposed unsigned multiplier also gives remarkable delay and PDP advantage. Therefore, the proposed multiplier is suitable for high speed processing systems.

References

1. S. F. Oberman and M. J. Flynn, Design issues in division and other floating-point operations. *IEEE Transactions on Computers* 46, 154 (**1997**).
2. J. H. McClellan, R. W. Shafer, and M. A. Yoder, DSP First: A Multimedia Approach. Prentice Hall PTR (**1997**).
3. H. Guilt, Fully iterative fast array for binary multiplication. *Electronics Letters* 5, 263 (**1969**).
4. T. G. Noll, Carry-save architectures for high-speed digital signal processing. *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology* 3, 121 (**1991**).
5. G. Goto, T. Sato, M. Nakajima, and T. Sukemura, A 54×54-b regularly structured tree multiplier. *IEEE Journal of Solid-State Circuits* 27, 1229 (**1992**).
6. G. Goto, High speed digital parallel multiplier, US Patent 5465226 (**1995**).
7. H. Thapliyal, M. Srinivas, and H. R. Arabnia, Design and analysis of a VLSI based high performance low power parallel square architecture. AMCS (**2005**), pp. 72–76.
8. S. Akhter, VHDL implementation of fast $N \times N$ multiplier based on Vedic mathematic, *Proceedings of the IEEE European Conference on Circuit Theory and Design* (**2007**), pp. 472–475.
9. H. D. Tiwari, G. Gankhuyag, C. M. Kim, and Y. B. Cho, Multiplier design based on ancient Indian Vedic mathematics, *Proceedings of the International SoC Design Conference* (**2008**), Vol. 2, pp. 65–68.
10. M. Ramalatha, K. D. Dayalan, P. Dharani, and S. D. Priya, High speed energy efficient ALU design using Vedic multiplication techniques, *Proceedings of the International Conference on Advances in Computational Tools for Engineering Applications* (**2009**), pp. 600–603.
11. D. Jaina, K. Sethi, and R. Panda, Vedic mathematics based multiply accumulate unit, *Proceedings of the IEEE International Conference on Computational Intelligence and Communication Networks* (**2011**), pp. 754–757.
12. V. Kunchigi, L. Kulkarni, and S. Kulkarni, High speed and area efficient Vedic multiplier, *Proceedings of the International Conference on Devices, Circuits and Systems* (**2012**), pp. 360–364.
13. Y. Bansal, C. Madhu, and P. Kaur, High speed vedic multiplier designs—A review, *Proceedings of the IEEE Recent Advances in Engineering and Computational Sciences* (**2014**), pp. 1–6.
14. S. R. Huddar, S. R. Rupanagudi, M. Kalpana, and S. Mohan, Novel high speed vedic mathematics multiplier using compressors, *Proceedings of the IEEE International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing* (**2013**), pp. 465–469.

IP: 5.10.31.210 On: Wed, 18 Dec 2019 09:25:34
Copyright: American Scientific Publishers
Delivered by Ingenta

15. G. Wang, A unified unsigned/signed binary multiplier, *Asilomar Conference on Signals, Systems & Computers* (**2004**).

16. R. P. Rajput and M. S. Swamy, High speed modified booth encoder multiplier for signed and unsigned numbers, *Proceedings of the IEEE International Conference on Computer Modelling and Simulation* (**2012**), pp. 649–654.

17. J. Fadavi-Ardekani, $M*N$ booth encoded multiplier generator using optimized wallace trees. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 1, 120 (**1993**).

18. G. G. Kumar and S. K. Sahoo, Implementation of a high speed multiplier for high-performance and low power applications, *Proceedings of the IEEE International Symposium on VLSI Design and Test* (**2015**), pp.1–4.

Ganesh Kumar Ganjikunta

Ganesh Kumar Ganjikunta *completed his B.Tech and M.Tech in Electronics and Communication Engineering from Jawaharlal Nehru Technological University (JNTU) - Anantapur, Anantapur, India in the year 2009 and 2011, respectively. He received the Ph.D. degree in Electrical Engineering from BITS-Pilani, Hyderabad Campus, in 2019. Presently he is working as Assistant Professor in the Department of Electronics and Communication Engineering at SNIST, INDIA. His research interests focus on VLSI arithmetic circuits and Digital Signal Processing.*

Sibghatullah Khan

Sibghatullah Khan *completed his B.E, M.E and PhD in Electronics and Telecommunication Engineering from SGB Amravati universriy, India in the year 2007, 2012, and 2018 respectively. Presently he is working as Assosciate Professor in the Department of Electronics and Communication Engineering at SNIST, INDIA. His research interests focus on Bio-Medical Signal Processing.*

IP: 5.10.31.210 On: Wed, 18 Dec 2019 09:25:34
Copyright: American Scientific Publishers
Delivered by Ingenta

M. Mahaboob Basha

M. Mahaboob Basha *has obtained B.Tech degree in Electronics and Communication Engineering from J.N.T.U University, Hyderabad, INDIA in the year 2001 and M.Tech degree in Communication and signal processing from S.K. University, Anantapuramu, INDIA in the year 2006. Presently working as Associate professor in the Department of Electronics and Communication Engineering at SNIST, INDIA. Published 15 journals in National and International level. Professional Experience in teaching field is 15 Years. He is currently working with Ph.D in the area of "Low Power VLSI Design" from JNTUA university, Anantapuramu, INDIA. Interesting area of research includes Signal processing, high performance VLSI circuits and Image processing.*