ATM: Approximate Toom-Cook Multiplication for Speech Processing Applications

by

Salman Ahmed, Deepthi Amuru, Zia Abbas

Report No: IIIT/TR/2020/-1

Centre for VLSI and Embeded Systems Technology
International Institute of Information Technology
Hyderabad - 500 032, INDIA
October 2020

ATM: Approximate Toom-Cook Multiplication for Speech Processing Applications

Mohammed Salman Ahmed, Deepthi Amuru, Zia Abbas
Center for VLSI and Embedded Systems Technologies (CVEST)
International Institute of Information Technology (IIIT) Hyderabad
Hyderabad, India
Email: salman.ahmed@research.iiit.ac.in, deepthi.amuru@research.iiit.ac.in, zia.abbas@iiit.ac.in

***Abstract*— Approximate Computing has paved way for elaborate savings in design area and latency of modern system architectures processing images or signals, by a deliberate yet tolerable loss of functional accuracy. This paper thus proposes a design of an approximate multiplier based on the efficient Toom-Cook algorithm, that has a lower complexity of $O(N^{log_d(2d-1)})$ than $O(N^2)$, for order d. Inherent integer divisions in the algorithm has restricted its feasibility in hardware, unless without suitable approximation. On an average, the proposed multiplier achieves 53%, 18% and 57% improvements in area, delay and power only with less than 1% mean error. Owing to these benefits due to lower computational complexity, the multiplier can be configured to achieve significant savings with a high quality output and that suits well to the nature of the speech processing systems, hence the design works well for the epoch extraction system in speech.**

Keywords*— *Approximate Computing, Toom-Cook Multiplication, Epoch Extraction, Speech

I. INTRODUCTION

Biased human perceptions and redundancy in data can be translated into design simplifications, with controlled errors in the output, but achieve many-fold savings in chip design. The challenge arises in optimization of the design while maintaining this quality constraint, which is best addressed through approximate circuit design [1].

With exponential growth of data-intensive approaches and slowing down of the Moore's law [2], [3], approximate computing provides an attractive option by exploiting the error resilience of such applications. Over the recent years, a number of techniques have been proposed for approximate addition and multiplication. A few of them are as follows. In [4], RoBA multiplier achieves simplification by rounding the input operands to nearest powers of 2 and as a result the multiplication requires only simple shifts and addition operations. However, the maximum error of 11% is quite high and remains the same with upscaling of the multiplier, as large operands differ significantly from their rounded counterparts. In [5], the approximations are adopted in the partial product addition tree of the multiplier. However, the multiplier is more suited for shorter bit lengths. The proposed multiplier not only outperforms due to its lower complexity, but also shares similarities with the DRUM multiplier [6] in advantages of scalability, flexibility in the choice of the base multiplier, etc.

The paper is organized as follows: In Section II, the Toom-Cook algorithm is explained and the scope and motivation of the approximation that is achieved, is discussed. Section III deals with the design implementation details and Section IV presents the trade-offs and results. Section V discusses the application of the proposed multiplier in epoch extraction. Section VI summarizes and concludes.

II. PROPOSED APPROXIMATE TOOM-COOK MULTIPLICATION

Toom-Cook multiplication algorithm is proven to be faster than conventional multiplication as it has reduced number of sub-multiplications. From the hardware implementation point, the problem of divisions though by small numbers, offset the savings in the multiplications. The issue has been solved recently for modular multiplication [7], [8], however, normal integer multiplication is required in several processing cores, and for the algorithm to be realizable in this respect, must incorporate some approximations.

The algorithm assumes polynomial representation of the input operands, around some base $x = 2^B$, selected based on the size of the input operands and the order d. The polynomial $P(x)$ with coefficients p_i evaluates the multiplicand.

$$P(x) = p_{k-1}x^{k-1} + \cdots + p_2x^2 + p_1x^1 + p_0$$

Similarly, $Q(x)$ with q_i evaluates the multiplier. The final product is evaluated simply as $R(x) = P(x).Q(x)$.

$$\begin{bmatrix} R(x_0) \\ R(x_1) \\ \vdots \\ R(x_{2k-2}) \end{bmatrix} = \begin{bmatrix} x_0^0 & x_0^1 & \dots & x_0^{2k-2} \\ x_1^0 & x_1^1 & \dots & x_1^{2k-2} \\ \vdots & \vdots & \dots & \vdots \\ x_{2k-2}^0 & x_{2k-2}^1 & \dots & x_{2k-2}^{2k-2} \end{bmatrix} \begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{2k-2} \end{bmatrix}$$

$$\mathbf{R} = \mathbf{X}_\mathrm{E}\mathbf{r} \Rightarrow \mathbf{r} = {\mathbf{X}_\mathrm{E}}^{-1}\mathbf{R}$$

The above equations show the evaluation of the product polynomial at different evaluation points. The evaluation matrix inverse (${\mathbf{X}_\mathrm{E}}^{-1}$) can thus provide the product polynomial coefficients $\mathbf{r}$, given the evaluated product at different points. The product $R(x)$ is of degree $2k-2$, requiring $2k-1$ points to solve for its coefficients. Since $\mathbf{X}_\mathrm{E}$ is a Vandermonde matrix, its inverse in the form of upper and lower triangular matrices ($\mathbf{U}^{-1}\mathbf{L}^{-1}$), is given by the following equation [9].

$$l_{i,j} = 0\ (i < j), l_{0,0} = 1, l_{i,j} = \prod_{m=0, m\neq j}^{i} \frac{1}{x_j - x_m}$$

$$u_{i,i} = 1, u_{i,0} = 0, u_{i,j} = u_{i-1,j-1} - u_{i,j-1}x_{j-1}$$

The divisions arise due to the term $\frac{1}{x_j - x_m}$, and are directly dictated by the choice of the evaluation points. Another constraint on this choice is based on the ease of evaluation of the input polynomials, i.e. $P(x)$ and $Q(x)$. The second constraint demands the points either be $0, \pm 1, \infty$ (∞ is a valid option, as explained in [10]) or be chosen in powers of 2 so that operation be represented as a shift in hardware. However, this choice will disregard the first one as for some values of a, b, let the term $\frac{1}{x_j - x_m} = \frac{1}{2^a - 2^b}$, this cannot be in powers

978-1-7281-3320-1/20/$31.00 ©2020 IEEE

of 2 unless $a - b = 1$, which is not the case for the overall set of points chosen.

One might suggest to approximate and choose points such that $2^{a-b} \gg 1$. Since this is applicable only when the size of the input operands is quite large, even a slight difference will be magnified by the larger operand giving significant error deviation. The best solution, thus, is to approximate by limiting the use to the standard evaluation points only, i.e. $0, \pm 1, \infty$.

The first four coefficients of $R(x)$ in the order of significance can be computed by evaluating it at $0, \pm 1, \infty$, which is sufficient to have a good quality output. A reduction in the number of evaluation points means the same reduction in the sub-multiplications.

From [10], it is clear that an even number of evaluation points implies that the input operands are unbalanced. Also, with even points, it becomes easier to solve using the new $\mathbf{X}_E{}^{-1}$ due to the resulting symmetries. The degree of imbalance between $P(x)$ and $Q(x)$ is decided dynamically based on their coefficients. The significant advantage of this skewness in reduced error can be attributed from the following probability equation.

$$P_u(n) = 1 - \frac{n^2 + \sum_{t=1}^{n-1}((n-t)2^{t-1})^2}{(2^n - 1)^2}$$

Neglecting the inputs to be zero, one can prove the above equation to yield the probability that any random set of input operands of size n will be unbalanced after leading zero counting and shifting (as described in Section III). If $n = 8$, then $P_u(n) = 81.27\%$. Thus, this high chance favors the use of the proposed design.

III. Design of Approximate Toom-Cook Multiplier

The proposed design is illustrated in the Fig.1. In fixed-point operands, the leading and ending zeroes do not contribute to the final product, except shift it by a finite number of bits. The multiplicand and multiplier input operands are thus first subjected to leading zero counting (LZC), discarding the leading zeroes. The LZC can be the same as in [6]. The paper [11] also gives an efficient low power implementation for the same. Ending zeroes and some lower significant bits are removed when the operands are split.

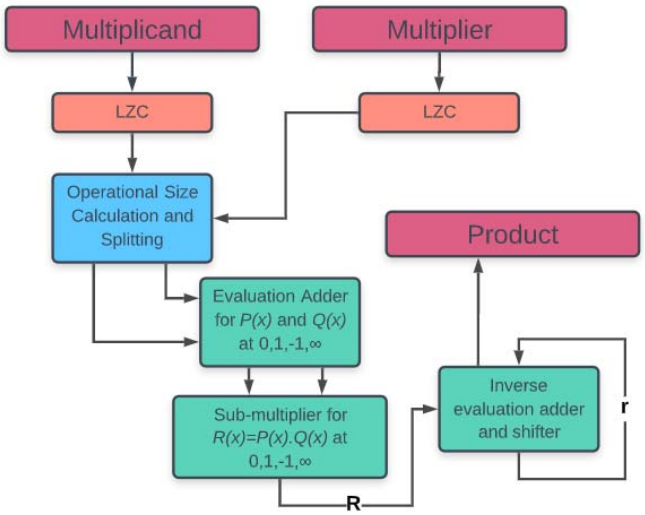

Fig. 1. Approximate Toom-Cook Multiplier

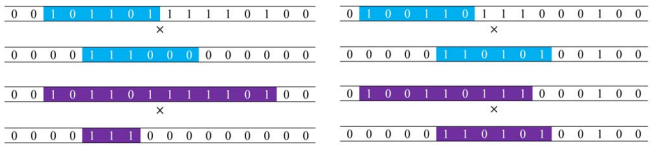

Fig. 2. Dynamic Selection of bits, ATM ($n = 16, b = 3, s = 7.5$) [($4b$, $1b$) left] [($3b$, $2b$) right] and DRUM ($n = 16, b = 6$)

Since the proposed algorithm uses only four evaluation points, the operands could be split or unbalanced only on the order $(2b, 3b)$, $(3b, 2b)$, $(4b, 1b)$ or $(1b, 4b)$ [10], where b is the size of the base multiplier. Hence, for input operands of size n, the size b is at most $n/4$. The degree of approximation can be increased by reducing the size b and hence increase savings in area, delay and power. The LZC reduced inputs are split on b and the order of splitting is chosen using simple OR logic and a small comparator. If second highest degree coefficient of the polynomial is zero, then the order is $(4b, 1b)$ or $(1b, 4b)$, based on the input, for which the above special case arises. Normally, the order is $(2b, 3b)$ or $(3b, 2b)$, the choice is made by partially comparing (on truncated lengths) the third highest degree coefficients. The operational size s is simply the average of the dynamic size of the operands. Note, the size of the base multiplier must be chosen such that $b = \left\lceil \frac{s}{2.5} \right\rceil$.

Fig. 2 helps to elucidate the concept of a better dynamic selection in the proposed design with a brief example, exploiting the inherent imbalance created due to LZC. This clearly implies a reduced error as compared to the DRUM multiplier for roughly the same area or vice versa. After determining the order of splitting, the remaining bits are truncated, i.e. only the highlighted bits enter into the multiplication process.

During the evaluation phase, the points 0 and ∞, do not require any evaluation. The evaluations at points ± 1 are processed by a simple adder/subtractor. The evaluated results are multiplied point wise i.e. four accurate multiplications to yield the evaluations of the product. The choice of the base multiplier is designer-defined as in the case of DRUM. Since its size is divisible and thus smaller, the design could either be parallelized or be implemented in a sequential fashion, based on designer constraints.

The inverse evaluation overhead is also small. The final coefficients $\mathbf{r}$ are extracted by simple additions, subtractions or 1-bit shifts as given below.

$$r_0 = R(0), r_1 = \Big(\big(R(1) - R(-1)\big) \gg 1\Big) - R(\infty),$$

$$r_2 = \Big(\big(R(1) + R(-1)\big) \gg 1\Big) - R(0), r_3 = R(\infty)$$

One might suggest for the special case of order being $(4b, 1b)$ or $(1b, 4b)$, the evaluation and inverse evaluation phases are redundant. This is indeed the case, therefore, the operands are directly passed through and simply multiplied. Finally, the coefficients are shifted based on the total LZC count and the multiples of the size b and are added (as per the polynomial equation given in Section II) to get the approximate product.

IV. Results and Discussions

The proposed design is composed in Verilog HDL and synthesized at $32nm$ technology using Synopsys Design

Compiler, with medium area effort. The supply voltage is set at $0.95V$ and a temperature of 125°C at SS process corner, to effectively test the design in terms of its performance, area and power. The accuracy calculation is done using ModelSim simulator, by exercising the design for about 200k randomly generated input combinations and measuring the difference between the accurate and the approximate product. Some of the popular error metrics for quality measurement are normalized mean error distance (NMED) and mean relative error distance (MRED) [5], [12].

$$NMED = \frac{mean\left(\left|R_{acc.} - R_{approx.}\right|\right)}{(2^n - 1)^2}$$

$$MRED = mean\left(\frac{\left|R_{acc.} - R_{approx.}\right|}{R_{acc.}}\right)$$

Approximate multipliers have also been developed in papers [13], [14] and have been compared and contrasted effectively in [5], [6]. Here, we compare the proposed design with DRUM [6] and an accurate design optimized at the same constraints.

Fig. 3 shows the relative error distribution for ATM with $\mu = 0.92\ \%$ and $\sigma = 0.6\%$. The error accumulation in the final output of a system depends on its structure. A system may realize a filter that continuously subtracts its previous samples from its present samples that may result in a good cancellation of the error, if the error is biased (Section V). It is therefore, important to appreciate the range of error ~3% which is certainly less in the proposed design.

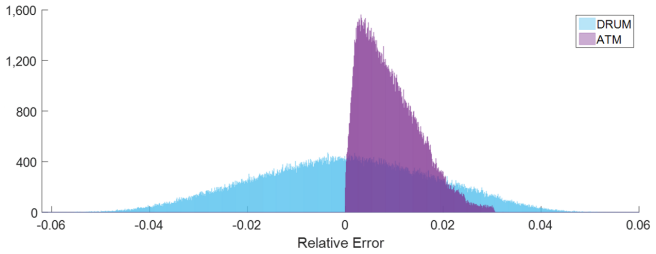

Fig. 3. Relative Error Distribution of ATM ($n = 16, b = 3, s = 7.5$) and DRUM ($n = 16, b = 6$)

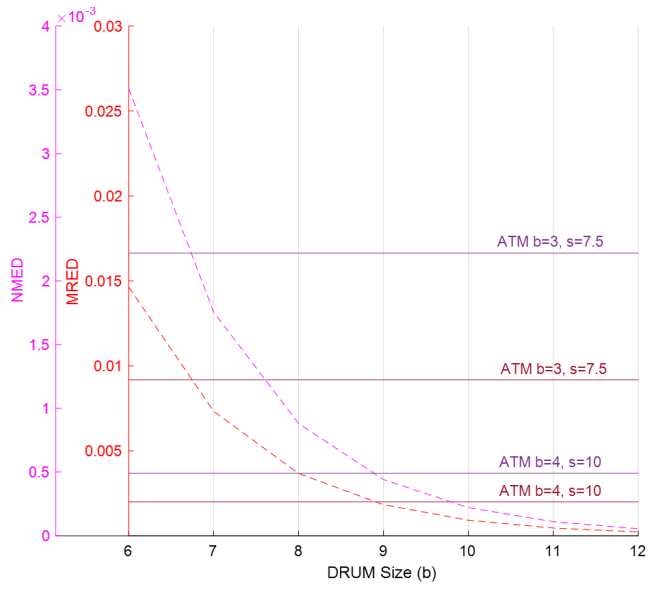

Fig. 4. MRED and NMED comparison of ATM and DRUM

TABLE I. POST-SYNTHESIS DESIGN PARAMETERS COMPARISON

Design	Size (n, b, s)	Design Area (μm^2)	Critical Path Delay (ns)	Leakage Power (μW)	Total Power (μW)
Acc.	16	2228.2	3.71	92.0	713.7
DRUM	16, 7	1633.5	3.48	58.8	309.3
ATM	16, 3, 7.5	1294.2	3.31	49.8	336.8
DRUM	16,9	1993.4	3.87	75.2	452.3
ATM	16, 4, 10	1671.9	3.42	67.0	539.4
Acc.	32	9205.6	6.64	368	3710
DRUM	32, 11	4047.1	5.23	137	771.9
ATM	32, 5, 12.5	2482.3	4.84	95.3	794.8
DRUM	32, 13	4645.0	5.83	161	1031
ATM	32, 6, 15	2739.2	5.15	108	1028

The MRED and NMED are 0.198% and 0.049% for ATM with $b = 4$. Better quality of the product does not justify a fair comparison in terms of the other performances. Fig.4 illustrates the variation of the multiplier size of DRUM and intersecting its curve with the achieved accuracy for ATM. For example, this shows ATM and DRUM with $b = 4$ and $b = 9$, respectively, are almost equal in terms of MRED and NMED. Hence, they can be compared as shown in Table I.

As clear from Table I, the ATM performs quite well except for some increase in dynamic power as compared to DRUM at low sizes, it is because of the associated overhead in evaluation and its inverse that becomes of less significance with increasing size. As compared to the accurate design, ATM with size parameters $b = 6, s = 15$ provides benefits of 70.2%, 22.4%, 70.6% and 72.3% in terms of area, delay, leakage power and total power, respectively. A feature of the proposed design is that it scales effectively while maintaining a certain level of precision, which is crucial for the design of large multipliers.

V. EPOCH EXTRACTION OF SPEECH USING ATM

Approximate circuit design has assumed compatibility with applications of image processing, machine learning and many more. In speech systems, approximations are realizable as evident in this section and also by a recent approach for speech recognition [15].

Epoch or glottal closure instant extraction is a vital task in speech processing systems [16], [17]. The epochs denote the closing of the vocal folds while speaking, extraction of which delves a deeper analysis into the speech signal. The paper [17] presents a stable zero-phase zero-frequency resonator that can tolerate some loss of precision. This motivates the use of the proposed multiplier in the design of the resonator.

The epoch extraction procedure, as adopted for demonstrating the efficacy of ATM, is quite simple. Followed by pre-emphasis, the speech signal is passed through a resonator system described by the following difference equation. The vital multiplications in this application are those represented in the equation.

$$(1 + 4r^2 + r^4)y(n) = x(n) + 2r(r^2 + 1)\big(y(n-1) + y(n+1)\big) - r^2\big(y(n-2) + y(n+2)\big)$$

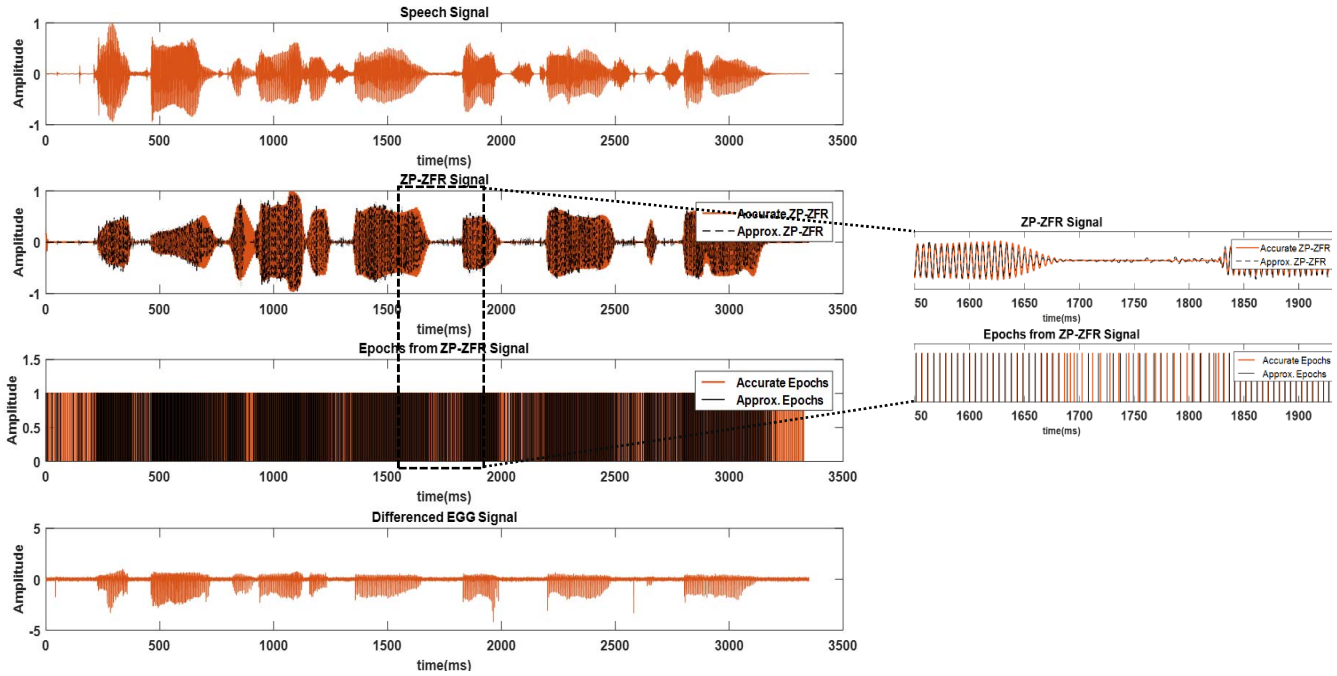

Fig. 5. Epoch Extraction of speech using Accurate multiplier ($n = 16$) and ATM ($n = 16, b = 3, s = 6$)

Here, $y(n)$ corresponds to the ZP-ZFR response and $x(n)$ to the speech signal. The value r is taken as 0.99 for the stable resonator. The trend in the response is removed and its downward peaks define the epochs. The reader is encouraged to refer [17] for more details.

The CMU US slt (US female) Arctic database [18] containing 1132 dual channel signals composed of speech and the electroglottograph (EGG) signals, has been employed as the standard for testing and measuring the identification accuracy of epoch extraction using the proposed multiplier. EGG signals represent the ground truths for the extracted speech signals.

Fig. 5 details the speech extraction process. As represented in a normalized manner, the speech signal is passed through an accurate and an approximate resonator. All the speech samples and r are represented in an integer format by proper scaling so that they can be processed by the approximate multiplier. It is noticeable that the responses closely follow each other. Slight deviations in the epochs are also clear in the magnified version on the right. The deviations occur more in the unvoiced regions, which do not affect the accuracy much, as epochs in this region are unwanted and are ultimately discarded.

Speech extraction quality metrics [16] are identification, miss and false alarm rates characterized based on whether an epoch, no epochs or multiple epochs are detected in a larynx cycle.

From Fig. 6, it is clear that there is only a trivial change in the speech extraction quality while utilizing the proposed ATM provided the operational size s is higher than 7.5. However, a reduction in s further from it, manifests into a rapid deterioration of the performance of the system. The identification accuracy also worsens from 0.17 ms. Hence, a proper choice of s is dictated by any given application. Although the design provides coarse variability precision tuning, it is sufficient for most applications.

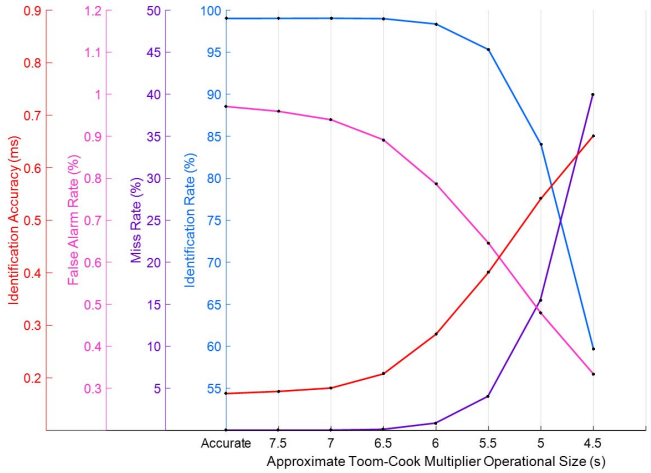

Fig. 6. Variation of Identification Accuracy, False Alarm Rate, Miss Rate and Identification Rate with reduction in operational size s of the Approximate Toom-Cook Multiplier (n= 16)

VI. CONCLUSIONS

In this paper, a novel design of an approximate multiplier inspired by one of the faster multiplication algorithms, i.e. Toom-Cook algorithm is proposed and has been very successful in terms of savings of area, critical path delay and power. And at the same time, the design is easily scalable to high precisions, also keeping up with its performance. The efficacy of the proposed design is tested on the epoch extraction system in speech processing, and it only suffers a trivial degradation up to a certain size, which is indeed tolerable.

ACKNOWLEDGEMENTS

The authors acknowledge the efforts of Mohammad Hashim Javid and Krishna Gurugubelli for their support in development of the epoch extraction system using ATM.

REFERENCES

[1] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," *2013 18th IEEE European Test Symposium (ETS)*, Avignon, 2013, pp. 1-6.

[2] Z. Abbas and M. Olivieri, "Impact of technology scaling on leakage power in nano-scale bulk CMOS digital standard cells," *Microelectronics Journal*, vol. 45, no. 2, pp. 179–195, 2014.

[3] Z. Abbas, A. Mastrandrea, and M. Olivieri, "A voltage-based leakage current calculation scheme and its application to nanoscale MOSFET and FinFET standard-cell designs," *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 22, no. 12, pp. 2549–2560, 2014.

[4] R. Zendegani, M. Kamal, M. Bahadori, A. Afzali-Kusha and M. Pedram, "RoBA Multiplier: A Rounding-Based Approximate Multiplier for High-Speed yet Energy-Efficient Digital Signal Processing," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 2, pp. 393-401, Feb. 2017.

[5] H. Jiang, C. Liu, F. Lombardi and J. Han, "Low-Power Approximate Unsigned Multipliers With Configurable Error Recovery," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 189-202, Jan. 2019.

[6] S. Hashemi, R. I. Bahar and S. Reda, "DRUM: A Dynamic Range Unbiased Multiplier for approximate applications," *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Austin, TX, 2015, pp. 418-425.

[7] Z. Gu and S. Li, "A Division-Free Toom–Cook Multiplication-Based Montgomery Modular Multiplication," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 8, pp. 1401-1405, Aug. 2019.

[8] J. Ding, S. Li and Z. Gu, "High-Speed ECC Processor Over NIST Prime Fields Applied With Toom–Cook Multiplication," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 3, pp. 1003-1016, March 2019.

[9] L. R. Turner, "Inverse of Vandermonde matrix with applications," NASA Lewis Research Center, Cleveland, Ohio, United States, Tech. Report. NASA-TN-D-3547, Aug. 1966.

[10] M. Bodrato, "High Degree Toom'n'Half for Balanced and Unbalanced Multiplication," *2011 IEEE 20th Symposium on Computer Arithmetic*, Tubingen, 2011, pp. 15-22.

[11] G. Dimitrakopoulos, K. Galanopoulos, C. Mavrokefalidis and D. Nikolos, "Low-Power Leading-Zero Counting and Anticipation Logic for High-Speed Floating Point Units," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 7, pp. 837-850, July 2008.

[12] J. Liang, J. Han and F. Lombardi, "New Metrics for the Reliability of Approximate and Probabilistic Adders," in *IEEE Transactions on Computers*, vol. 62, no. 9, pp. 1760-1771, Sept. 2013.

[13] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park and N. S. Kim, "Energy-Efficient Approximate Multiplication for Digital Signal Processing and Classification Applications," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 6, pp. 1180-1184, June 2015.

[14] P. Kulkarni, P. Gupta and M. Ercegovac, "Trading Accuracy for Power with an Underdesigned Multiplier Architecture," *2011 24th International Conference on VLSI Design*, Chennai, 2011, pp. 346-351.

[15] B. Liu, H. Qin, Y. Gong, W. Ge, M. Xia and L. Shi, "EERA-ASR: An Energy-Efficient Reconfigurable Architecture for Automatic Speech Recognition With Hybrid DNN and Approximate Computing," in *IEEE Access*, vol. 6, pp. 52227-52237, 2018.

[16] K. S. R. Murty and B. Yegnanarayana, "Epoch Extraction From Speech Signals," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1602-1613, Nov. 2008.

[17] K. Gurugubelli and A. K. Vuppala, "Stable Implementation of Zero Frequency Filtering of Speech Signals for Efficient Epoch Extraction," in *IEEE Signal Processing Letters*, vol. 26, no. 9, pp. 1310-1314, Sept. 2019.

[18] A. W. Black, "CMU_ARCTIC Databases," *Festvox*. [Online]. Available: http://www.festvox.org/cmu_arctic/. [Accessed: 23-Sep-2019]