Chapter 13
PVT Variability Check on UCM Architectures at Extreme Temperature–Process Changes

Rajkumar Sarma
https://orcid.org/0000-0002-5551-1006
Lovely Professional University, India

Cherry Bhargava
https://orcid.org/0000-0003-0847-4780
Lovely Professional University, India

Shruti Jain
JUIT, India

ABSTRACT

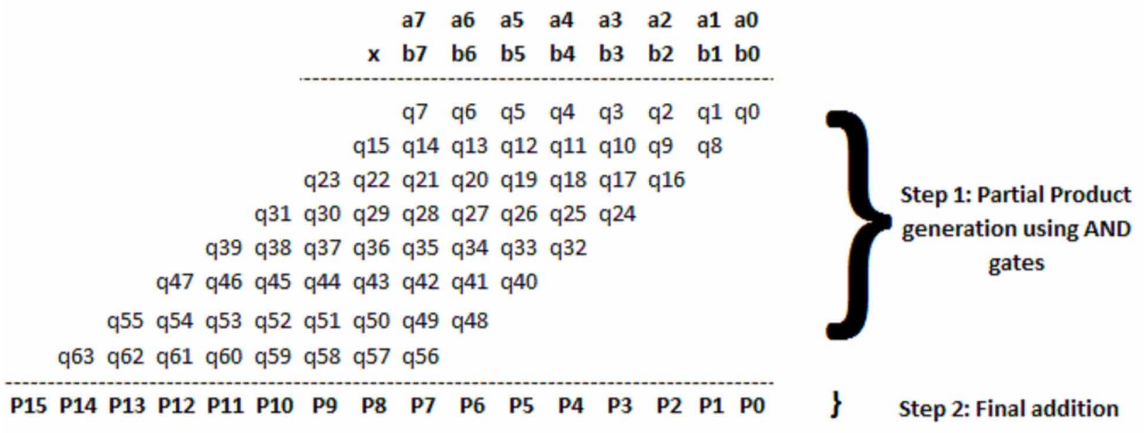
The UCM (universal compressor-based multiplier) architecture promises to provide faster multiplication operation in supply voltage as low as 0.6 V. The basic component of UCM architecture is a universal compressor architecture that replaces the conventional Wallace tree algorithm. To extend the work further, in this chapter, a detailed PVT (process-voltage-temperature) analysis is performed using Cadence Virtuoso 90nm technology. The analysis shows that the delay of the UCM has reduced more significantly than the Wallace tree algorithm at extreme process, voltage, and temperature.

INTRODUCTION

Today's portable devices are capable of doing image filtering to face recognitions, an audio signal enhancement to voice recognition & gesture-based control to biometric authentication. All those functionalities are the applications of digital signal processing (DSP). A large number of mathematical operations are performed repeatedly and quickly on series of data samples by DSP algorithms. Most operating systems and general-purpose microprocessors can successfully execute DSP algorithms but

DOI: 10.4018/978-1-7998-1464-1.ch013

Figure 1. Basic multiplication operation



because of power efficiency constraints, they are not suitable for use in portable devices such as PDAs and mobile phones. However, the rapid growth of portable electronics has introduced the major challenges of high throughput for VLSI design engineers. Among the other digital blocks, multiplier plays a vital role while evaluating the performance of a DSP block. While performing convolution, filtering or any other DSP operations it is always desired to use an efficient multiplier unit. A basic design of a multiplier is as shown in the figure 1.

As shown in the figure 1, the multiplicand's & the multiplier's individual terms are ANDed to produce the partial products & positioned as per their weights. For example, in the figure 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 will be 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 full adder (in the case of 3 partial products), half adder (in the case of 2 partial products) or any compressor circuit (for adding 'n' number of partial products simultaneously).

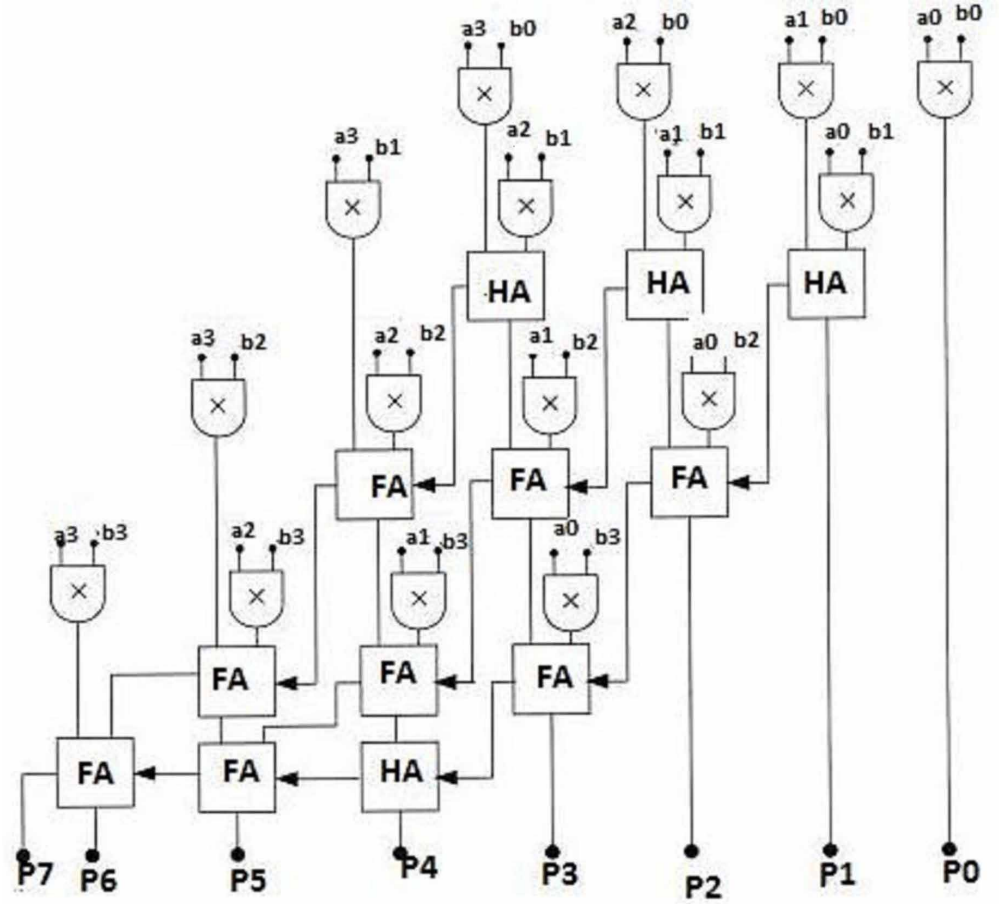
In this research paper the novel UCM architecture as proposed in (Sarma, Bhargava & Jain, 2019), is further validated with the PVT analysis in cadence spectre tool in 90 nm CMOS technology. The UCM architecture uses a novel compressor-based multiplier algorithm which reduces the delay substantially.

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, a detailed conclusion, future scopes & application of the UCM architecture is discussed.

VARIOUS MULTIPLIER ARCHITECTURES

As we know that the processing elements mainly involve the multiplication of two numbers. So, there is a need of multiplier in such type of processing systems. Various fast & efficient multipliers are described in the literature. Array multiplier (as shown in figure 2) is a basic multiplier which follows the principle of product generation & addition. But as the total number of addition levels increases, this architecture becomes bulkier with higher PDP.

Figure 2. Array multiplier architecture

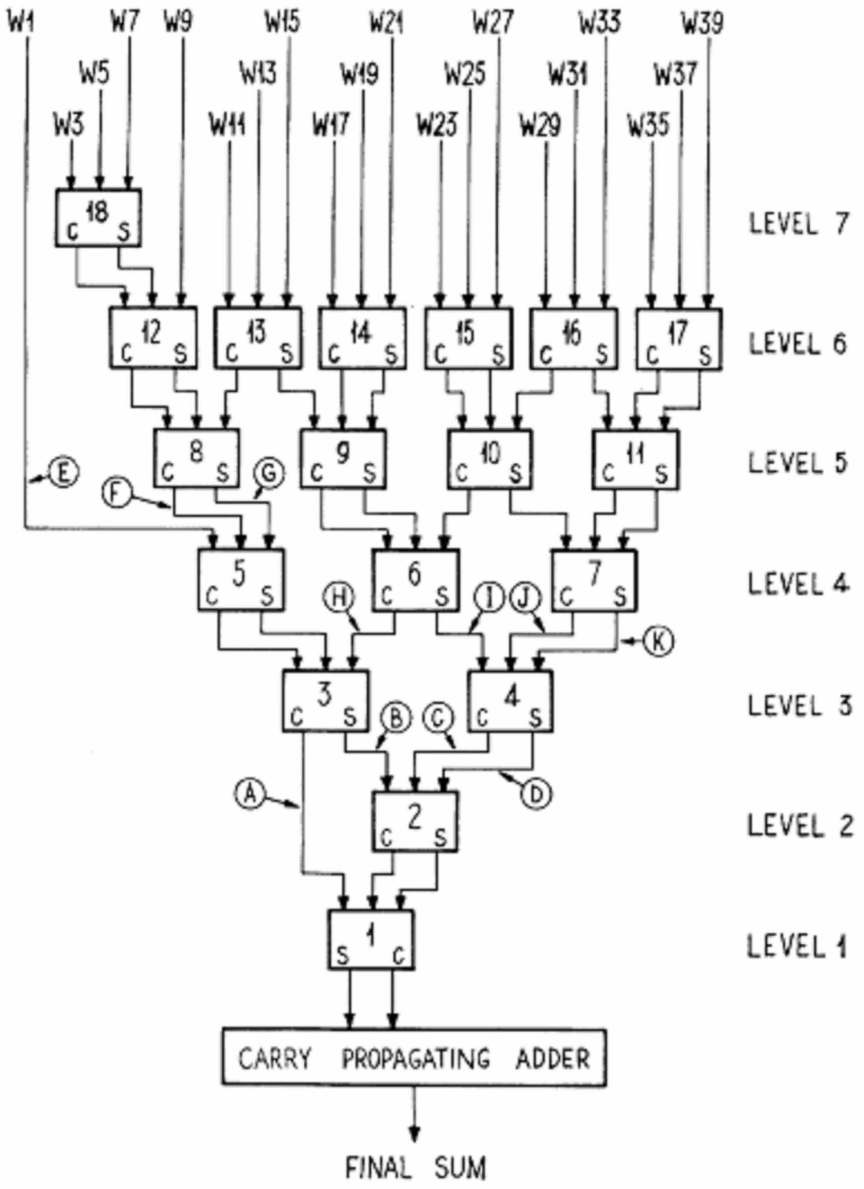


Solution for this problem can be Wallace tree multiplier based on Wallace tree structure. Here a multiplier is designed which generates the product of two numbers using purely combinational logic, i.e., in one gating step. Using straight forward diode-transistor logic, it appears presently possible to obtain products in under 1 micro sec, and quotients in 3 micro sec. A rapid square-root process is also outlined in the literature (Wallace, 1964). The figure for the same is shown in Figure 3.

However, in Wallace tree multiplier every partial product is added in a single direction from top to bottom so the number of adder increases. To overcome this problem a rectangular styled Wallace tree multiplier is proposed (Toh et al., 2001), in which the partial products are divided into two groups and added in the opposite direction. The partial products in the first group are added downward, and the partial products in the second group are added upward. On the other hand, in the literature (Onomi et al., 2001), a phase mode parallel multiplier is also proposed. The proposed multiplier has a Wallace-tree structure comprising trees of carry save adders for the addition of partial products. This structure has a regular layout; hence it is suitable for a pipeline scheme.

The conventional Wallace tree multiplier is based on carry save adder. In (Guevorkian et al., 2013) the speed of the multiplier has improved by introducing compressors instead of the carry save adder. 3-2 compressor, 4-2 compressor, 5-2 compressors and 7-2 compressors are used with Wallace tree multiplier.

Figure 3. Wallace tree multiplier (addition of partial product)



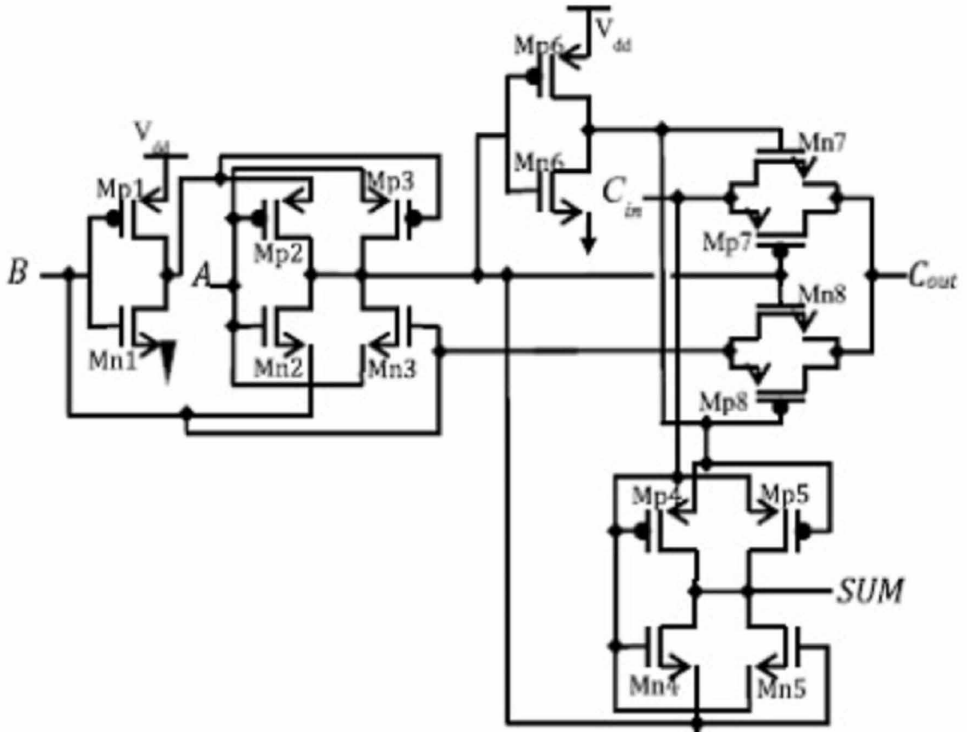
Higher order compressors have better performance compared with 3-2 compressor. So, the speed of the multiplier can be improved by introducing the higher order compressors.

A few architectures in the literature focused on the optimization of adder cell also. As adder is a basic cell in multiplier or divider, the optimization is mainly focused on adder part. A Carry-Select-Adder Optimization Technique is proposed in the literature (Liao et al., 2002) in which a carry-select-adder partitioning algorithm is for high-performance Booth-encoded Wallace-tree multipliers. 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 separate approach (Sousa, 2003), an improved algorithm for designing efficient modulo ($2^n + 1$) 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 ($2^n + 1$) multipliers and is almost as efficient as those for ordinary integer multiplication. On the other hand, (Singh, De & Maity, 2012) a comparative analysis is carried out for designing multiplier using complementary MOS (CMOS) logic style, complementary pass-transistor (CPL) logic style and double-pass transistor (DPL) logic style. Similarly a single precision reversible floating-point multiplier is proposed in the literature (Nachtigal, Thapliyal & Ranganathan, 2010). A 24-bit multiplier is proposed in this work by decomposing the whole 24-bit in three portions of 8 bit each.

The internal to the multiplier is adder. Therefore, an optimized adder can further enhance the performability of a multiplier. An adder (also referred to as summer) is a logic circuit which adds two or more variables. Adders are unavoidable part of logic circuits as they are not only used for addition but also to calculate the addresses, increment operations, table indices etc. Most common adders operate on binary numbers although they can be constructed for BCD, excess -3 formats etc. In the literature there are various full adder architectures are proposed. A novel low power hybrid full adder using MOSIS 90nm Technology (Khan, Kakde, & Suryawanshi, 2013) is proposed, which consumes very low power. The proposed design is compared with its conventional full adder which consists of 28 transistors. In a different approach, a hybrid 1-bit full adder is proposed by (Bhattacharyya et al., 2014) which uses CMOS as well as TG logic styles. The entire design was implemented in 90nm as well as 180nm technologies. At 1.8V supply voltage, the proposed design, offers very less power and moderately low delay. The adder proposed by (Bhattacharyya et al., 2014) is shown in figure 4.

Figure 4. Full adder design by (Bhattacharyya et al., 2014)



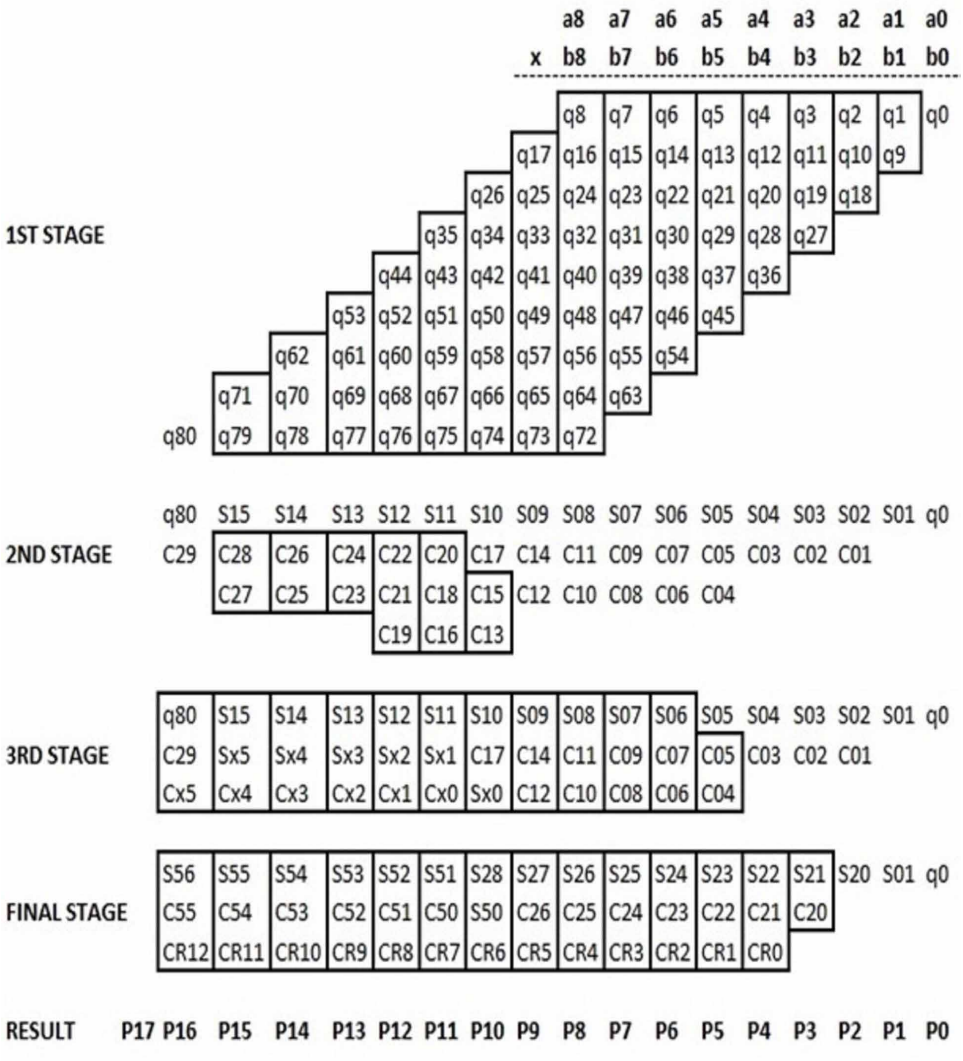
UCM ARCHITECTURE

The 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.

Addition of Partial Products

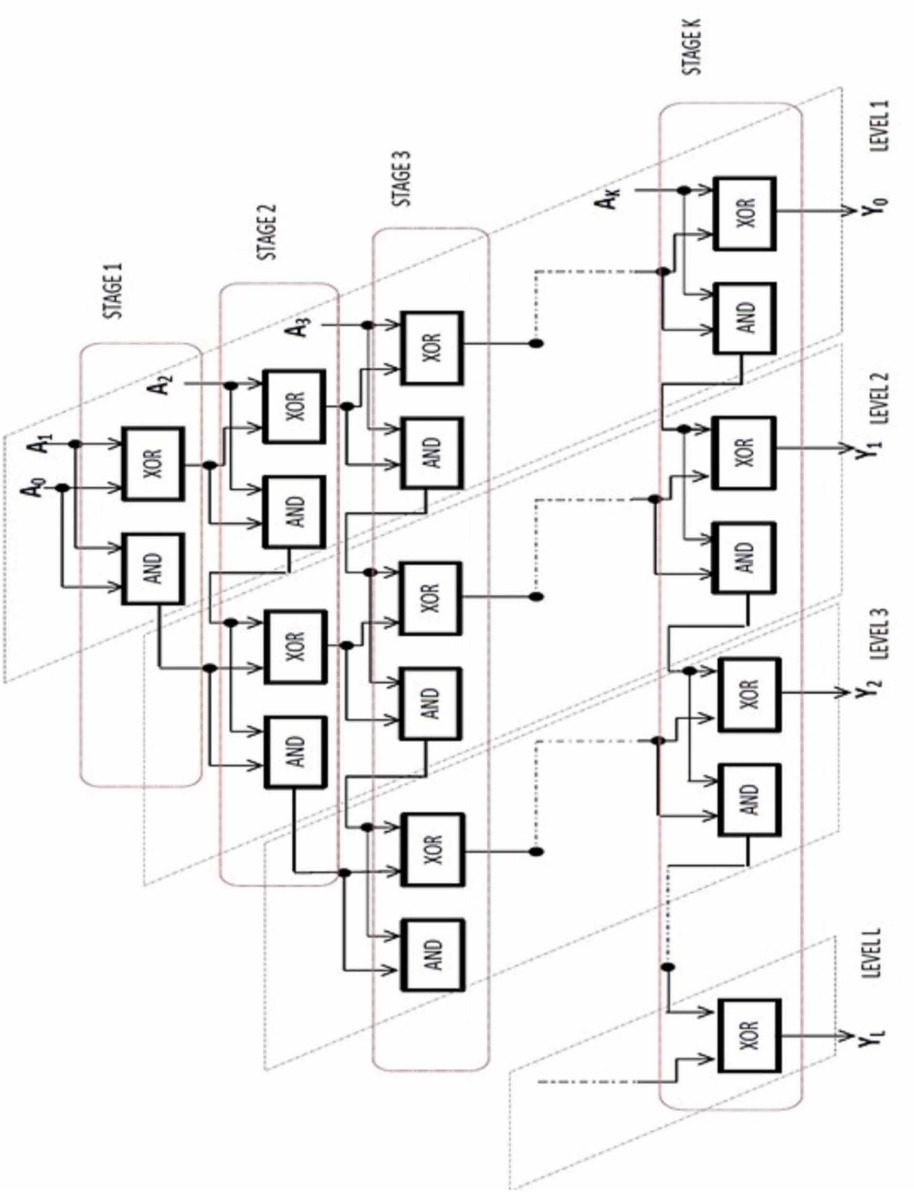
While adding partial products, the partial products are aligned in such a way that the summation of bit location of multiplicand & multiplier are equal. The summation of bit location can be called as `weight'

Figure 5. UCM architecture for 9 x 9 bit multiplication



of a particular partial product. For example, in the figure 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

Figure 6. AND-XOR gate arrangement with K stages & L levels having A0, A1, A2,....., AK partial products (with equal weights) for a particular column



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:

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, ..., ∞. For example, for adding 3 partial products in a column, the total number of levels will be: $2n\text{-}1 \geq 3$, so n=2. Similarly, if suppose i=8, i.e. $2n\text{-}1 \geq 8$, so n=4 & so on. The basic block diagram for K stages & L levels is shown in figure 5. In figure 6, A0, A1, A2 up to AK are the partial products; the term Y0 is the sum & Y1, Y2, Y3,....., YL are the carries. Therefore, in simple words, the algorithm shown in the figure 6 is a N-bit compressor circuit which generates sum of a particular column & single/multiple carries.

Special Cases

1. In the last level, instead of AND-XOR pair, only XOR gate is to be used.
2. 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.
3. 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 dada 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 figure 5.

PVT ANALYSIS

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 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, the design cycle must be validated through PVT and 3-sigma variation before fabrication.

The work proposed by (Sarma, Bhargava & Jain, 2019), 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 figure 7 & 8.

The comparison shows that the UCM architecture, as proposed by (Sarma, Bhargava & Jain, 2019), 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 differ-

ence in delay of UCM & Wallace tree architecture for 9x9 bit operation is more than 5x5 bit operation (120 ps & 20 ps respectively). Therefore, (Sarma, Bhargava & Jain, 2019) has 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 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°, 0° & +50° Celsius). Table 1 & 2 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°, 0° & +50° Celsius)

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 figure 9 & 10. The graphs in figure 9 & 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. 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 corner in -40°,0° & +50° Celsius for 9x9 bit multiplication at 900 mV. Moreover, as shown in the table 1, 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

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

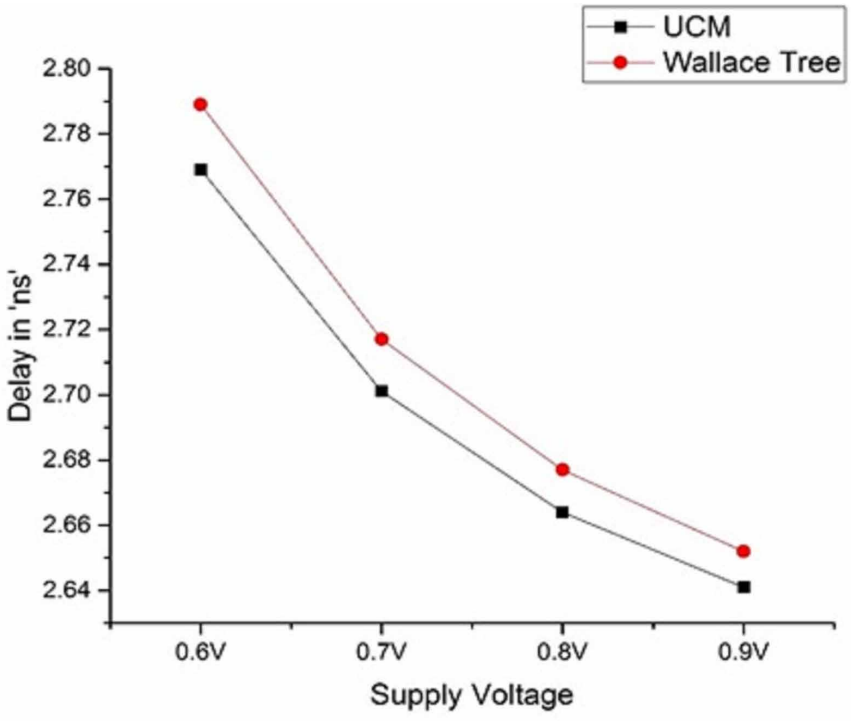


Figure 8. Graphical comparison of 9x9 bit UCM & 9x9 bit Wallace tree multiplier at voltages below 1V

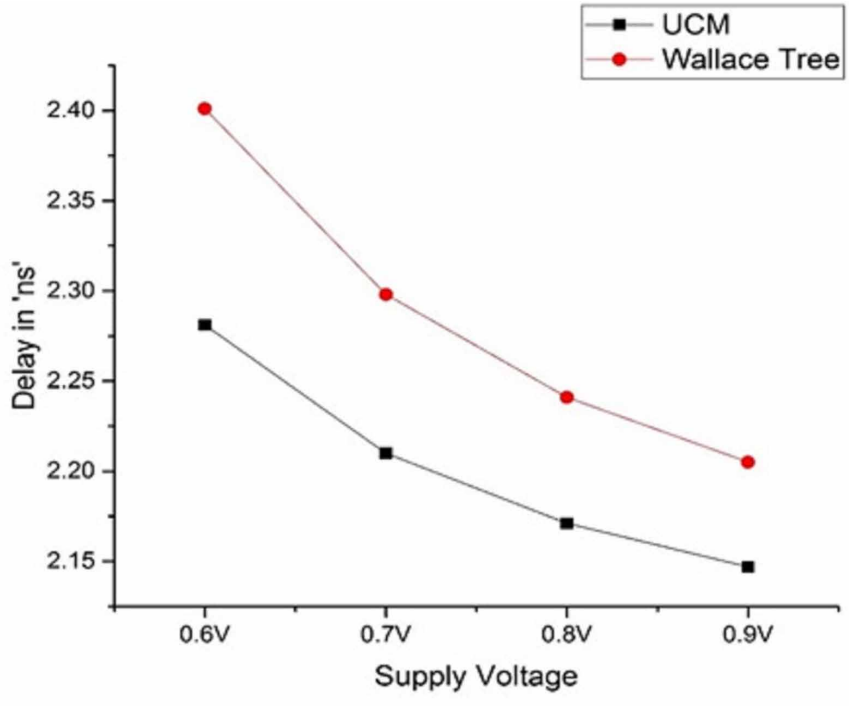


Table 1. Delay comparison of UCM & Wallace tree 5x5 bit architecture at 0.6 V & 0.9 V supply voltage in different corners along with variation in temperature (-40°,0° & +50° Celsius)

	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 2. 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°,0° & +50° 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

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.

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.

Figure 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)

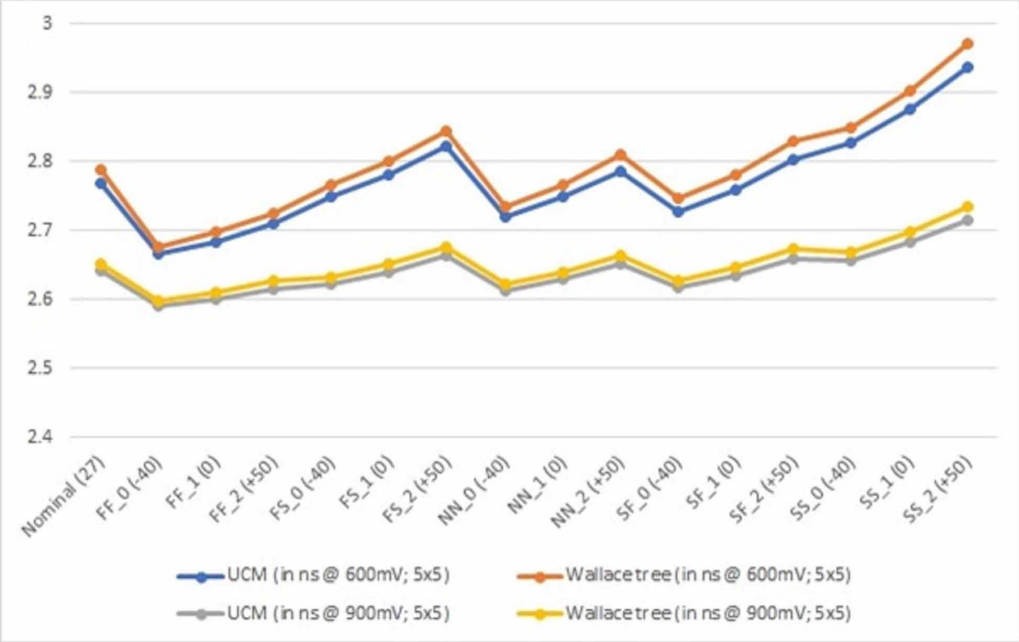
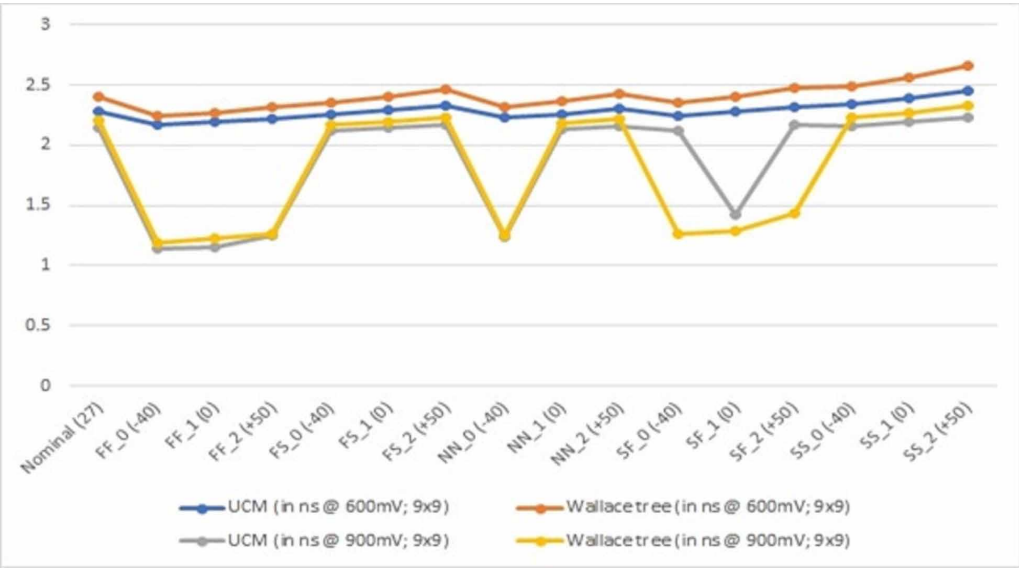


Figure 10. Graphical comparison of delay of UCM & Wallace tree 9x9 bit architecture at 0.6 V & 0.9 V supply voltage in different corners along with variation in temperature (-40°,0° & +50° Celsius)



REFERENCES

Bhattacharyya, P., Kundu, B., Ghosh, S., Kumar, V., & Dandapat, A. (2014). Performance analysis of a low-power high-speed hybrid 1-bit full adder circuit. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 23*(10), 2001-2008.

Guevorkian, D., Launiainen, A., Lappalainen, V., Liuha, P., & Punkka, K. (2005). A method for designing high-radix multiplier-based processing units for multimedia applications. *IEEE Transactions on Circuits and Systems for Video Technology*, *15*(5), 716–725. doi:10.1109/TCSVT.2005.846436

Jaiswal, K. B., Seshadri, P., & Lakshminarayanan, G. (2015, March). Low power wallace tree multiplier using modified full adder. In *2015 3rd international conference on signal processing, communication and networking (ICSCN)* (pp. 1-4). IEEE. 10.1109/ICSCN.2015.7219880

Kataeva, I., Engseth, H., & Kidiyarova-Shevchenko, A. (2007). Scalable matrix multiplication with hybrid CMOS-RSFQ digital signal processor. *IEEE Transactions on Applied Superconductivity*, *17*(2), 486–489. doi:10.1109/TASC.2007.901451

Khan, S., Kakde, S., & Suryawanshi, Y. (2013, December). VLSI implementation of reduced complexity wallace multiplier using energy efficient CMOS full adder. In *2013 IEEE International Conference on Computational Intelligence and Computing Research* (pp. 1-4). IEEE. 10.1109/ICCIC.2013.6724141

Krishna, K. G., Santhosh, B., & Sridhar, V. (2013). Design of wallace tree multiplier using compressors. *International journal of engineering sciences & research. Technology*, *2*, 2249–2254.

Kshirsagar, R. D., Aishwarya, E. V., Vishwanath, A. S., & Jayakrishnan, P. (2013, December). Implementation of pipelined booth encoded wallace tree multiplier architecture. In *2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE)* (pp. 199-204). IEEE. 10.1109/ICGCE.2013.6823428

Kuo, T. Y., & Wang, J. S. (2008, May). A low-voltage latch-adder based tree multiplier. In *2008 IEEE International Symposium on Circuits and Systems* (pp. 804-807). IEEE.

Liao, M. J., Su, C. F., Chang, C. Y., & Wu, A. H. (2002, May). A carry-select-adder optimization technique for high-performance Booth-encoded wallace-tree multipliers. In *2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No. 02CH37353)* (Vol. 1, pp. I-I). IEEE. 10.1109/ISCAS.2002.1009782

Luu, X. V., Hoang, T. T., Bui, T. T., & Dinh-Duc, A. V. (2014, October). A high-speed unsigned 32-bit multiplier based on booth-encoder and wallace-tree modifications. In *2014 International Conference on Advanced Technologies for Communications (ATC 2014)* (pp. 739-744). IEEE. 10.1109/ATC.2014.7043485

Nachtigal, M., Thapliyal, H., & Ranganathan, N. (2010, August). Design of a reversible single precision floating point multiplier based on operand decomposition. In *10th IEEE International Conference on Nanotechnology* (pp. 233-237). IEEE. 10.1109/NANO.2010.5697746

Onomi, T., Yanagisawa, K., Seki, M., & Nakajima, K. (2001). Phase-mode pipelined parallel multiplier. *IEEE Transactions on Applied Superconductivity*, *11*(1), 541–544. doi:10.1109/77.919402

Paradhasaradhi, D., Prashanthi, M., & Vivek, N. (2014, March). Modified wallace tree multiplier using efficient square root carry select adder. In *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE)* (pp. 1-5). IEEE. 10.1109/ICGCCEE.2014.6922214

Rao, M. J., & Dubey, S. (2012, December). A high speed and area efficient Booth recoded Wallace tree multiplier for Fast Arithmetic Circuits. In *2012 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics* (pp. 220-223). IEEE. 10.1109/PrimeAsia.2012.6458658

Reddy, B. M., Sheshagiri, H. N., Vijayakumar, B. R., & Shanthala, S. (2014, December). Implementation of Low Power 8-Bit Multiplier Using Gate Diffusion Input Logic. In *2014 IEEE 17th International Conference on Computational Science and Engineering* (pp. 1868-1871). IEEE.

Sarma, R., Bhargava, C., Dhariwal, S., & Jain, S. (2019). UCM: A Novel Approach for Delay Optimization. *International Journal of Performability Engineering*, *15*(4).

Singh, A. K., De, B. P., & Maity, S. (2012). Design and Comparison of Multipliers Using Different Logic Styles. *International Journal of Soft Computing and Engineering*, *2*(2), 374–379.

Sousa, L. A. (2003). Algorithm for modulo (2^n+ 1) multiplication. *Electronics Letters*, *39*(9), 752–754. doi:10.1049/el:20030467

Toh, N., Naemura, Y., Makino, H., Nakase, Y., Yoshihara, T., & Horiba, Y. (2001). A 600-MHz 54-bit multiplier with rectangular-styled Wallace tree. *IEEE Journal of Solid-State Circuits*, *36*(2), 249–257. doi:10.1109/4.902765

Wallace, C. S. (1964). A suggestion for a fast multiplier. *IEEE Transactions on Electronic Computers*, *EC-13*(1), 14–17. doi:10.1109/PGEC.1964.263830

Yi, Q., & Han, J. (2009, July). An improved design method for multi-bits reused booth multiplier. In *2009 4th International Conference on Computer Science & Education* (pp. 1914-1916). IEEE.