

# Accelerated PVT Analysis of UCM Architecture using Cadence ADE-XL

Rajkumar Sarma, Cherry Bhargava, Shruti Jain

**Abstract:** A Process-Voltage-Temperature (PVT) Variation check is run on the novel Universal Compressor based Multiplier (UCM) architecture, which promises for fast multiplication in ultra-low supply voltages (less than 0.9 V) for higher order operation. The analysis further shows that for 5x5 bit & 9x9 bit operation with supply voltage as low as 0.6 V, the delay has reduced by 0.73% & 5.05% (mean values) respectively than Wallace tree multiplier architecture. The analysis is carried out in Cadence Spectre tool using ADE-XL at CMOS 90 nm technology.

**Index Terms:** Multiplier, Compressor design, ADE-XL, Low power, High speed, Cadence Virtuoso, PVT analysis, Delay optimization.

## I. INTRODUCTION

A multiplier is a key element in a digital system. For the applications such as digital signal processing, digital image processing, Multiply and Accumulate architecture etc. uses multiplier rigorously. Though a multiplication is performed using repetitive addition method, where a multiplicand is added with itself for as many as number of times the multiplier, in digital system the multiplication process is slightly different. A basic design of a multiplier is as shown in the Fig. 1. As shown in the Fig. 1, the multiplicand's & the multiplier's individual terms are ANDed to produce the partial products & positioned as per their weights. For example, as shown in Fig. 1, 'A2B0', 'A1B1' & 'A0B2' are aligned in a single column because the weight is two for all of the mentioned partial products. i.e. the summation of the bit location is any of 2+0, 1+1, 0+2, which are in all cases is equal to 2. Hence, for the addition of partial products, its alignment is vital. At the next step, the partial product with same weights are added using fulladder (in the case of 3 partial products), halfadder (in the case of 2 partial products) or any compressor circuit (for adding 'n' number of partial products simultaneously). In this research paper the novel UCM architecture as proposed in [1] is further validated with the PVT analysis in Cadence spectre tool using ADE-XL in 90 nm CMOS technology. The UCM architecture uses a novel compressor-based multiplier algorithm which reduces the delay substantially.

Revised Manuscript Received on June 27, 2019.

**Rajkumar Sarma**, School of Electronics & Electrical Engineering, Lovely Professional University, Phagwara, Punjab 144411, India

**Cherry Bhargava**, School of Electronics & Electrical Engineering, Lovely Professional University, Phagwara, Punjab 144411, India

**Shruti Jain**, Department of Electronics and Communication Engineering, Jaypee University of Information Technology, Waknaghat, Himachal Pradesh 173234, India

The following sections are discussed as follows: in section 2, various different notable architectures related to multiplier are discussed in detail, in section 3, a quick review on the novel UCM architecture has been explained, in section 4, a detailed PVT analysis of the UCM architecture is discussed & in section 5, conclusion, future scopes & application of the UCM architecture is discussed.

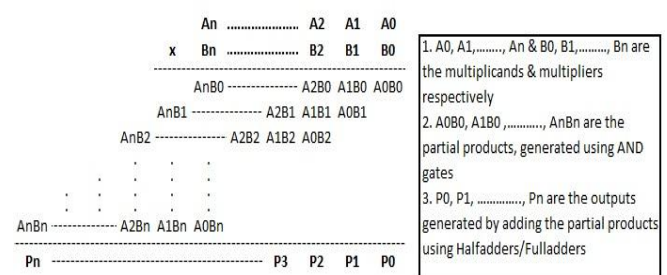


Fig. 1, Basic multiplication operation

## II. VARIOUS MULTIPLIER ARCHITECTURES

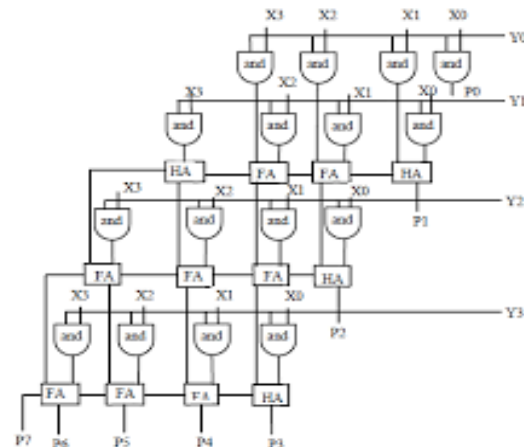
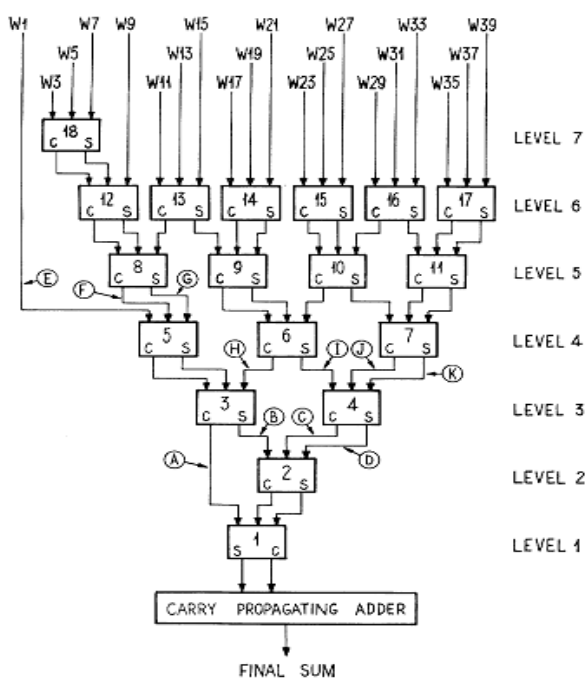


Fig. 2, Array multiplier architecture

In any application, the processing elements basically consists of the multiplication of different inputs or raw data. So, there is a huge demand of multiplier in such kind of processing elements. Various high speed & power efficient multipliers are explained in the literature. Array multiplier (as shown in Fig. 2) is a basic multiplier which produces the partial products using an array of AND gates & then the ANDed products are added using summer/adder. But the main disadvantage of array multiplier is that it can add a maximum of three product terms at a time (in case of full adder) & therefore,

this architecture becomes bulkier with higher PDP when the total number addition levels are more. Wallace tree multiplier based on Wallace tree algorithm can solve the issue of bulkier structure of Array multiplier. By replacing the adder/summer part with Wallace tree algorithm, the multiplier can be made much more efficient. Here a multiplier is designed which generates the product of two numbers using purely combinational logic, i.e., in one gating step. Using straight forward diode-transistor logic, it appears presently possible to obtain products in under 1 micro sec, and quotients in 3 micro sec. A rapid square-root process is also outlined [18]. The Fig. 3 shows the same.



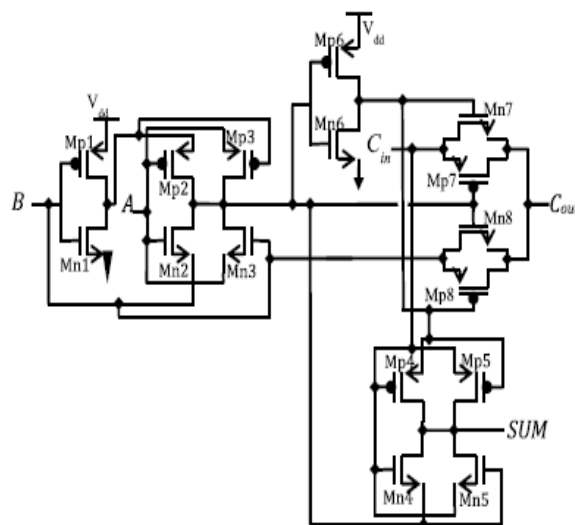
**Fig. 3, Wallace tree multiplier (addition of partial product)**

However, the problem with Wallace tree multiplier is that the addition of partial product is done in a single direction due to which the number of adder increases. This problem was sorted by a rectangular styled Wallace tree multiplier [3] in which the partial products are categorized into two groups and they are added in the opposite direction. The downward addition is done for the first group of the partial products & similarly upward addition is done for the second group of the partial products. On the other hand, in [12] a phase mode parallel multiplier [12] is proposed where it has a Wallace-tree structure which consists of trees of carry save adders for partial products addition. This structure has a regular layout & therefore, it is suitable for pipeline processing.

The conventional Wallace tree multiplier is basically consisting of carry save adder (CSA), which adds three variables at a time. Therefore, to enhance the speed of operation further, instead of CSA, compressor circuits can be used. In a similar approach, in [20], the speed of the multiplier is improved by using different compressors instead of the CSA. 3:2 compressor, 4:2 compressor, 5:2 compressors & 7:2 compressors are used rigorously to improve the speed of the existing Wallace tree multiplier. In the same study, it is summarized that the higher order

compressors (4:2, 5:2 or 7:2) performs better than 3:2 compressor. Therefore, the delay of the multiplier can be reduced by using higher order compressors.

As adder is a core unit in multiplier (and divider) circuit, the optimization on delay in multiplier can be further achieved by optimizing the adder circuit. Same kind of study is also seen in the literature, which optimized the adder circuit. In a novel approach, a Carry-Select-Adder (CSA) Optimization Technique [9] is proposed where a carry-select-adder (CSA) partitioning algorithm is used for Booth-encoded Wallace tree algorithm. By taking into various data arrival times, a branch-and-bound algorithm is proposed and a generalized technique to partition an n-bit carry-select adder into a number of adder blocks is proposed such that the overall delay of the design can be minimized. In a different approach, an algorithm [17] for implementing an efficient modulo  $(2n + 1)$  multipliers had been proposed. By manipulating the Booth tables and by applying a simple correction term, the proposed multiplier is the most efficient among all the known modulo  $(2n + 1)$  multipliers and is almost as efficient as those for ordinary integer multiplication. A comparative study in [16] is done for implementing multiplier using complementary MOS (CMOS), complementary pass-transistor (CPL) & double-pass transistor (DPL) logic style. A single precision reversible floating-point multiplier is proposed in [11]. A 24-bit multiplier is proposed in this work by decomposing the whole 24 bit in three portions of 8 bit each.



**Fig. 4, Full adder design in [21] which consumes very less power**

The internal to the multiplier is adder. Therefore, an optimized adder can further enhance the performability of a multiplier. An adder or summer circuit adds two or three variables. Adder is very common in logic circuits as it is used not only for summation but also to calculate the location addresses, increment/decrement operations, table indices etc. The common adders not only operate on binary number system but also for weighted & non-weighted codes. In the literature there are plenty number of full adders which are proposed to be efficient than others.

A novel low power MOSIS 90 nm technology-based hybrid full adder is proposed in [6].

The low power design is compared with conventional full adder consisting of 28 transistors. In a different approach, a hybrid 1-bit full adder is proposed which uses CMOS as well as TG logic styles [21]. The design is implemented in 90nm CMOS technology as well as 180nm CMOS technology. At 1.8V supply voltage, the proposed full adder, offers very little power and moderately low delay. The proposed adder in [21] is shown in Fig. 4.

### III. UCM ARCHITECTURE

The basic UCM architecture consists of three stages. The stage 1 & stage 3 remains the same for UCM architecture (as that of Wallace tree), because whether it is partial product generation or addition of intermediate sum or carry using fast adder, these can be chosen according to the requirement of the designer. Hence, it is more important to replace the stage 2 i.e. addition of partial product which creates sum & carry separately.

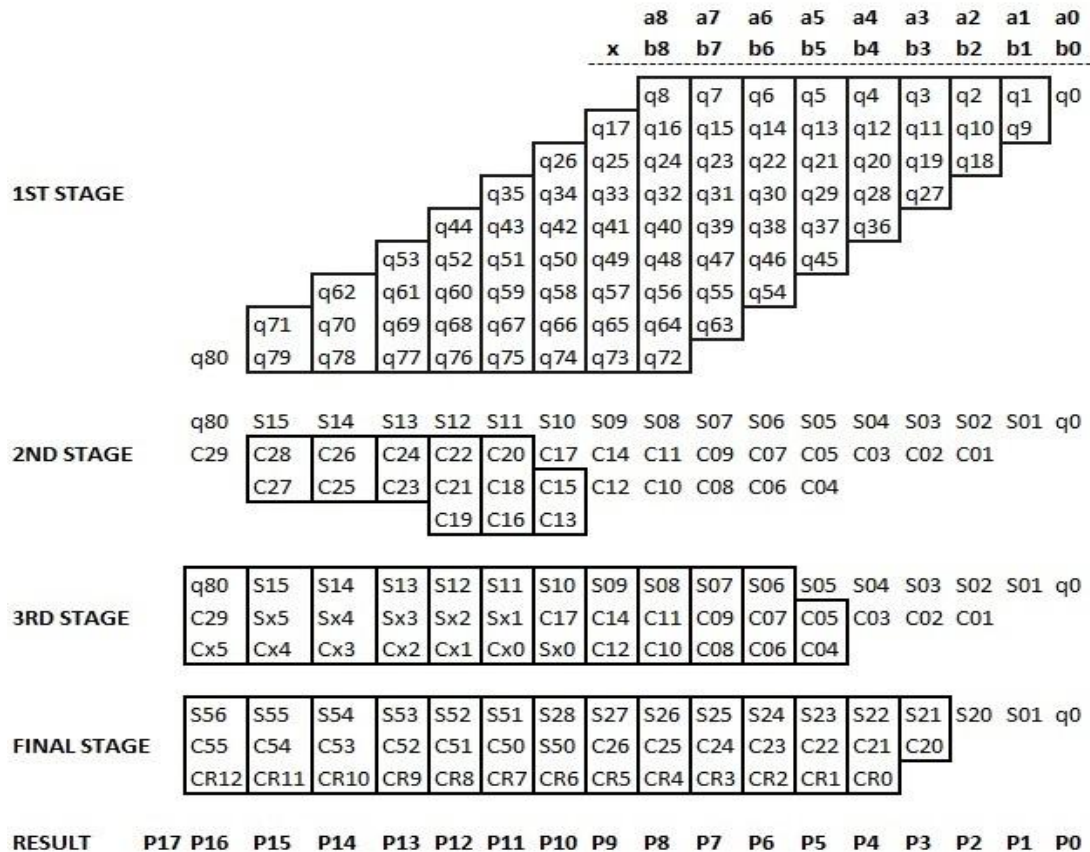


Fig. 5, UCM architecture for 9 x 9 bit multiplication

#### A. Addition of partial products

While adding partial products, the partial products are aligned in such a way that the summation of bit location of multiplicand & multiplier are equal. The summation of bit location can be called as 'weight' of a particular partial product. For example, in the Fig. 5, 'q35', 'q43', 'q51', 'q59', 'q67' & 'q75' are aligned in a single column because of the reason that the weight is eleven for all of the mentioned partial products, i.e.  $q35=a8b3$ ,  $q43=a7b4$ ,  $q51=a6b5$  etc. So, the summation of the bit location is either of  $8+3$  or  $7+4$  or  $6+5$ , which is in all cases are equal to 11. Hence, for the addition of partial products, its alignment is very important. Once the partial products are aligned the next step is to add all the partial product falling in that particular column. For adding a particular column firstly, the total number of stages & levels need to be identified. Each stage consists of an AND-XOR gate pair & the total number of stages in one level is counted from top to bottom. The total number of stages in the first level is 'i-1', where 'i' is the total number of partial products to be added in a particular column.

On the other hand, the horizontal count of AND-XOR pair is the total number of levels required for the design. In a different angle, we can say that the total number of levels required in a design is the total number of AND-XOR pair required in the bottom most stages. Basically, it is the count of AND-XOR pair from right to left. In each level, the total number of stages required will be decremented by one until it satisfies the formula:

$$2^n - 1 \geq i$$

$$\Rightarrow 2^n \geq i + 1$$

$$\Rightarrow n(\log_{10} 2) \geq \log_{10}(i + 1)$$

$$\therefore n \geq \frac{\log_{10}(i + 1)}{\log_{10} 2}$$

$$\text{or } n \geq \log_2(i + 1)$$



where 'i' is the total number of the partial product to be added & 'n' is the total number of levels required. 'i' & 'n' are integers starting from 1, 2, 3, .....,  $\infty$ . For example, for adding 3 partial products in a column, the total number of levels will be:  $2^n - 1 \geq 3$ , so  $n=2$ . Similarly, if suppose  $i=8$ , i.e.  $2^n - 1 \geq 8$ , so  $n=4$  & so on. The basic block diagram for K

stages & L levels is shown in Fig. 6. In Fig 6,  $A_0, A_1, A_2$  up to  $A_K$  are the partial products; the term  $Y_0$  is the sum &  $Y_1, Y_2, Y_3, \dots, Y_L$  are the carries. Therefore, in simple words, the algorithm shown in the Fig. 6 is a N-bit compressor circuit which generates sum of a particular column & single/multiple carries.

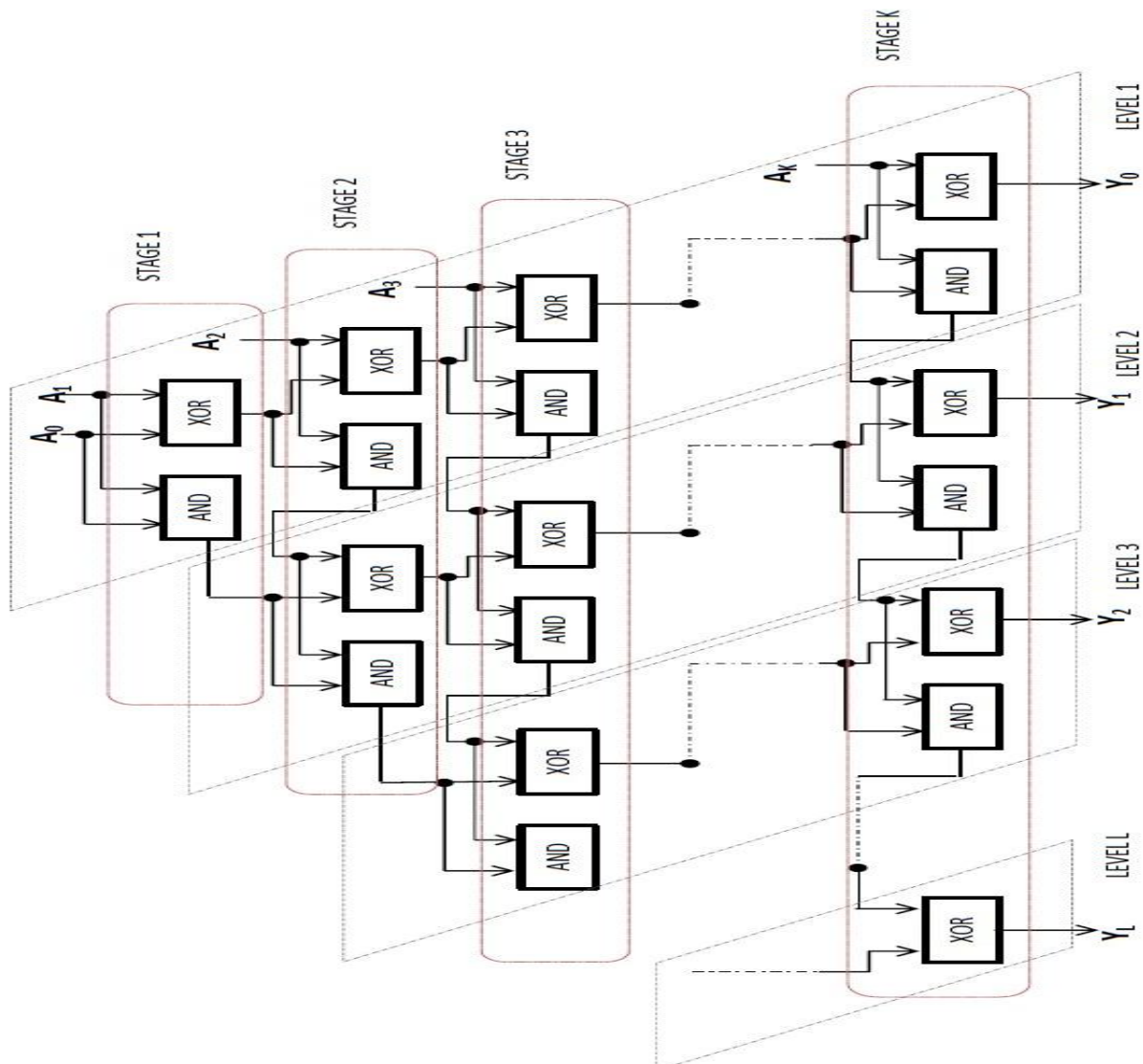


Fig. 6, AND-XOR gate arrangement with K stages & L levels having  $A_0, A_1, A_2, \dots, A_K$  partial products (with equal weights) for a particular column

### B.Special cases

- In the last level, instead of AND-XOR pair, only XOR gate is to be used.
- If  $i=2$ , only one level is to be used to get the sum as well as carry. In this case, the output from the AND is the carry.
- For  $i=1$ , the input itself is the output (sum) & there is no carry output.

It is very important to note that the output through the level 1 is the sum of the partial products present in a particular column & the outputs of rest of the levels i.e. level 2 to level L are the corresponding carry bits. After getting the sum as well as carry bit of all columns, the next step is to add up the sum bits with the carry bit of the previous columns. For this any of the efficient algorithms such as data algorithm, Wallace tree algorithm or even ripple carry adder

can be used as the number of rows has reduced substantially. A detailed design is shown in Fig. 5.

### IV. PVT ANALYSIS & RESULTS

VLSI is an art of chip design, where specification is transformed to functional hardware. Cadence provides tools for front end as well as back end designs, where, after rigorous design steps, GDS-II file are finally sent for fabrication. But due to process complexity (i.e. pressure, supply voltage, temperature etc.) the YIELD of the fabricated designs is found to be very low. Major reason for yield loss is fabrication parameter variation among wafer to wafer. To improve the yield of design; the IC should be able to sustain extreme variation.

Therefore, validation of the design cycle through PVT and 3-sigma variation becomes essential before fabrication.

The work published in [1], provides a comparison of delays for 5x5 bit as well as for 9x9 bit operation for 0.6V, 0.7V, 0.8V & 0.9V. The same has been shown in the Fig. 7 & Fig. 8.

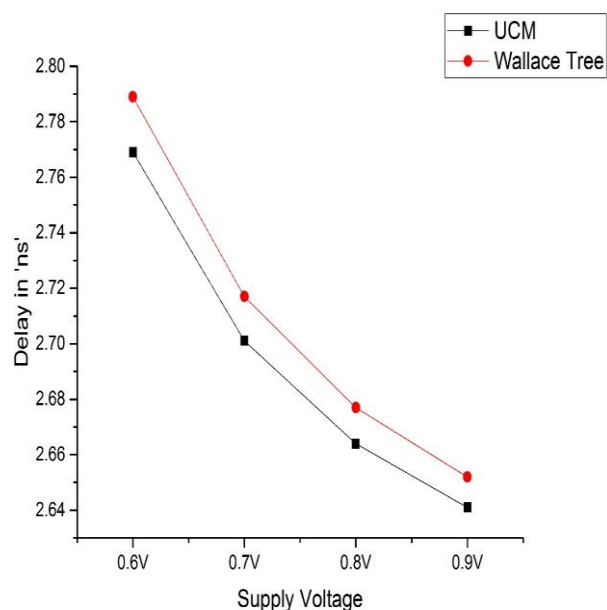


Fig. 7, Graphical comparison of 5x5 bit UCM & 5x5 bit Wallace tree multiplier at voltages below 1V

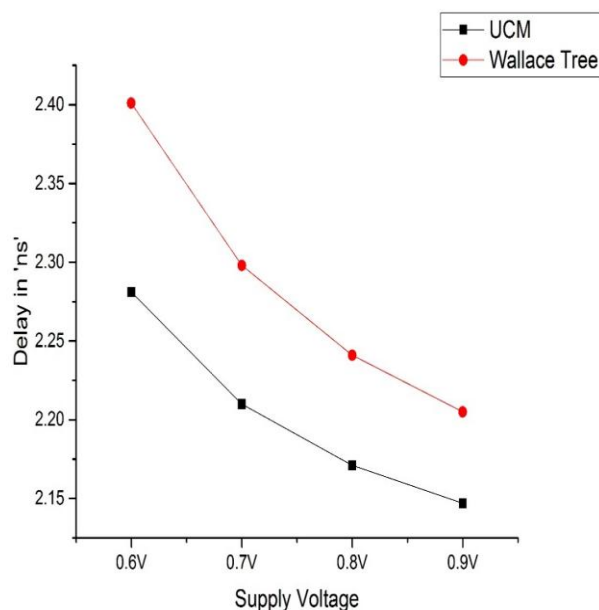


Fig. 8, Graphical comparison of 9x9 bit UCM & 9x9 bit Wallace tree multiplier at voltages below 1V

The comparison shows that the UCM [1] architecture performs better than Wallace tree architecture at ultra-low supply voltages (less than 0.9V). Moreover, the UCM architecture performs even better for higher order bit multiplication. For example, the difference in delay of UCM & Wallace tree architecture for 9x9 bit operation is more than 5x5 bit operation (120 ps & 20 ps respectively). Therefore, the author [1] summarized that UCM architecture performs

better than Wallace tree for higher order bit multiplication at ultra-low supply voltages (less than 0.9V).

To validate the performance of the UCM [1] architecture further, a PVT analysis is carried out at different corners (Fast-Fast, Fast-Slow, Normal-Normal, Slow-Fast & Slow-Slow) & at three different extreme temperatures ( $-40^{\circ}$ ,  $0^{\circ}$  &  $+50^{\circ}$  Celsius). Table I & II shows the delay comparison of UCM & Wallace tree 5x5 bit & 9x9 bit architecture respectively at 0.6 V & 0.9 V supply voltage in different corners along with variation in temperature ( $-40^{\circ}$ ,  $0^{\circ}$  &  $+50^{\circ}$  Celsius)

Table I, Delay comparison of UCM & Wallace tree 5x5 bit architecture at 0.6 V & 0.9 V supply voltage in different corners along with variation in temperature ( $-40^{\circ}$ ,  $0^{\circ}$  &  $+50^{\circ}$  Celsius)

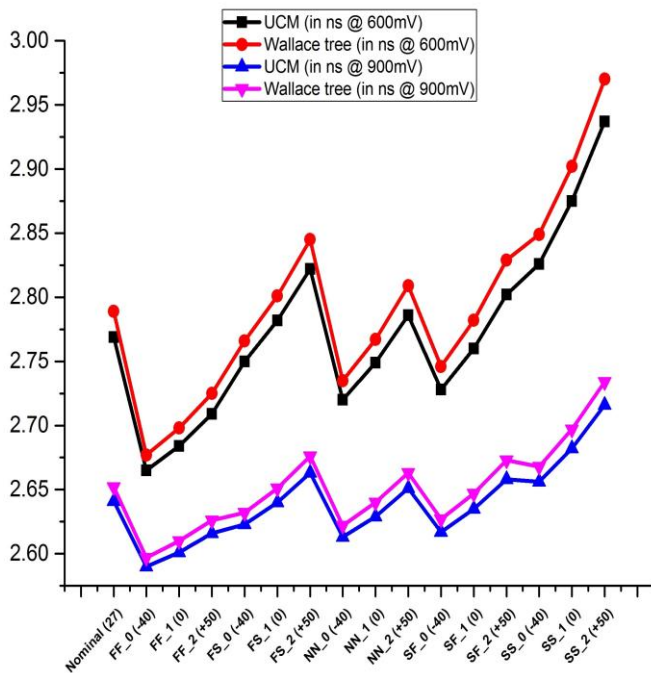
	UCM (in ns @ 600mV)	Wallace tree (in ns @ 600mV)	UCM (in ns @ 900mV)	Wallace tree (in ns @ 900mV)
Nominal (27)	2.769	2.789	2.641	2.652
FF_0 (-40)	2.665	2.677	2.59	2.597
FF_1 (0)	2.684	2.698	2.601	2.61
FF_2 (+50)	2.709	2.725	2.616	2.626
FS_0 (-40)	2.75	2.766	2.623	2.632
FS_1 (0)	2.782	2.801	2.64	2.651
FS_2 (+50)	2.822	2.845	2.663	2.676
NN_0 (-40)	2.72	2.735	2.613	2.622
NN_1 (0)	2.749	2.767	2.629	2.64
NN_2 (+50)	2.786	2.809	2.651	2.663
SF_0 (-40)	2.728	2.746	2.617	2.627
SF_1 (0)	2.76	2.782	2.635	2.647
SF_2 (+50)	2.802	2.829	2.658	2.673
SS_0 (-40)	2.826	2.849	2.656	2.668
SS_1 (0)	2.875	2.902	2.682	2.697
SS_2 (+50)	2.937	2.97	2.716	2.734

Table II, Delay comparison of UCM & Wallace tree 9x9 bit architecture at 0.6 V & 0.9 V supply voltage in different corners along with variation in temperature ( $-40^{\circ}$ ,  $0^{\circ}$  &  $+50^{\circ}$  Celsius)

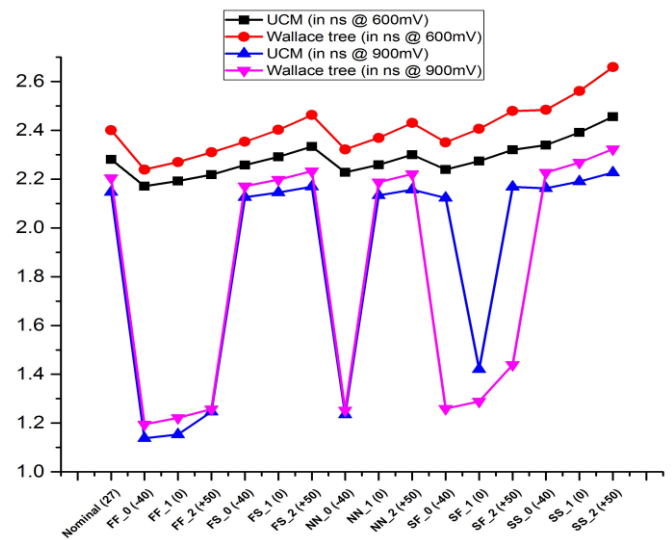
	UCM (in ns @ 600mV)	Wallace tree (in ns @ 600mV)	UCM (in ns @ 900mV)	Wallace tree (in ns @ 900mV)
Nominal (27)	2.281	2.401	2.147	2.205
FF_0 (-40)	2.171	2.239	1.138	1.195
FF_1 (0)	2.192	2.27	1.153	1.222
FF_2 (+50)	2.218	2.31	1.247	1.257
FS_0 (-40)	2.258	2.353	2.126	2.171

FS_1 (0)	2.291	2.402	2.145	2.198
FS_2 (+50)	2.334	2.463	2.169	2.233
NN_0 (-40)	2.228	2.322	1.235	1.252
NN_1 (0)	2.259	2.369	2.134	2.187
NN_2 (+50)	2.3	2.43	2.157	2.221
SF_0 (-40)	2.239	2.351	2.123	1.259
SF_1 (0)	2.274	2.406	1.421	1.289
SF_2 (+50)	2.32	2.479	2.168	1.439
SS_0 (-40)	2.339	2.484	2.162	2.227
SS_1 (0)	2.391	2.561	2.19	2.268
SS_2 (+50)	2.456	2.659	2.227	2.323

Moreover, a graphical comparison of delay of UCM & Wallace tree 5x5 bit & 9x9 bit architectures at 0.6 V & 0.9 V supply voltage in different corners along with variation in temperature (-40°, 0° & +50° Celsius) are shown in Fig. 9 & Fig. 10. The graphs in Fig. 9 & Fig. 10 clearly shows that there is a significant improvement in delay of UCM architecture in comparison to the Wallace tree architecture for 5x5 bit as well as 9x9 bit multiplication. Most important part is that, for 5x5 bit multiplication, at different corners & at extreme temperatures, the UCM architecture proves to be the better performer than Wallace tree architecture at ultra-low supply voltages.



**Fig. 9, Graphical comparison of delay of UCM & Wallace tree 5x5 bit architecture at 0.6 V & 0.9 V supply voltage in different corners along with variation in temperature (-40°, 0° & +50° Celsius)**



**Fig. 10, Graphical comparison of delay of UCM & Wallace tree 9x9 bit architecture at 0.6 V & 0.9 V supply voltage in different corners along with variation in temperature (-40°, 0° & +50° Celsius)**

On the other hand, for 9x9 bit multiplication, the delay of UCM has a much more significant drop in comparison to the Wallace tree at 600 mV (at different corners & at extreme temperatures). Whereas, the delay of the UCM architecture is seems to be slightly higher than Wallace tree at slow-fast (SF) corner in -40°, 0° & +50° Celsius for 9x9 bit multiplication at 900 mV. The reason for the same might be the use different process at SF corner. Moreover, as shown in the Table I, the minimum & maximum delay for 5x5 bit multiplication using UCM architecture at 600 mV are 2.665 ns & 2.937 ns respectively. Whereas the same for Wallace tree are 2.677 ns & 2.97 ns respectively. Similarly, the minimum & maximum delay for 5x5 bit multiplication using UCM architecture at 900 mV are 2.59 ns & 2.716 ns respectively. Whereas the same for Wallace tree are 2.597 ns & 2.734 ns respectively. Same thing if we observe for 9x9 bit multiplication using UCM architecture at 600 mV, the minimum & maximum delays are 2.171 ns & 2.456 ns respectively whereas for Wallace tree the values are 2.239 ns & 2.659 ns. On the other hand, for 9x9 bit multiplication using UCM architecture at 900 mV, the minimum & maximum delays are 1.138 ns & 2.227 ns respectively whereas for Wallace tree the values are 1.195 ns & 2.323 ns.

## V. CONCLUSION

The UCM architecture has a wide range of acceptability in the field of digital system design. UCM architecture not only performs the best in a nominal Process, Voltage & Temperature but also in a wide range of variation in extreme temperature, process & ultra-low supply voltages. Especially, in the case of the higher order multiplication (9x9 bit) operation with supply voltage as low as 0.6 V, the delay has reduced by 5.05% (mean value) than Wallace tree multiplier architecture. Therefore, UCM multiplier will have a wide range of acceptability in the circuits where speed is the top most priority.



## REFERENCES

1. R. Sarma, C. Bhargava, S. Dhariwal, and S. Jain, "UCM: A novel approach for delay optimization," *International Journal of performance Engineering*, vol. 15, no. 4, pp. 1190-1198, 2019.
2. D. Guevorkian, A. Launainen, V. Lappalainen, P. Liuha, and K. Punkka, "A Method for Designing High-Radix Multiplier-Based Processing Units for Multimedia Applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 5, pp. 716-725, 2005.
3. N. Itoh, Y. Naemura, H. Makino, Y. Nakase, T. Yoshihara, and Y. Horiba, "A 600-MHz 54 54-bit Multiplier with Rectangular-Styled Wallace Tree," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 2, pp. 249-257, 2001.
4. K. B. Jaiswal, N. Kumar, P. Seshadri, and L. G. "Low Power Wallace Tree Multiplier Using Modified Full Adder," in *Proceedings of the 3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*, 2015.
5. I. Kataeva, H. Engseth, and A. Kidiyarova-Shevchenko, "Scalable Matrix Multiplication With Hybrid CMOS-RSFQ Digital Signal Processor," *IEEE Transactions on Applied Superconductivity*, vol. 17, no. 2, pp. 486-489, 2007.
6. S. Khan, S. Kakde, and Y. Suryawanshi, "VLSI Implementation of Reduced Complexity Wallace Multiplier Using Energy Efficient CMOS Full Adder," in *Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research*, 2013.
7. R. D. Kshirsagar, E. V. Aishwarya, A. S. Vishwanath, and P. Jayakrishnan, "Implementation of Pipelined Booth Encoded Wallace Tree Multiplier Architecture," in *Proceedings of the International Conference on Communication and Green Computing Conservation of Energy (ICGCE)*, Chennai, 2013.
8. T. Y. Kuo and J. S. Wang, "A Low-Voltage Latch-Adder Based Tree Multiplier," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Seattle, WA, 2008.
9. M. Liao, C. Su, C. Chang, and A. C. Wu, "A Carry-Select-Adder Optimization Technique for High-Performance Booth-Encoded Wallace-Tree Multipliers," *IEEE International Symposium on Circuits and Systems*, ISCAS 2002, 2002.
10. X. V. Luu, T. T. Hoang, T. T. Bui, and A. V. Dinh-Duc, "A High-speed Unsigned 32-bit Multiplier Based on Booth encoder and Wallace-tree Modifications," in *Proceedings of the International Conference on Advanced Technologies for Communications (ATC'14)*, 2014.
11. M. Nachtigal, H. Thapliyal, and N. Ranganathan, "Design of a Reversible Single Precision Floating Point Multiplier Based on Operand Decomposition," in *Proceedings of the 10th IEEE conference on Nanotechnology*, Kintex, Korea, 2010.
12. T. Onomi, K. Yanagisawa, M. Seki, and K. Nakajima, "Phase-Mode Pipelined Parallel Multiplier," *IEEE Transactions on Applied Superconductivity*, vol. 11, no. 1, pp. 541-544, 2001.
13. C. Paradhasaradhi, M. Prashanthi, and N. Vivek, "Modified Wallace Tree Multiplier using Efficient Square-Root Carry Select Adder," in *Proceedings of the International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE)*, Coimbatore, 2014.
14. M. J. Rao and S. Dubey, "A High Speed and Area Efficient Booth Recoded Wallace Tree Multiplier for fast Arithmetic Circuits," in *Proceedings of the Asia Pacific Conference on Postgraduate Research in Microelectronics & Electronics (PRIMEASIA)*, BITS Pilani, Hyderabad, 2012.
15. B. M. Reddy, H. N. Sheshagiri, B. R. Vijaykumar, and S. S., "Implementation of Low Power 8-Bit Multiplier using Gate Diffusion Input Logic," in *Proceedings of the 17th IEEE International Conference on Computational Science and Engineering*, 2014.
16. A. K. Singh, B. P. De, and S. Maity, "Design and Comparison of Multipliers Using Different Logic Styles," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 2, no. 2, pp. 374-379, 2012.
17. L. Sousa, "Algorithm for modulo  $(2^n+1)$  multiplication," *Electronics Letters*, pp. 752-754, 01 May 2003.
18. C. S. Wallace, "A Suggestion for a Fast Multiplier," *IEEE Transactions on Electronic Computers*, pp. 14-17, 1964.
19. Q. Yi and H. Jing, "An Improved Design Method for Multi-bits Reused Booth Multiplier," in *Proceedings of the 4th International Conference on Computer Science & Education*, 2009.
20. K. Gopi Krishna, B. Santhosh, and V. Sridhar, "Design of Wallace Tree Multiplier using Compressors," *International Journal of Engineering Sciences & Research Technology*, vol. 2, no. 9, pp. 2249-2254, 2013.
21. P. Bhattacharyya, B. Kundu, S. Ghosh, V. Kumar, and A. Dandapat, "Performance Analysis of a Low-Power High-Speed Hybrid 1-bit Full Adder Circuit," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 1-8, 2014.

## AUTHORS PROFILE



Mr. Rajkumar Sarma received his B.E. in Electronics and Communications Engineering from Vinayaka Mission's University, Salem, India & M.Tech degree from Lovely Professional University, Phagwara, Punjab and currently pursuing PhD from Lovely Professional University, Phagwara, Punjab. He is an Assistant Professor in the School of Electronics and Electrical Engineering, Lovely Professional University, Punjab since July 2012. His research interests include Analog and Digital VLSI design, Prototype development using FPGA etc. The author has around 15 research publication.



Dr. Cherry Bhargava is working as an assistant professor and head, VLSI domain, School of Electrical and Electronics Engineering at Lovely Professional University, Punjab, India. She has more than 14 years of teaching and research experience. She is PhD (ECE), IKGPTU, M.Tech (VLSI Design & CAD) Thapar University and B.Tech (Electronics & Instrumentation) from Kurukshetra University. She is GATE qualified with All India Rank 428. She has authored about 50 technical research papers in SCI, Scopus indexed quality journals and national/international conferences. She has four e-books to her credit. She has registered two copyrights and filed one patent. She is recipient of various national and international awards for being outstanding faculty in engineering and excellent researcher. She is an active reviewer and editorial member of various prominent SCI and Scopus indexed journals. She is a lifetime member of IET, IAENG, NSPE, IAOP, WASET and reliability research group. Her area of expertise includes reliability of electronic systems, digital electronics, VLSI design, artificial intelligence and related technologies.



Dr. Shruti Jain received her doctoral degree from Jaypee University of Information Technology, Waknaghat, Solan. She has a teaching experience of around 13 years. She has specialization in Biomedical Signal Processing, Computer- Aided design of FPGA and VLSI circuits, combinatorial optimization. She has published more than 50 papers in reputed journals and 30 papers in International conferences.