JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES
www.journalimcms.org

ISSN (Online) : 2454 -7190 Vol.-19, No.-7, July (2024) pp 59-76 ISSN (Print) 0973-8975

A COMPREHENSIVE REVIEW ON LOW POWER FIXED WIDTH DIGITAL MULTIPLIER ARCHITECTURES

Biswarup Mukherjee

Department of Computer Science and Engineering, Brainware University, West Bengal, India.

Email : biswarup@ieee.org

https://doi.org/10.26782/jmcms.2024.07.00005

(Received: March 27, 2024; Revised: May 07, 2024; Accepted: June 15, 2024)

Abstract

In contemporary portable electronic devices featuring real-time DSP chips, a pivotal challenge lies in minimizing power consumption. The efficiency of these DSP chips is directly impacted by the substantial power dissipation of their multiplier sub-circuits. Consequently, numerous architectures emphasizing low-power consumption, high-speed operation, and compact layout structures for multiplier units have emerged in the literature over recent decades. This manuscript offers insights into select state-of-the-art fixed-width multiplier architectures tailored for low-power operation, presenting a detailed comparative analysis in terms of power consumption, area utilization, and processing delay. Notable among the fixed-width multiplier architectures are the serial, array, Vedic, Booth, Wallace-tree, and Modified Booth-Wallace designs. For operations involving larger operands, the Modified Booth-Wallace architecture is favored due to its reduced latency. This study concentrates on a comprehensive examination and evaluation of various low-power fixed-width multiplier architectures, highlighting diverse operand sizes. Simulation-based assessments utilizing the 45nm PTM model indicate that the Modified Booth-Wallace tree architecture achieves a 73% reduction in latency compared to a basic array multiplier. Moreover, CMOS-based designs demonstrate superior noise margin performance compared to GDI and CCGDI techniques. Notably, the dynamic voltage-controlled CCGDI-based architecture showcases a 60% enhancement in Power-Delay Product (PDP) compared to the conventional CMOS-based Modified Booth-Wallace multiplier architecture. The manuscript's novelty lies in its succinct overview of the latest multiplier architectures implemented at the 45nm technology node, specifically tailored for low-power DSP chips.

Keywords: Booth Algorithm, GDI, Low power VLSI, Multiplier, Wallace Tree architecture

I. Introduction

Modern DSP chips are intricate systems comprising numerous Intellectual Properties (IPs), with the multiplier section garnering particular attention owing to its significance in engineering applications. Although floating-point multiplication algorithms are prevalent in modern DSP, they are suboptimal for ASIC

implementation. Floating-point multiplier circuits entail greater logic complexity and power consumption, alongside reduced clock speed, elongated pipelines, and diminished throughput capabilities compared to integer or fixed-point multipliers. In numerous image processing algorithms, dense floating-point rectangular matrix multiplication is conducted, leading to prolonged runtime for floating-point matrix multiplication. To mitigate this issue, companies typically opt to convert some or all of the algorithms to a fixed point rather than simply augmenting the number of logic devices in the design. This necessitates interfacing a fixed point to a floating-point conversion unit. For instance, the Texas Instrument DRA-78x chip facilitates 2 C66x floating-point VLIW digital signal processing with a 16x16 fixed-point multiplier architecture. Similarly, the Blackfin ADSP BF592 also integrates a 16x16 fixed-point multiplier architecture within a 16-bit MAC. Consequently, the pursuit of designing a faster fixed-point multiplier unit remains pertinent for ASIC designers. The multiplication operation entails a sequence of addition and shift operations, iterated for each operand value, thus higher operand values result in more repetitions and increased delay in yielding the final product. Nonetheless, employing faster algorithms can generate the final product with minimal latency by generating intermediate product terms known as partial products and accumulating them using the shifting method to produce the final result [XXII]. While various faster algorithms for multiplication have been reported, the existing literature also encompasses numerous brief reviews of multiplier architectures, albeit predominantly focusing on radix multiplication or high-speed multiplication techniques. To my understanding, no comprehensive review examines a range of low-power fixed-width multiplier architectures, particularly emphasizing their adaptation to different operand sizes. The novelty of this paper lies in its succinct overview of the latest multiplier architectures implemented at the 45nm technology node, catering to low-power niche DSP chips.

The subsequent sections of the paper are organized as follows: Section 2 delves into the details of various algorithms and architectures employed in modern digital multipliers. Section 3 outlines the historical progression of low-power logic, while sections 4 and 5 expound on various low-power multiplier architectures and their performance analysis, respectively. Finally, the paper concludes based on a comprehensive review of low-power multiplier architectures.

II. Different Types of Digital Multiplier Architecture

Numerous faster multiplication algorithms can be found in the open literature [IX], [XIII], [XIV], [XVII], [XXIII], [XXIV], [XXVII] although only a select few are discussed here.

II.i. Serial Multiplier

In serial multiplication, all partial products are generated sequentially by a clocked adder unit. The circuit depicted in Fig.1 takes P and Q as serial multiplicand and multiplier inputs [III]. When employing clocked synchronization, the duration required for product generation correlates directly with the dimensions of both the multiplier and the multiplicand. Since all partial products are generated sequentially, this multiplication algorithm is not suitable for operands with higher values. Nevertheless, while this type of multiplier may be slower, it demonstrates excellent power and area

utilization.

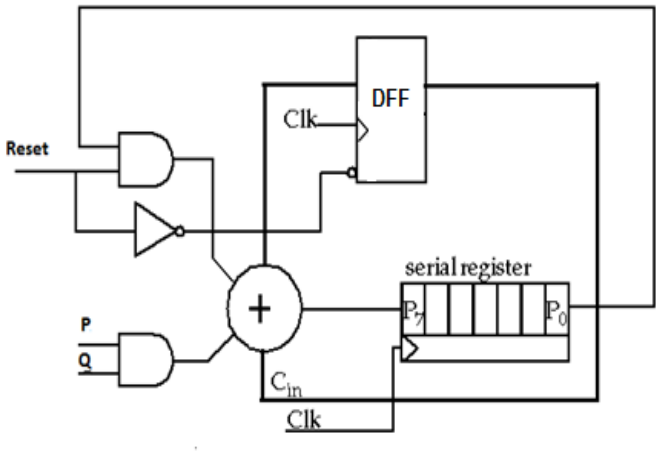

Fig.1. Schematic of a PxQ Serial Multiplier [III]

II.ii. Array Multiplier

The most prevalent multiplier architecture utilized for multiplication with lower operand sizes is the array multiplier. In this architecture, straightforward 1-bit multiplier blocks execute bitwise multiplication. Fig.2 portrays the fundamental building block of an array multiplier, while Fig.3 illustrates a block diagram of a 4x4 multiplier.

The fundamental building block comprises an AND gate and a FULL ADDER (FA). These elements compute the partial product (X.Y) for the corresponding location. The prior sum in and carry-in are added to the conjunction of the X and Y input bits. The outcomes of this operation generate a carry-out bit (Cout) and a new sum-out bit (Sum Out), subsequently forwarded to the next stage. By interconnecting these building blocks in a position-wise manner, a larger operand multiplier unit can be configured.

Fig.3 showcases a 4x4 combinational array multiplier constructed using these fundamental building blocks. The multiplicand bits (Ai) are distributed along the block diagonals, while the multiplier bits (Bi) propagate along the rows. An optimized iteration of the 4x4 array multiplier architecture can be developed using an optimal number of AND gates, full adders (FA), and half adders (HA), as depicted in Fig. 4 [XXI].

Although this multiplier presents a simple architecture for low operand sizes, it encounters challenges concerning both area and speed as the operand size escalates.

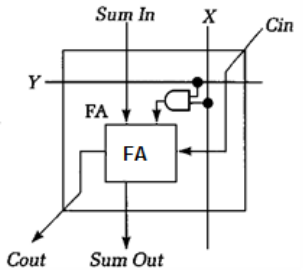

Fig. 2. The Basic building block of an array multiplier

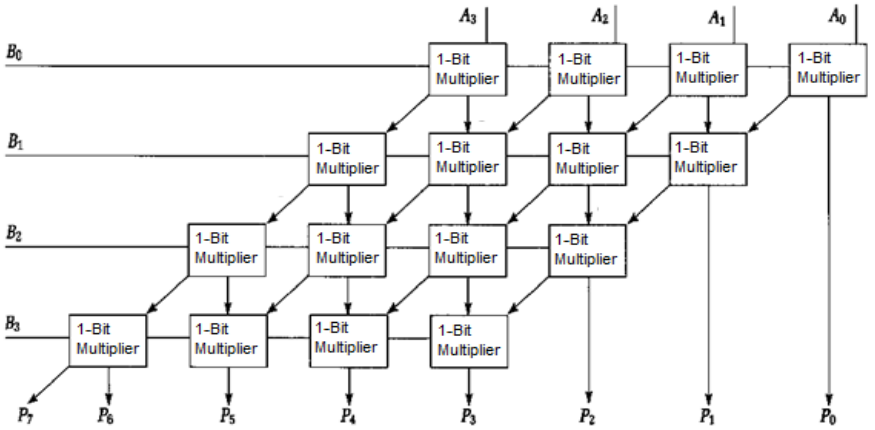

Fig. 3. Architecture of a 4x4 array multiplier using 16 numbers of basic building blocks shown in Fig.2 [VIII]

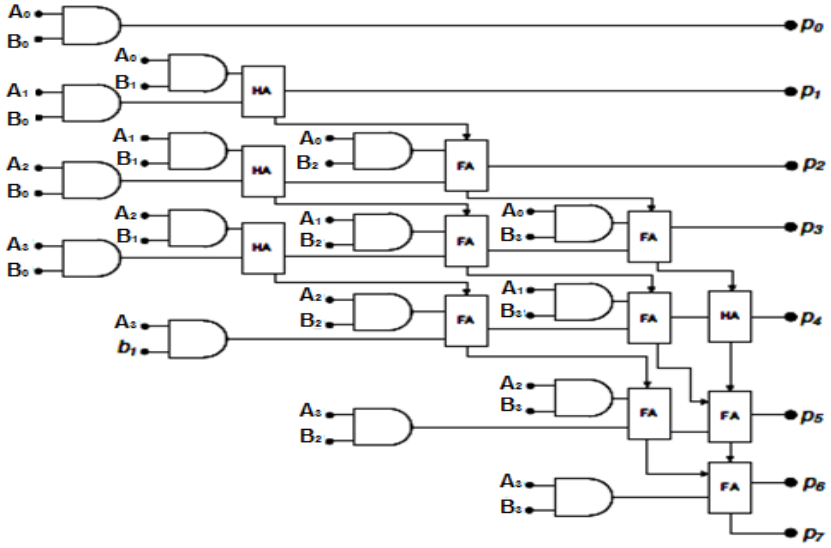

Fig. 4. 4x4 array multiplier architecture using 16 numbers of AND-2 gate, 4 numbers of HA, and 8 numbers of FA [VIII]

II.iii. Vedic Multiplier

Vedic mathematics algorithms, rooted in the ancient Indian system of mathematics, employ sixteen distinct Sutras or principles for highly efficient computation [XXIII]. According to Vedic mathematics, a multiplier can be implemented with reduced latency. Fig. 5 illustrates a basic schematic of a 2x2 Vedic multiplier utilizing AND gates and half adders (HA).

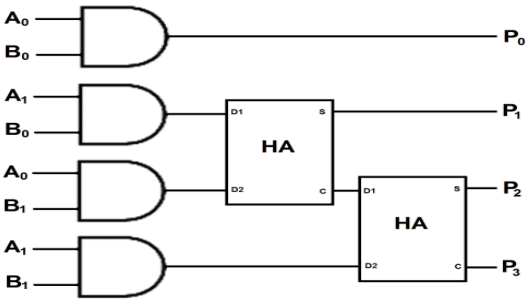

Fig. 5. Block diagram of 2x2 Vedic multiplier using AND gates and HA

To execute multiplication using Vedic mathematics, the higher operands are initially grouped by considering 2 bits each. For instance, as illustrated in Fig. 6, a four-bit multiplicand and multiplier are grouped into (A1A0), (A3A2), and (B1B0), (B3B2) respectively. Subsequently, the grouped bits are multiplied using a 2x2 bit Vedic multiplier block in the sequence depicted in Fig. 6. Finally, the corresponding partial products are accumulated using ripple carry adders (RCA).

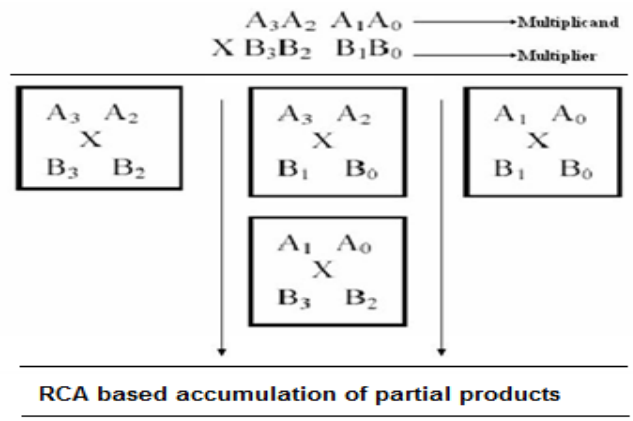

Fig. 6. 4x4 Vedic multiplication schemes using 2x2 Vedic multiplier unit [IV]

II.iv. Booth Multiplier

For enhanced speed and scalability with larger operand sizes, the signed bit multiplication Booth algorithm emerges as a valuable tool. These multipliers exhibit accelerated response times because the Booth algorithm reduces the count of intermediate partial products. Unlike the array multiplier, where partial products are generated in a Radix-2 manner, here, the partial products are formed in Radix-2n. The

exact value of 'n' can be ascertained by examining the grouping of bits that are being considered within the multiplier. For instance, in a radix-4 system, an M-bit multiplier yields M/2 partial products. Similarly, in a radix-8 system, the number of partial products amounts to M/3, and so forth [XVII], [XXVII].

II.v. Wallace Tree multiplier

The Wallace Tree multiplier stands out as a valuable method for expediting the accumulation of partial products. A multiplier structured upon the Wallace tree architecture undertakes the accumulation of partial products in three distinct stages. Initially, adjacent rows are grouped for partial product accumulation. Subsequently, these groups undergo reduction via circuits such as half adders (HA), full adders (FA), compressors, or counters. Further enhancements to the Wallace tree architecture can be explored in [XXVI]. Recent advancements documented in the open literature have significantly bolstered the overall multiplier performance by introducing novel architectures of counters [XII], [XXVIII], [XXX]. Fig. 7 illustrates a generalized architecture of the NxN bit Wallace tree architecture.

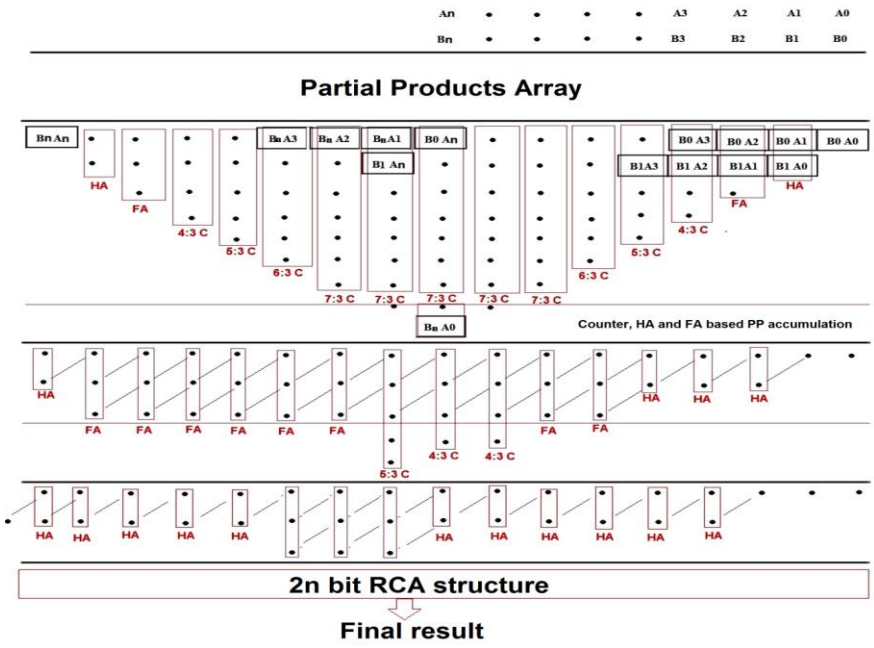

Fig. 7. Generalized architecture of NxN Wallace tree Multiplier [II]

II.vi. Modified Booth Wallace Multiplier

The Radix-4 Modified Booth Wallace Multiplier architecture combines the Booth multiplication algorithm for generating partial products with a Wallace tree architecture for their accumulation [XXII]. This innovative approach merges the benefits of reduced complexity in partial product generation from the Booth algorithm with the swift accumulation technique utilized in the Wallace tree structure. With its streamlined complexity, this type of multiplier architecture exhibits exceptional performance, particularly at high data rates.

In this section, a range of digital multiplication techniques has been explored. Some of these techniques, such as the serial multiplier and array multiplier, boast simple architectures. However, they encounter increased latency as the operand size grows. On the other hand, faster multipliers like Vedic, Booth, or Modified Booth Wallace yield quicker results but entail substantial power consumption owing to their circuit complexity. Consequently, the design of low-power, fast multipliers has emerged as a primary concern for researchers over the past few decades.

III. History of Low-Power Logic

Power consumption of Integrated Circuits (ICs) wasn't a significant concern until the early 1990s, except for specialized components. However, with the progression of niche devices, low-power Very Large Scale Integration (VLSI) design has garnered considerable attention over the past three decades. The conventional approach to low-power VLSI design relies on complementary metal-oxide-semiconductor (CMOS) logic. The power consumption of any CMOS VLSI circuit can be classified into two primary categories: static power and dynamic power. Fig. 8 illustrates the various parameters that contribute to the power consumption of an IC.

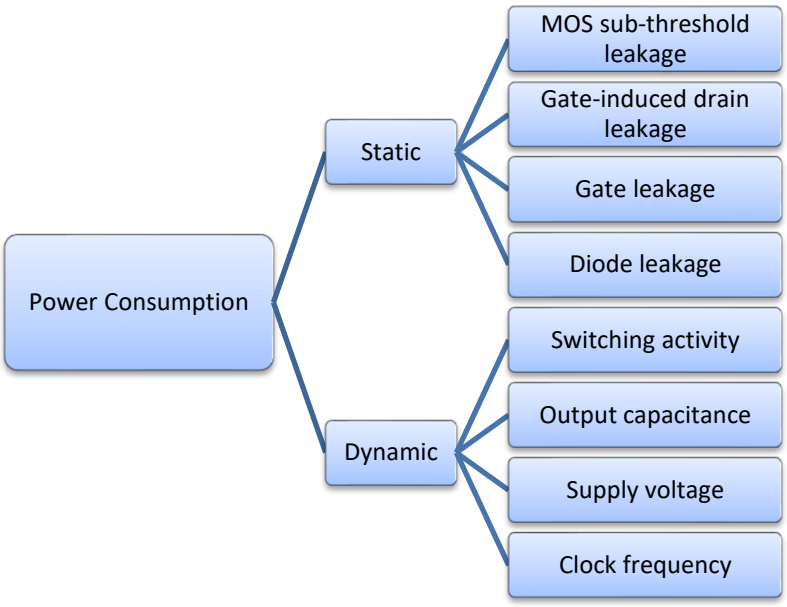

Fig. 8. Different types of power consumption in an IC

In the majority of modern integrated circuits (ICs), dynamic power consumption predominates over static power consumption, primarily due to the intricate design complexity and the frequency of operation. The dynamic power consumption of a circuit can be articulated as in equation (1).

$$P_{dynamic} = \alpha . f . C . V^2 \quad (1)$$

From equation (1), it is apparent that the dynamic power consumption of a circuit can be reduced by reducing the switching activity (α), operating frequency (f), output capacitance (C), and biasing voltage (V) of the circuit [XI]. Techniques such as PTL, CPL, Domino logic, DCVS, MCML, C2MOS, and DPL are employed to decrease the output capacitance of a circuit [XX][XXV]. Moreover, decreasing the number of transistors in a design directly reduces the output capacitance and switching activity of any circuit. Consequently, the gate diffusion input (GDI) technique represents a significant step towards low-power design [I].

IV. Various Low Power Architectures for Multiplier

With the introduction of various low-power logic techniques, multiplier circuits have been undergoing modifications to incorporate these advancements. In the early 2000s, a low-power fixed-width array multiplier architectures were documented in [XXXI], [XXXIII]. Subsequently, Dastjerdi et al. introduced a serial low-power multiplier architecture in 2009 [XIX], as depicted in Fig. 9.

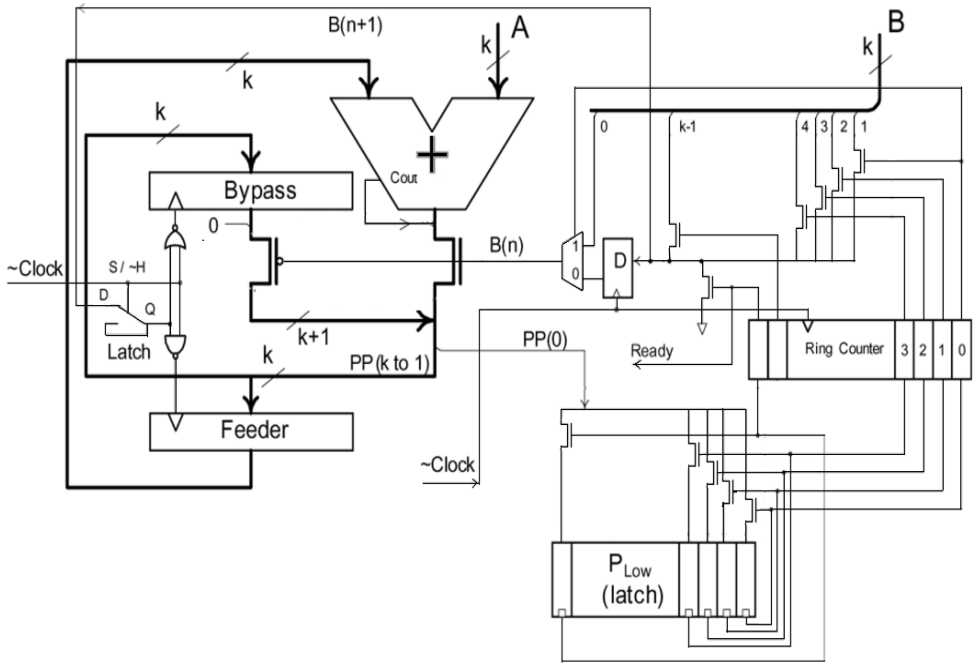

Fig. 9. BZ-FAD Low power multiplier architecture [XIX]

This architecture [XIX] exhibits a 30% reduction in power consumption compared to [XXXI]. To achieve a notable decrease in power consumption for a multiplier, Zhang et al. implemented dynamic voltage scaling in the design [XXXII]. Dynamic voltage scaling represents one of the latest approaches in low-power VLSI design [XVIII]. The overall architecture of the dynamic voltage scaling multiplier is depicted in Fig. 10.

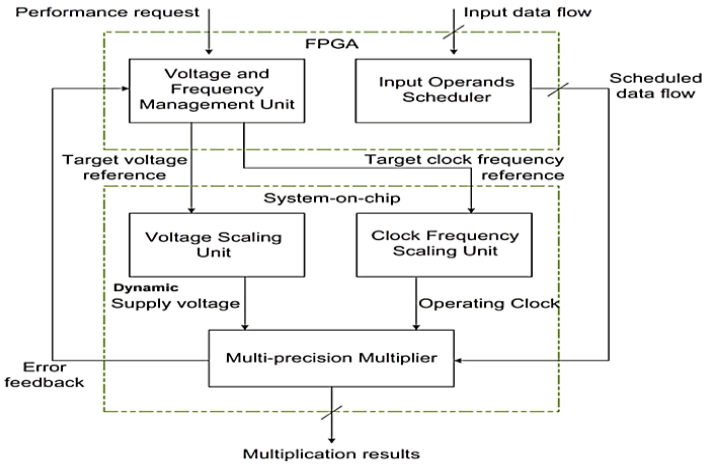

Fig. 10. Dynamic Voltage Scaling Multiplier Architecture [XXXII]

The multiplier system depicted in Fig. 10 comprises five distinct modules. The first module is a multi-precision multiplier capable of operating at various supply voltages and frequencies, dictated by the input operand scheduler (IOS), which constitutes the second module. Additionally, two supplementary modules, namely a voltage-controlled oscillator (VCO) and a voltage scaling unit (VSU), are integrated for dynamic voltage scaling. Lastly, a dynamic voltage/frequency management unit (VFMU) is incorporated to accommodate user requirements.

Recently, GDI and CCGDI [VII] logic techniques have been introduced in various multiplier architectures [II], [III], [IV], [V] [VIII],. The architecture of a CCGDI-based modified Booth Wallace (MBW) multiplier architecture is illustrated in Fig. 11.

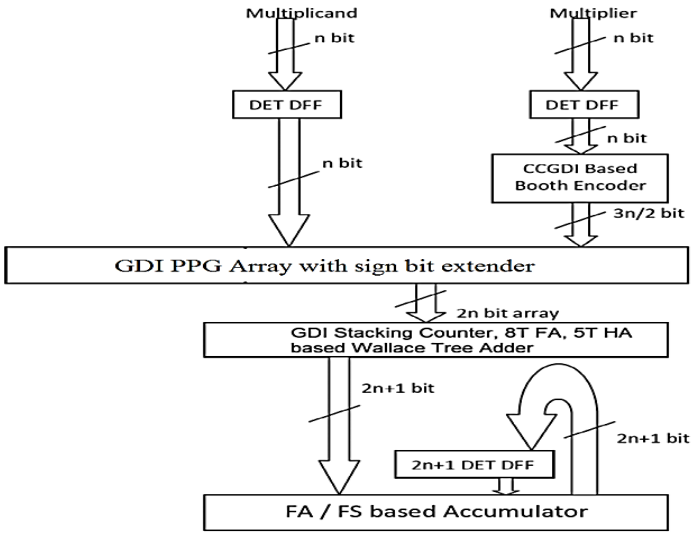

Fig. 11. CCGDI-based MBW Multiplier [V]

The optimal architecture was discovered in 2020 [VI] by combining the CCGDI technique with dynamic voltage scaling. The architecture is depicted in Fig. 12.

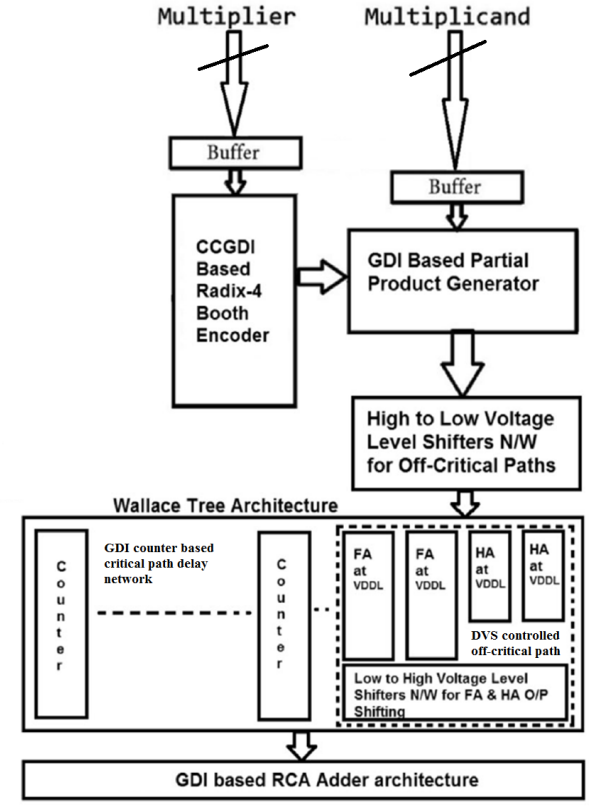

Fig. 12. DVS controlled CCGDI based MBW Multiplier [VI]

The architecture encompasses the CCGDI Booth Encoder and GDI partial product generator. Dynamic voltage scaling (DVS) is integrated into off-critical delay paths through the utilization of high-to-low level shifters and low-to-high voltage level shifters. Additionally, the critical path delays of higher-order partial products are accumulated by low-power parallel full adder-based counters.

Floating-point multiplier architectures are known for their intricate design and extended runtimes. In addressing this challenge, Gao et al. introduced pipelined IEEE 754-2008 decimal floating-point (DFP) multipliers utilizing fixed-point multipliers [XXIX]. However, the implementation of such a multiplier necessitates FPGA involvement, leading to unnecessary power consumption. More recently, a novel data type known as the posit data type has emerged as a replacement for floating-point numbers [XVI]. Zhang et al. pioneered the first posit data type multiplier in 2020 [XV], with the schematic of the low-power posit multiplier depicted in Fig. 13. Unlike floating-point multipliers, it boasts a reduced runtime. Further modifications can be explored in [X].

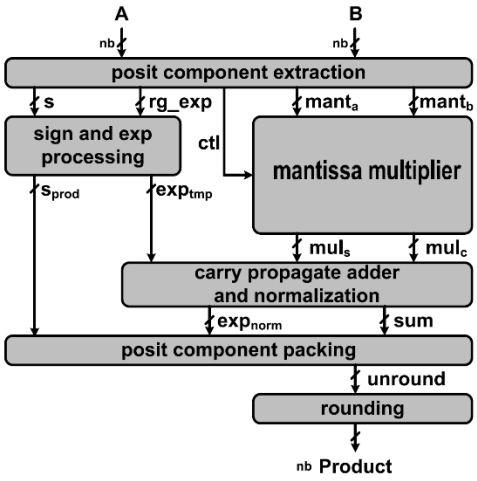

Fig. 13. Low power posit Multiplier [XV]

V. Results and Performance Analysis of Various Low-Power Multipliers

The performance evaluation of a VLSI circuit primarily revolves around core design layout area, power consumption, noise margin, and output delay. Although other parameters such as energy per transition, signal-to-noise rejection ratio, and driving capability are noteworthy, for the sake of simplicity, this article concentrates solely on the primary parameters.

Firstly, the layout area of the chip is analyzed, with many multiplier architectures necessitating an additional field programmable gate array (FPGA) board for logic operation control [XXXII]. Thus, the layout area analysis is confined to the core multiplier section, focusing solely on the transistor count.

For power and delay measurements, simulations are conducted using Computer-Aided Design (CAD) tools. The operand sizes are set at 4-bit, 8-bit, and 16-bit, respectively, with the circuits simulated using a typical transistor (TT) process corner at 45nm technology, operating at a frequency of 1MHz.

For ratioless architectures built with CMOS, Gate Diffusion Input (GDI), and Complementary Current-mode Logic with Gate Diffusion Input (CCGDI) techniques, the aspect ratios of NMOS transistors are set at 2/1, and the PMOS transistors are set at 5/1. In the DVS-controlled multiplier architecture, which is a ratioed design, the aspect ratios remain consistent with the reported design [VI].

The noise margin is evaluated through static DC analysis at a biasing voltage of 1 volt, as reported in [VI], where the noise margin is measured for the high voltage level. The detailed analysis report is tabulated in Table 1.

A comparative analysis among various fixed-width multiplier architectures reveals noteworthy findings. For instance, the BZ-FAD Multiplier [XIX] demonstrates a 77% improvement in Power-Delay Product (PDP) compared to the conventional CMOS array multiplier XXII. The Vedic multiplier [XXIII] exhibits excellent speed, boasting a 69% better latency than the BZ-FAD model. GDI-based architectures show

significantly lower power consumption compared to conventional CMOS architectures, with the GDI Radix-4 MBW Multiplier [III] consuming 32% less power than the GDI Wallace tree architecture [II]. The DVS-controlled CCGDI technique-based Radix-4 MBW multiplier [VI] stands out for its exceptional response, with a PDP of 585 fJ for a 16-bit operand.

Comparatively, floating-point multiplier architecture fares poorer in terms of response when juxtaposed with fixed-point architectures. The Posit multiplier exhibits inferior latency and power characteristics compared to IEEE floating-point multiplier architecture, albeit boasting a much higher dynamic range compared to other multiplier architectures.

Table 1: Simulation-based performance of various low power multiplier architectures in terms of transistor count, latency, power, and PDP at 45nm technology (TT process corner)

Design	Operand size	Latency (ns)	Power (µw)	PDP (fJ)	Area (Tr. count)
CMOS Array Multiplier [XXII]	4 bits	1.39	52.91	73.54	544
	8 bits	6.75	224.56	1515.78	2176
	16 bits	12.16	8454.65	102808	8704
CMOS Vedic Multiplier [XXIII]	4 bits	1.035	75.49	78.13	240
	8 bits	1.630	379.56	618.68	960
	16 bits	4.985	9875.17	49227.71	3840
BZ-FAD Multiplier [XIX]	4 bits	2.23	29.45	65.67	698
	8 bits	8.14	42.39	345.05	758
	16 bits	16.18	206.85	3346.83	948
GDI Array Multiplier [VIII]	4 bits	1.29	39.98	50.02	160
	8 bits	6.35	168.74	1071.51	640
	16 bits	11.76	6543.98	76957	2.5k
GDI Vedic Multiplier [IV]	4 bits	0.9115	124.34	85.56	176
	8 bits	1.134	478.98	541.15	698
	16 bits	4.675	2098.56	9797.6	2.8K
GDI Wallace Tree Multiplier [II]	4 bits	1.22	12.87	13.028	137
	8 bits	1.85	68.96	127.57	543
	16 bits	2.64	423.54	1118.10	2.2K

Radix-8 Booth Multiplier [XVII]	4 bits	0.994	148.87	147.97	564
	8 bits	2.789	408.61	1135.9	2.3K
	16 bits	3.56	2382.76	8480.90	8.4K
Radix-4 Booth Multiplier [XIV]	4 bits	0.785	86.23	67.66	488
	8 bits	1.94	120.34	233.46	1.4K
	16 bits	3.25	450.87	1465.30	6.1K
GDI Radix -4 MBW Multiplier [III]	4 bits	1.31	18.34	20.55	302
	8 bits	2.09	60.19	125.79	944
	16 bits	3.56	285.98	1018.10	1.5K
CCGDI Radix -4 MBW Multiplier [V]	4 bits	1.28	17.67	22.61	300
	8 bits	1.89	57.45	108.58	938
	16 bits	3.23	257.47	831.62	1.5K
DVS controlled CCGDI Radix-4 MBW Multiplier [VI]	4 bits	1.35	18.45	24.90	228
	8 bits	1.76	40.08	70.54	1.1K
	16 bits	2.45	238.98	585.50	1.6K
Floating point multiplier using fixed point multiplier [XXIX]	4 bits	2.85	28.15	80.22	1098
	8 bits	3.86	65.18	251.59	1258
	16 bits	4.65	406.18	1888.73	1948
Low power posit multiplier [XV]	4 bits	1.23	39.45	48.52	878
	8 bits	3.14	62.39	195.90	1165
	16 bits	6.18	231.85	1432.83	2.2k
Approximate posit multiplier [X]	4 bits	4.51	42.34	190.95	952
	8 bits	8.89	88.19	784.00	1244
	16 bits	13.56	285.98	3877.88	2.5K

Fig. 14 illustrates the latency, power consumption, noise margin, and Power-Delay Product (PDP) characteristics of various state-of-the-art low-power multiplier architectures. The results highlight several key findings:

- The Radix-4 Booth multiplier achieves a faster response compared to the Radix-8 Booth multiplication.
- The BZ-FAD multiplier demonstrates outstanding power efficiency compared to other architectures, although its latency suffers at higher operating frequencies.

- CMOS-based circuits generally exhibit superior noise margin compared to GDI-based architectures. Specifically, the CMOS array multiplier showcases a 46mV NMH, which is the best among all other designs. However, the DVS-controlled CCGDI-based architecture demonstrates a remarkable 44% improvement in NML compared to the BZ-FAD design.
- For higher operands of multiplication, the CCGDI multiplier with DVS control displays the best response in terms of Power-Delay Product (PDP).

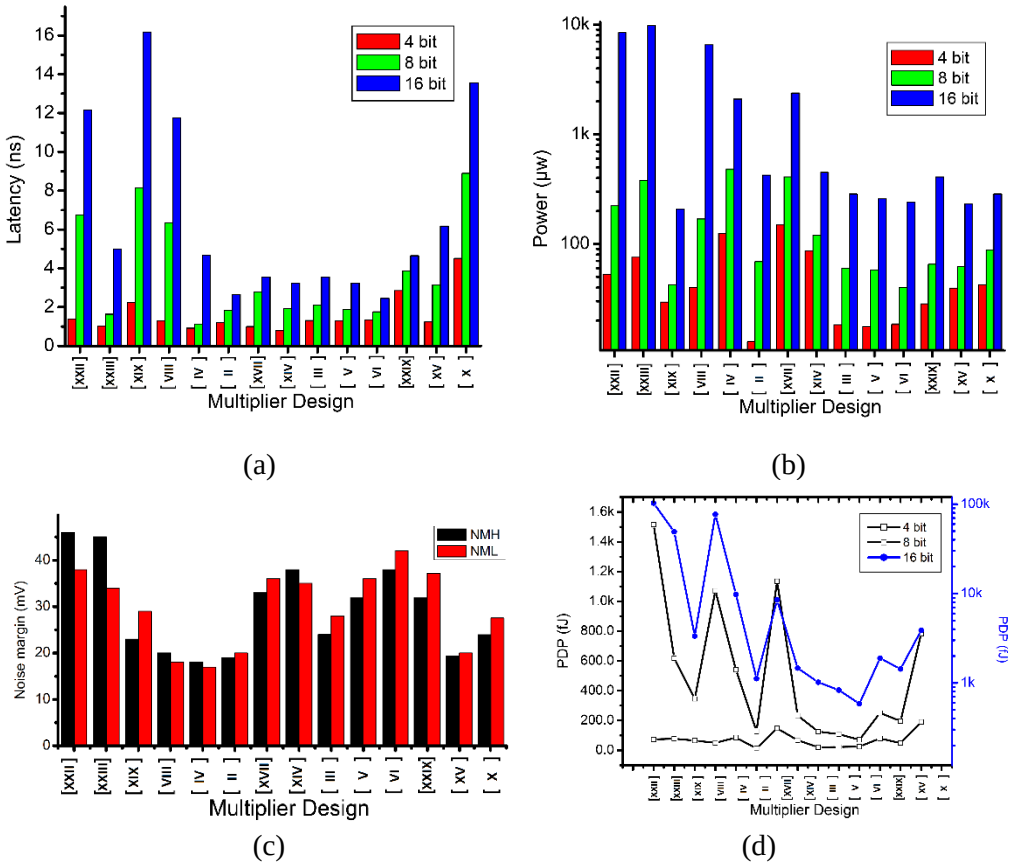

Fig. 14. Performance characteristics of various state-of-art low-power multiplier architectures. (a) Latency at 1MHz frequency, (b) Power consumption at 45 nm TT technology, (c) Noise margin at biasing voltage 1 volt, (d) PDP at different operand size

VI. Conclusion

This paper offers a succinct overview of various cutting-edge fixed-width low-power multiplier architectures. The simulation results, conducted using 45nm technology, encompass average power consumption, latency, noise margin, and Power-Delay Product (PDP), covering the primary performance metrics. The multipliers underwent testing across different operand sizes at an operational speed of 1 MHz. Upon analyzing the outcomes, several conclusions can be drawn. The conventional CMOS array multiplier demonstrates excellent performance for lower operand

multiplications, boasting the best noise margin. However, for high-speed, high-operand multiplication tasks, array multipliers exhibit shortcomings in terms of area and delay. Various algorithms for high-speed multiplication were explored, with the Vedic multiplication algorithm theoretically showing faster response times. However, in practice, the Vedic multiplier suffers from poor latency due to circuit complexity. Among the investigated multiplication algorithms, the Radix-4 modified Booth Wallace tree algorithm emerges as the top performer in ASIC design for higher operand operations. To render it suitable for low-power applications, the DVS-controlled CCGDI technique is employed, resulting in a 57% performance improvement compared to conventional CMOS-based designs. Floating-point multiplier architectures are generally less favored over fixed-point architectures due to their higher runtime, longer pipeline, and reduced throughput. The Posit multiplier exhibits poor performance in terms of latency and power compared to the IEEE floating-point multiplier architecture. Nevertheless, the Posit multiplier boasts a significantly higher dynamic range compared to other multiplier architectures.

Conflicts of Interest

The author declared no conflict of interest. All related prior publications in other journals or conferences have been fully declared.

References

I. Arkadiy Morgenshtein, Alexander Fish, and Israel A. Wagner. : 'Gate-Diffusion Input (GDI): A Power-Efficient Method for Digital Combinatorial Circuits'. *IEEE Transaction on VLSI Systems*. Vol. 10(5), pp. 566-581, October 2002. 10.1109/TVLSI.2002.801578

II. B. Mukherjee, A Ghosal. : 'Counter Based Low Power, Low Latency Wallace Tree Multiplier Using GDI Technique for On-chip Digital Filter Applications'. *IEEE International Conference on Devices for Integrated Circuit (DevIC)*, March 2019. 10.1109/DEVIC.2019.8783456

III. B. Mukherjee, A Ghosal. : 'Design and Analysis of a Low Power High Performance GDI Based Radix 4 Multiplier Using Modified Booth Wallace Algorithm'. *IEEE Electron Device Kolkata Conference* (2018 IEEE EDKCON), 2018. 10.1109/EDKCON.2018.8770494

IV. B. Mukherjee, A Ghosal. : 'Design and Implementation of Low Power, High Speed, Area Efficient Gate Diffusion Input Logic Based Modified Vedic Multiplier for Digital Signal Processor'. *RAICMHAS International Conference*. 2019.

Biswarup Mukherjee

V. B. Mukherjee, A. Ghosal. : 'Design of a low power, double throughput CCGDI based radix-4 MBW multiplier and accumulator (MAC) unit for on-chip RISC processors of MEMS sensor'. *Journal of Micromechanics and Microengineering*. Vol. 29(6), pp. 064003, 2019. 10.1088/1361-6439/ab1504

VI. B. Mukherjee, A. Ghosal. : 'Low Power Dynamic Voltage Scaling CCGDI Based Radix-4 MBW Multiplier Using Parallel HA and FA Counters for On-Chip Filter Applications'. *Sadhana, Academy Proceedings in Engineering Sciences.* Vol 45(1), article id: 0119, 2020. 10.1007/s12046-020-01340-2

VII. B. Mukherjee, B Roy, A Biswas, A Ghosal. : 'Cyclic Combinational Gate diffusion input (CCGDI) technique-a new approach of low power digital combinational circuit design'. *IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. pp. 1-6, 2015. 10.1109/ICECCT.2015.7226150

VIII. B. Mukherjee, B Roy, A Biswas, A Ghosal. : 'Design of a low power 4×4 multiplier based on five transistor (5-T) half adder, eight transistor (8-T) full adder & two transistor (2-T) AND gate'. *IEEE 2nd International conference on Computer, Communication, Control and Information Technology (C3IT)*. 2015. 10.1109/C3IT.2015.7060143

IX. Binta Lamba, Anurag Sharma. : 'A review paper on different multipliers based on their different performance parameters'. *2nd International Conference on Inventive Systems and Control (ICISC)*. Pp.324-327, 2018. 10.1109/ICISC.2018.8399088

X. C. J. Norris and S. Kim. : 'An Approximate and Iterative Posit Multiplier Architecture for FPGAs'. *IEEE International Symposium on Circuits and Systems (ISCAS)*, Daegu, Korea 2021, pp. 1-5, 10.1109/ISCAS51556.2021.9401158.

XI. Christian Piguet. 'Low-Power CMOS Circuits Technology, Logic Design and CAD Tools'. *CRC Press*. Taylor & Francis Group, 2006

XII. Christopher Fritz , Adly T. Fam. 'Fast Binary Counters Based on Symmetric Stacking'. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. Vol. 25(10), pp. 2971–2975, 2017. 10.1109/TVLSI.2017.2723475

XIII. Glenn Colón-Bonet, Paul Winterrowd. : 'Multiplier Evolution: A Family of Multiplier VLSI Implementations'. *The Computer Journal*. Vol. 51(5), pp. 585-594, 2008. 10.1093/comjnl/bxm123

XIV. H. Xue, R. Patel, N. V. V. K. Boppana, S. Ren. : 'Low-power-delay-product radix-4 8*8 Booth multiplier in CMOS'. *Electronics Letters*. Vol. 54(6), pp. 344-346, 2018. 10.1049/el.2017.3996

XV. H. Zhang and S. -B. Ko. : 'Design of Power Efficient Posit Multiplier'. *IEEE Transactions on Circuits and Systems II: Express Briefs*. Vol. 67(5), pp. 861-865, May 2020, 10.1109/TCSII.2020.2980531.

XVI. J. L. Gustafson and I. Yonemoto. : 'Beating floating point at its own game: Posit arithmetic'. *Supercomput. Front. Innovat. Int. J.*, Vol. 4(2), pp. 71–86, Jun. 2017.

XVII. Jiang H, Han J, Qiao F, Lombardi F., : 'Approximate Radix-8 Booth Multipliers for Low-Power and High-Performance Operation'. *IEEE Transactions on Computers*. Vol.65, pp: 2638–2644, 2016. 10.1109/TC.2015.2493547

XVIII. M. Lanuzza, P. Corsonello and S. Perri. : 'Fast and Wide Range Voltage Conversion in Multisupply Voltage Designs'. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. Vol. 23(2), pp. 388-391, 2015. 10.1109/TVLSI.2014.2308400.

XIX. M. Mottaghi-Dastjerdi, A. Afzali-Kusha and M. Pedram. : 'BZ-FAD: A Low-Power Low-Area Multiplier Based on Shift-and-Add Architecture'. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. Vol. 17(2), pp. 302-306, Feb. 2009. 10.1109/TVLSI.2008.2004544

XX. Md. W. Allam. : 'New Methodologies for Low-Power High-Performance Digital VLSI Design'. *PhD. Thesis, University of Waterloo, Ontario, Canada*, 2000.

XXI. N.Anuar, Y. Takahashi,T. Sekine. : '4×4-bit array two phase clocked adiabatic static CMOS logic multiplier with new XOR'. *IEEE VLSI System on Chip Conference (VLSI-SoC)*. Pp.364-368, 2010. 10.1109/VLSISOC.2010.5642688

XXII. Neil Weste and David Harris. : 'CMOS VLSI Design: A Circuits and Systems Perspective'. *Addison-Wesley Publishing Company, (4th. ed.), USA*, pp: 476-490, 2010.

XXIII. P. Mehta and D. Gawali. : 'Conventional versus Vedic Mathematical Method for Hardware Implementation of a Multiplier'. *International Conference on Advances in Computing, Control, and Telecommunication Technologies*. pp. 640-642, 2009. 10.1109/ACT.2009.162

XXIV. R. Patel, Y. S. Chauhan. : 'Multiplier design and performance comparison using reversible and conventional logic gates'. *Annual IEEE India Conference (INDICON)*, New Delhi, 2015. pp. 1-6, 10.1109/INDICON.2015.7443439.

XXV. Rabaey J.M., A. Chandrakasan, B.Nikolic. : 'Digital Integrated Circuits, A Design'. *2nd 2002, prentice Hall, Englewood Cliffs,NJ*.

XXVI. Ron S. Waters, Earl E. Swartzlander. : 'A Reduced Complexity Wallace Multiplier Reduction'. *IEEE Transaction on Computers*. Vol.59(8), pp. 1134-1137, 2010. 10.1109/TC.2010.103

XXVII. S. K. Chen, C. W. Liu, T. Y. Wu and A. C. Tsai. : 'Design and Implementation of High-Speed and Energy-Efficient Variable-Latency Speculating Booth Multiplier (VLSBM)'. *IEEE Transactions on Circuits and Systems I: Regular Papers*. Vol. 60(10), pp. 2631-2643, 2013, 10.1109/TCSI.2013.2248851.

XXVIII. Shahzad Asif, Yinan Kong. : 'Design of an algorithmic Wallace multiplier using high speed counters'. *Proceedings of IEEE International Conference on Computer Engineering & Systems (ICCES)*, Cairo, Egypt. PP: 133 - 138, 2015. 10.1109/ICCES.2015.7393033

XXIX. Shuli Gao, D. Al-Khalili, J. M. P. Langlois and N. Chabini. : 'Decimal floating-point multiplier with binary-decimal compression based fixed-point multiplier'. *IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Windsor, ON, Canada, 2017, pp. 1-6, 10.1109/CCECE.2017.7946692.

XXX. Sreehari Veeramachaneni, Lingamneni Avinash, M. Kirthi Krishna, M.B. Srinivas. : 'Novel Architectures for Efficient (m, n) Parallel Counters'. *Proceedings of ACM Great Lakes Symposium on VLSI* . Stresa - Lago Maggiore, Italy. Vol. 11(13), PP: 188-191, 2007. 10.1145/1228784.1228833

XXXI. Wang J. S, Kuo C. S, and Yang T. H., : 'Low-power fixed-width array multipliers'. *Proc. IEEE Symp. Low Power Electron*. Des. pp. 307–312, 2004. 10.1109/LPE.2004.241092

XXXII. X. Zhang, F. Boussaid and A. Bermak. : '32 Bit X 32 Bit Multiprecision Razor-Based Dynamic Voltage Scaling Multiplier With Operands Scheduler'. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. Vol. 22(4), pp. 759-770, 2014. 10.1109/TVLSI.2013.2252032

XXXII. Z. Huang and M. D. Ercegovac. : 'High-performance low-power left to right array multiplier design'. *IEEE Trans. Computers*. Vol. 54(2), pp. 272–283, 2005. 10.1109/TC.2005.51