An Efficient Parallel DA-Based Fixed-Width Design for Approximate Inner-Product Computation

Basant Kumar Mohanty, *Senior Member, IEEE*, and Pramod Kumar Meher, *Senior Member, IEEE*

***Abstract*— Parallel distributed arithmetic (PDA)-based structures are widely used for high-speed computation of inner product in digital signal processing (DSP) applications. In this article, we have proposed novel PDA-based structures based on an efficient truncation model. To achieve higher bit saving with relatively less truncation error, we present here a novel approach using approximate look-up tables (LUTs), adder trees (ATs), and Wallace-like shift-AT (SAT) with truncated operands to obtain hardware-efficient fixed-width PDA-based inner-product structures. We have three variants of proposed structures based on the proposed truncation approach. We find that the proposed inner-product structure-1 using approximate LUT (ALUT) and approximate AT offers nearly 20% higher bit saving, 20% saving in area-delay product (ADP) and offers relatively less truncation error than the existing structures. The proposed structure-2 using ALUT, ATs, and proposed SAT offers nearly 50% higher bit-saving, 61% ADP saving and offers nearly the same accuracy compared to the existing approximate DA-based structures. Proposed structure-3 offers nearly 60% higher bit saving and calculates outputs with almost the same or marginally less accuracy than the existing structures for higher coefficient word lengths.**

***Index Terms*— Approximate computation, distributed-arithmetic (DA), inner product, shift adder.**

I. Introduction

INNER-product computation forms the core of many important digital signal processing (DSP) functions such as linear and circular convolutions, correlation, digital filtering, and discrete trigonometric transforms [1]. Many of these algorithms involve computation of inner product with a constant vector. Distributed arithmetic (DA) is widely used in such cases for efficient hardware realization of inner products [3]–[11].

DSP functions are increasingly used for the processing of multimedia signals that possess error resilience since the human brain is capable of filling the missing information. This error resilience behavior provides an opportunity to consider approximate computation in order to improve the area-delay efficiency of the design substantially. The potential of approximate computation is yet to be explored fully to optimize the hardware for implementing the DSP algorithms. Approximate logic functions of a full adder (FA) and a compressor are generally used to design low-complexity arithmetic circuits [12]–[14]. The use of logic approximation in arithmetic circuits help to save some combinational logic as well as the critical path delay (CPD) in full-width designs. However, as the word length grows in fixed-point arithmetic, the number of processing bits increases and that leads to a substantial increase in overall area-delay complexity. To avoid such word-length growth, multiplications and additions are implemented in fixed-width designs instead of full-width designs. Three types of truncation are generally used to obtain fixed-width designs: direct truncation, post-truncation, and partial-product truncation. In post-truncation, the lower order bits of the full-width output are truncated, whereas in the case of direct truncation, the lowest order bit of every adder output is truncated. In the case of shift addition, partial results are post-truncated to obtain a fixed-width output. Fixed-width multipliers are conventionally obtained by post-truncation and partial-product truncation, whereas fixed-width adder trees (ATs) are obtained using direct truncation and post-truncation.

Parallel DA (PDA)-based inner-product structures involve several look-up tables (LUTs) and ATs along with a shift-AT (SAT) for final addition. The LUTs and ATs account for more than 90% of the total area of the PDA structures [2]. It is observed that the structures using 16-word LUTs offer a minimum area and power complexities compared to other LUT sizes [2]. Furthermore, reduction in complexity of PDA-based structures is challenging, since there are no redundancies in the DA computation. Recently, the approximate sum-of-product design is used in the low-precision implementation of DA, where it is generally assumed that the elements of input vectors of the sum-of-product are small unsigned factions [15]. The inner-product implementation using approximate radix-8 Booth multiplication algorithm [16] for the realization of finite impulse response (FIR) adaptive filters is presented in [17]. The implementation using radix-8 Booth recording offers an efficient inner-product structure involving a variable coefficient vector. However, it is not a cost-effective choice to use radix-8 Booth recording in an inner-product structure that generally involves constant-coefficient vectors, known *a priori*.

DA-based fixed-width inner-product designs are used in different structures proposed in [3]–[11]. These fixed-width structures use direct truncation and post-truncation. We observe that

Manuscript received September 28, 2019; revised December 20, 2019 and January 12, 2020; accepted January 30, 2020. Date of publication March 11, 2020; date of current version April 24, 2020. *(Corresponding author: Basant Kumar Mohanty.)*

Basant Kumar Mohanty is with the Department of Electronics and Telecommunication Engineering, Mukesh Patel School of Technology Management and Engineering, SVKM's NMIMS (Deemed to be University) at Shirpur Campus, Dhule 425405, India (e-mail: bk.mohanti@gmail.com).

Pramod Kumar Meher is with Jain University, Bengaluru 562112, India (e-mail: pkmeher@gmail.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2020.2972772

the fixed-width structures based on post-truncation or logic approximation process almost the same number of bits as the full-width structure. The use of logic approximation along with post-truncation could help to reduce the area-delay complexity marginally over the post-truncation fixed-width structures at the cost of some accuracy loss. The area-delay saving offered by the fixed-width structures based on logic approximation is insignificant compared to the loss of accuracy in the result. We find that higher area-delay saving could be achieved in fixed-width DA structure by adopting a more suitable design approach. Processing of each and every bit consumes some amount of resource and energy. Therefore, a reduction in the number of processing bits (referred to as bit saving) could be a better design approach than the logic approximation to achieve higher area-delay saving in a fixed-width DA structure. We find that the fixed-width output also can be obtained from the full-width design using truncated operands and that facilitates the use of truncated LUT in DA computation. The use of truncated LUT words ensures higher bit saving in the PDA structure but introduces a significant amount of error in the output. The use of pre-truncation to obtain efficient fixed-width PDA structure has not been explored yet. In this article, we aim at design approximation using a pre-truncation model which could offer higher bit saving with a marginal loss of accuracy. The key contributions of this article are as follows.

1) An approximate fixed-width Wallace-like SAT (WSAT) is designed and customized for PDA structure.
2) Three variants of fixed-width inner-product structures are designed using approximate {LUT, AT, SAT}.
3) Error analysis of the proposed structures is performed.

The rest of this article is organized as follows: The proposed approximate fixed-width WSAT design is presented in Section II. A review of existing fixed-width PDA-based structures is presented in Section III. The proposed fixed-width PDA-based inner-product structure is presented in Section IV. Error analysis and hardware-time complexities are discussed in Section V. Conclusions are presented in Section VI.

II. Proposed Fixed-Width SAT

The area complexity of SAT increases substantially due to the sign extension. Radix-4 and radix-8 Booth multipliers use guard bits in the SAT to shift accumulate the partial products to avoid sign extension. A few schemes have also been proposed to approximate SATs that employ truncation with fixed and variable biases [18], [19]. However, the SAT bit matrix of PDA structure is different from the SAT bit matrix of Booth multipliers. The main differences are 1) one of the input operands is shifted by one bit and 2) does not include sign bit to convert 1's complement partial product into 2's complement form. Consequently, the encoding scheme used in the SAT of radix-4 or radix-8 Booth multipliers cannot be employed to avoid sign extension in the SAT of PDA structure. The bias estimation formula of [18] and [19] also cannot be used. We present the proposed approach to address these issues.

Suppose, a sequence of M numbers, $\mathbf{x} = \{x_1, x_2, \ldots, x_M\}$ each of bit width w and progressively left shifted by one bit.

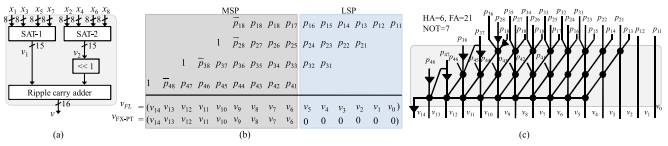

Fig. 1. (a) Block diagram of M-word full-width/fixed-width (based on post-truncation) SAT for $w = 8$. (b) Input bit matrix of $(M/2)$-word SAT with guard bits for sign extension, v_{fl} and v_{fxpt} represent full-width and fixed-width post-truncation output. (c) Bit-flow diagram of four-word WSAT {two-input and three-input nodes, respectively, represents HA and FA and a black triangle represents inverter}.

These shifted samples are added in an M words SAT. $\mathbf{x}$ is decomposed into two subsequences $\mathbf{x_e}$ and $\mathbf{x_o}$, where $\mathbf{x_e} = \{x_1, x_3, \ldots, x_{(M/2)-1}\}$ and $\mathbf{x_o} = \{x_2, x_4, \ldots, x_{(M/2)}\}$, where the adjacent samples of $\mathbf{x_e}$ and $\mathbf{x_o}$ are two-bit left shifted. Shift addition of samples $\mathbf{x}$ using $\mathbf{x_e}$ and $\mathbf{x_o}$ is performed as

$$v = v_1 + z^{-1}v_2 \tag{1a}$$

$$v_1 = \sum_{j=0}^{(w/2)-1} z^{-2j}x_{2j+1}, \quad v_2 = \sum_{j=0}^{(w/2)-1} z^{-2j}x_{2j+2} \tag{1b}$$

where $\{z^{-j}\}$ represents the left bit-shifting operation by the jth bit. As given in (1a) and (1b), M-point shift adder is realized using two separate $(M/2)$-point shift adders as shown in Fig. 1(a) for $M = 8$ and $w = 8$. Each $(M/2)$-point shift adder adds $(M/2)$ input operands with 2-bit progressive shifting. It is to be noted that implementation of SAT using the proposed decomposition scheme does not incorporate any overhead to the design. In fact, it helps to avoid sign extension by introducing guard bits similar to addition of radix-4 Booth encoded partial-products which offers significant saving in area and delay. Input bit-matrix $\mathbf{A}$ of $(M/2)$-point SAT along with guard bits is shown in Fig. 1(b) which is added using a WSAT to produce the full-width output $\{v_{\text{fl}}\}$ of size $(M + w - 1)$ bits. The structure of fixed-width WSAT based on post-truncation and full-width WSAT are identical except for the output word size, where fixed-width WSAT output $\{v_{\text{fxpt}}\}$ of size $(w + 1)$ is obtained from $\{v_{\text{fl}}\}$ by dropping $(M - 2)$ LSBs.

A. Design of Proposed Approximate SAT

The bit-matrix $\mathbf{A}$ [see Fig. 1(b)] is partitioned into two parts: most significant part (MSP) and least significant part (LSP). The lower $(M-2)$ columns of $\mathbf{A}$ form the LSP and the upper $(w + 1)$ columns form the MSP. LSP columns are truncated and a fixed bias is added with the MSP for compensation. The fixed bias is estimated using the probabilistic approach. The output of approximate WSAT is expressed as

$$v = \text{MSP} + 2^{M-2} \cdot \sigma \tag{2}$$

where the $(2^{M-2} \cdot \sigma)$ corresponds to the LSP and (σ) represents the expected value of the LSP is calculated by assuming the probability of each partial-product bit as $(1/2)$, and

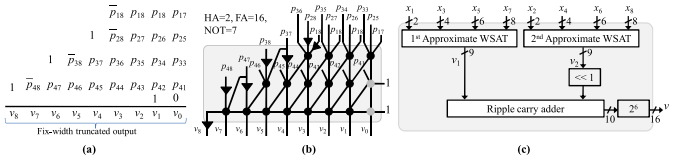

Fig. 2. (a) Input bit-matrix of approximate-WSAT for $(M/2) = 4$ and $w = 8$. (b) Bit-flow graph. (c) Proposed M-point approximate-WSAT.

using the formula

$$E[\text{LSP}] = ((M/2)-1)\left[\left(\frac{1}{2}\right)\cdot 2^{M-3} + \left(\frac{1}{2}\right)\cdot 2^{M-4}\right] + ((M/2)-2)\left[\left(\frac{1}{2}\right)\cdot 2^{M-5} + \left(\frac{1}{2}\right)\cdot 2^{M-6}\right] + \cdots + 2.\left(\frac{1}{2}\right) + \left(\frac{1}{2}\right). \quad (3)$$

The quantized value of σ is calculated as

$$\sigma = \left\lceil \left(\left(\frac{M}{2}\right)-1\right)\left(\frac{3}{8}\right) + \left(\left(\frac{M}{2}\right)-2\right)\left(\frac{3}{32}\right) + \cdots + \left(\frac{3}{2^{M-1}}\right)\right\rceil \approx 2. \quad (4)$$

As shown in Fig. 2(a), the LSP of **A** is truncated and a fixed bias $\sigma = 2$ is added to the MSP for error compensation. The structure of full-width WSAT for $(M/2) = 4$ and $w = 8$ is shown in Fig. 1(c) and the corresponding approximate-WSAT (for $(M/2) = 4$) is shown in Fig. 2(b). The proposed approximate-WSAT is shown in Fig. 2(c). The final output of approximate-WSAT (for $M = 8$) is scaled by $\{2^{M-2}\}$ for normalization of scale. For $(M/2) = 4$ and $w = 8$, the proposed approximate-WSAT involves 2 half-adders (HAs), 16 FAs, and 7 NOT gates where the corresponding full-width WSAT involves 6 HAs, 21 FAs, and 7 NOT gates.

B. Error Performance and Area-Delay Complexity

To evaluate the error performance, the proposed full-width-WSAT and approximate-WSAT designs along with the existing full-width and post-truncated SATs are coded in MATLAB for $N = 8$ and $w = 8, 12$, and 16. The output values that are generated for 10^5 test vectors are derived from uniformly distributed random numbers. The full-width design (WSAT/SAT) is considered the reference design to estimate the error. The maximum error distance (MED), average error distance (AED), average relative error (ARE), average accuracy (AAC)[1] are estimated and the values are listed in Table I. The AED, MED, ARE, and AAC are found to change marginally for higher bit width across all the designs. The post-truncated fixed-width designs of both SAT and WSAT have the lowest AED, MED, ARE, and the highest AAC. The proposed approximate-WSAT has nearly the same ARE and AAC compared with the post-truncated fixed-width SAT and WSAT.

[1]MED = max(abs($v_{\text{fw}} - v_i$)), AED = avg(abs($v_{\text{fw}} - v_i$)), ARE = avg[(abs(ED)/y_{fw})×100], AAC = avg[(1−(abs(ED)/v_{fw}))×100] [21], where, v_{fw} is the output of full-width SAT/WSAT and v_i is the output of fixed-width post-truncated SAT/WSAT or approximate-WSAT.

TABLE I
ERROR ESTIMATES OF FIXED-WIDTH WSAT

Designs	w	AED	MED	ARE (%)	AAC (%)
SAT [10]/ Proposed WSAT post-truncation	8	93.87	187	1.01	98.99
	12	94.86	189	0.11	99.93
	16	94.40	189	0.01	99.99
Proposed approximate WSAT	8	128.46	330	1.36	98.62
	12	128.98	347	0.15	99.91
	16	129.07	367	0.01	99.99

TABLE II
AREA-DELAY ESTIMATES OF FULL-WIDTH AND FIXED-WIDTH SATS

Designs	w	CPD (ns)	Area (sq.um)	ADP (sq.um.ns)	ADP saving
Post-truncated SAT [10]	8	1.2	975	1170	−−
	12	1.45	1375	1994	−−
	16	1.71	1810	3095	−−
Proposed Post-truncated WSAT	8	1.16	910	1056	9.7%
	12	1.41	1316	1856	6.9%
	16	1.67	1696	2832	8.5%
Proposed approximate WSAT	8	1.14	629	717	38.7%
	12	1.39	1029	1430	28.3%
	16	1.65	1436	2369	23.5%

The proposed post-truncated-WSAT and approximate-WSAT are synthesized along with the existing post-truncated SAT in a Synopsys design compiler (DC) using TSMC 65-nm CMOS standard cell library for $M = 8$ and $w = \{8, 12, 16\}$. The area and CPD reported by DC are listed in Table II. As shown in Table II, the proposed post-truncated WSAT involves marginally less area and less CPD compared with the existing post-truncated SAT and calculates the output with the same accuracy. The proposed approximate-WSAT design offers nearly 30% area-delay product (ADP[2]) saving over the existing post-truncated SAT designs on average for word sizes 8, 12, and 16 and calculating the output with marginally lower accuracy. The proposed WSAT is designed specifically for the PDA structure. To take advantage of the proposed approximate-WSAT, different approximation is considered to increase the bit saving in the PDA structure which is discussed in Section III.

III. REVIEW OF EXISTING FIXED-WIDTH PDA-BASED INNER-PRODUCT STRUCTURES

A DA-based structure,[3] which uses full-width LUTs (FLUTs), full-width ATs, and a full-width WSAT and calculates the output with maximum accuracy that could be realized for a given input word sizes, is referred to as full-width structure in this discussion. Suppose, the inner-product computation of N-point input vector $\{x(n)\}$ of bit width $\{b\}$ and coefficient vector $\{x(n)\}$ of bit width $\{w\}$ is performed

[2]ADP = Area×CPD.

[3]The "structure" refers to PDA-based inner-product structure.

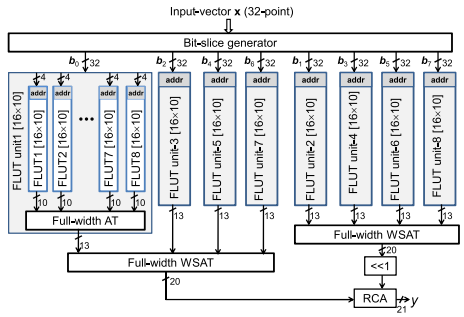

Fig. 3. Full-width structure for $N = 32$, $b = 8$, and $w = 8$. FLUT stands for full-width LUT.

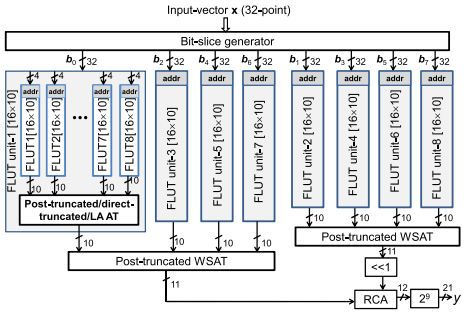

Fig. 4. Existing fixed-width DA structure based on post-truncation [10], direct truncation [2], for $N = 32$, $b = 8$, and $w = 8$.

using 16-word LUTs (for LUT decomposition factor 4), then the bit width of partial inner-product values stored in the LUT is $(w + 2)$. The full-width AT adds $(N/4)$ LUT words and produces an output of bit width $(w + p + 2)$, where $p = \log_2(N/4)$. Finally, full-width SAT shift-adds results of b full-width ATs (with 1-bit relative shift between adjacent operands) and produces an output of bit width $(b + p + w + 2)$. For analysis and discussion, the full-width structure is shown in Fig. 3 for ($N = 32$, $b = 8$, and $w = 8$).

A few DA-based fixed-width FIR filter structures are proposed in [2] and [10]. We have extracted the inner-product structure from the existing DA-based FIR filter structures [2], [10] by removing the delay unit from the FIR filter structure for discussion purpose. The existing fixed-width DA-based structures use FLUT, fixed-width AT, and fixed-width WSAT, where fixed-width AT is based on post-truncation or direct truncation, and fixed-width WSAT is based on post-truncation. The block diagram of these fixed-width DA structures is shown in Fig. 4 for $N = 32$, $b = 8$, and $w = 8$ for discussion. The fixed-width output of 12-bit is scaled by $\{2^{M+p-2} = 9\}$ for normalization of scale with a full-width output.

The existing fixed-width DA-based structure processes almost the same amount of bits as the full-width structure and offers a marginal area-delay saving but introduces relatively a larger amount of error in the inner-product output. Using logic approximation [16] along with post-truncation in the fixed-width AT and WSAT, the area-delay complexity of fixed-width structure could be reduced marginally over the post-truncation fixed-width structure [10] but at the cost of higher accuracy loss compared to other methods. Therefore, the existing truncation models and logic approximation do not lead to an efficient fixed-width structure. The adoption of a more suitable design approach with a different truncation model could help to achieve higher bit saving and area-delay saving in fixed-width DA structure. The proposed design approach is discussed in Section IV.

IV. Proposed Structures

Two fixed-width approximate AT designs are proposed in [20] where the operands are truncated by either $(p = \log_2 M)$ lower-order bits (referred to as AT-1 here) or by $(p-1)$ lower-order bits (referred to as AT-2 here), where M is the input-vector size. Both AT-1 and AT-2 use fixed bias to compensate for the error resulting from the truncated input operand. Out of these two designs, AT-2 offers higher accuracy and almost the same area-delay saving as the AT-1. Both AT-1 and AT-2 could be used in an approximate DA-based inner-product structure. Since AT-1 and AT-2 use truncated input operands (which is the output of LUT), we need not store those bits which are to be truncated by the AT. Instead of that we can use approximate LUTs (ALUTs) of lower bit width. Similarly, when we use the proposed approximate WSAT in a fixed-width DA structure, input operands proposed WSAT could be truncated. Therefore, we need not compute all the sum bits of the ATs (which will get truncated otherwise). The reduction of bit width of AT as well as the LUTs offers a substantial amount of bit saving resulting in substantial saving in overall area, delay, and power consumption of DA structure, since ATs and LUTs contribute to nearly 90% of area and energy cost of the structure. Several choices are available to obtain fixed-width DA structure using LUTs (full-width or approximate), ATs (full-width/post-truncated/direct truncated/truncated operand/logic approximation), and WSAT (post-truncated/truncated operand/logic approximation). The selection of component significantly affects the bit saving and the accuracy of fixed-width DA structure. The existing fixed-width DA structures [2], [10], [16], however, only used fixed-width ATs and WSAT based on post-truncated/direct truncation/logic approximation which offers a marginal bit saving. We explore other variants of fixed-width DA structures in this section.

We have used an indexing method as given in Table III to identify different variants of fixed-width DA-based structures through their constituent components (LUT, AT, and WSAT). Each structure is indexed using the format: $\text{structure}_{\{\text{LUT}\}\{\text{AT}\}\{\text{WSAT}\}}$, where the index number used for {LUT}, {AT}, and {WSAT} are given in Table III. Different variants of fixed-width DA structures are listed along with their components in Table IV. Structure_{252} (shown in Fig. 5) comprises ALUT, AT-2, and post-truncated WSAT.

The fixed-width structure (structure_{153}) using components (FLUT, AT-2, and approximate-WSAT) is shown in Fig. 6. The AT-2 outputs are post-truncated by $\{6, 4, 2, 0\}$ bits before they sent to the approximate-WSAT for shift addition. Similarly, fixed-width DA structures (structure_{123}, structure_{133}, structure_{143}, and structure_{153}) are obtained replacing AT-2 by post-truncated AT, direct-truncated AT and AT-1. These structures (structure_{123}, structure_{133}, structure_{143}, and structure_{153}) are identical with marginal variation in bit saving and output accuracy.

TABLE III
INDEXING SCHEME USED FOR NAMING THE APPROXIMATE STRUCTURES

Index	**Structure**$_{\{LUT\}\{AT\}\{WSAT\}}$		
	LUT	AT	WSAT
1	Full-width	Full-width	Full-width
2	Approximate	Post-truncated	post-truncated
3	−−	Direct-truncated	Approximate
4	−−	AT-1	LA
5	−−	AT-2	−−
6	−−	LA	−−

LEGEND: LA:logic approximation. AT-1: approximate adder-tree type-1, AT-2: approximate adder-tree type-2.

TABLE IV
LIST OF APPROXIMATE STRUCTURES AND THEIR COMPONENTS

Structure	**Component**		
	LUT	AT	WSAT
Structure$_{111}$	full-width	full-width	full-width
Structure$_{112}$	full-width	full-width	post-truncated
Structure$_{114}$	full-width	full-width	LA
Structure$_{122}$	full-width	post-truncated	post-truncated
Structure$_{132}$	full-width	direct-truncated	post-truncated
Structure$_{163}$	full-width	LA	LA
Structure$_{142}$	full-width	AT-2	post-truncated
Structure$_{143}$	full-width	AT-1	approximate
Structure$_{152}$	full-width	AT-2	post-truncated
Structure$_{153}$	full-width	AT-2	approximate
Structure$_{242}$	approximate	AT-1	post-truncated
Structure$_{252}$	approximate	AT-2	post-truncated
Structure$_{223}$	approximate	post-truncated	approximate
Structure$_{243}$	approximate	AT-1	approximate
Structure$_{253}$	approximate	AT-2	approximate

A. Proposed Structure-1

The proposed structure-1 is shown in Fig. 7. It consists of ALUTs, AT-2, and post-truncated WSAT. Each LUT unit comprises $(N/4)$ LUTs of 16-words each. Outputs of $(N/4)$ LUTs of each LUT unit are sent to an AT-2 to compute one partial inner product. AT-2 use LUT words truncated by $\{(p-1)\}$ bits. In other words, AT-2 use only upper $(w+3-p)$ bits of each LUT word of width $(w+2)$, and that creates $[16 \times (p-1) = 2]$ unused bits within each LUT of size $[16 \times (w+2)]$. These unused LUT bits are avoided reducing the LUT width by $\{(p-1) = 2\}$ bits to save LUT resource. Therefore, each word of LUT of size $[16 \times (w+2)]$ is approximated by truncating $\{(p-1) = 2\}$ lower order bits and stored in an ALUT of size $[16 \times (w+3-p)]$. The proposed structure-1 of Fig. 3(a) involves ALUTs of size $[16 \times w]$ for $N = 32$, instead of LUTs of size $[16 \times (w+2)]$ used by the existing structures. Consequently, the proposed structure-1 offers a saving of [16 bM$(p-1)$] LUT bits over the post-truncated structure of [10]. As shown in Fig. 3(a), outputs of AT-2 corresponding to {LUT unit-1, LUT unit-2}, {LUT unit-3, LUT unit-4}, {LUT unit-5, LUT unit-6}, and {LUT unit-7, LUT unit-8} are post-truncated respectively, by $\{6, 4, 2, 0\}$ bits before they sent to approximate-WSAT to compute the fixed-width inner-product output.

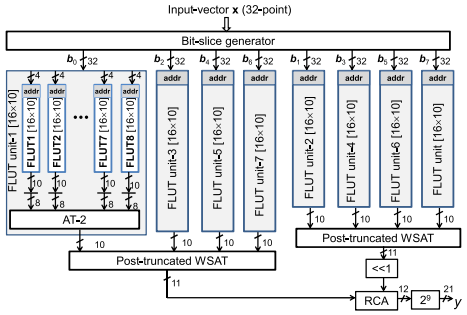

Fig. 5. Fixed-width structure$_{152}$ for $N = 32$, $b = 8$, and $w = 8$.

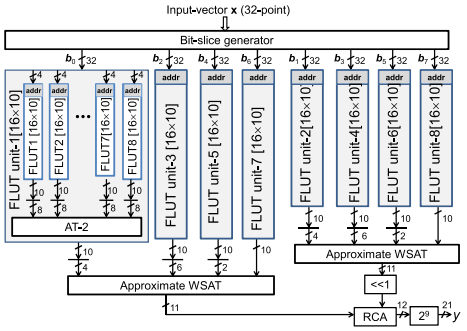

Fig. 6. Fixed-width structure$_{153}$ for $N = 32$, $b = 8$, and $w = 8$.

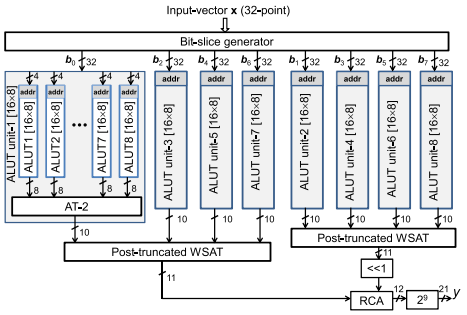

Fig. 7. Proposed structure-1 (structure$_{252}$). ALUT stands for approximate LUT.

B. Proposed Structure-2

The approximate-WSAT of proposed structure-1 (see Fig. 7) uses post-truncation of AT outputs. Instead of post-truncation, the required AT output word size also can be attained by the truncation of AT input operand. Bit width of the set of LUTs associated with the AT is reduced similar to LUT approximation of proposed structure-1 such that the AT outputs are directly sent to the approximate-WSAT without post-truncation. Based on this approach, a fixed-width DA structure (structure$_{223}$) is derived which is shown in Fig. 8. It consists of ALUTs, post-truncation/direct-truncated AT, and approximated-WSAT. Interestingly, this structure involves different size ALUTs instead of a constant size ALUT of proposed structure-1. The fixed-width DA structure (structure$_{223}$) of Fig. 8 involves ALUTs of size $[16 \times (w-4)$,

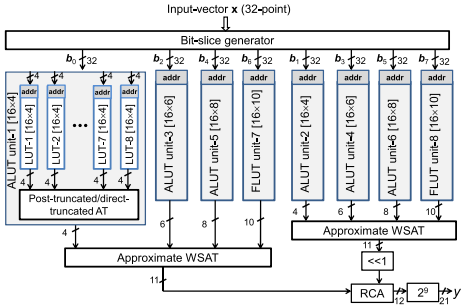

Fig. 8. Fixed-width structure$_{223}$ and structure$_{233}$.

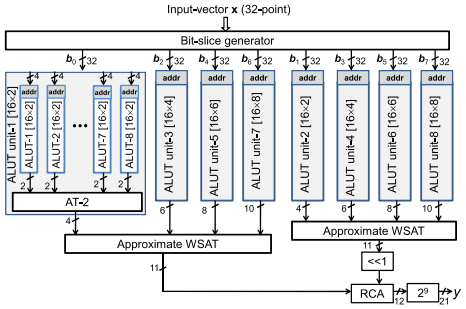

Fig. 9. Proposed structure-2 (structure$_{253}$) for $N = 32$, $b = 8$, and $w = 8$.

$16 \times (w-2), 16 \times w, 16 \times (w+2)]$, respectively, for {LUT unit-1, LUT unit-2}, {LUT unit-3, LUT unit-4}, {LUT unit-5, LUT unit-6}, and {LUT unit-7, LUT unit-8}. The fixed-width DA structure (structure$_{223}$) of Fig. 8 is modified to obtain the proposed structure-2, where AT of structure$_{223}$ is replaced by AT-2. Since AT-2 uses input operand truncated by $\{(p-1)\}$ bits, word size of associated ALUTs further reduced by $(p-1)$ bits. Therefore, the proposed structure-2 use ALUTs of bit width less by $\{(p-1)\}$ than those used by fixed-width DA structure (structure$_{223}$). The proposed structure-2 is shown in Fig. 9.

C. Proposed Structure-3

The proposed structure-1 and structure-2 employ pre-truncation and post-truncation at the AT and SAT stages. To achieve the higher bit saving, we have performed pre-operand truncation in proposed structure-3. The structure of Fig. 8 is also modified by replacing the AT by AT-1 to obtain the proposed structure-3. AT-1 uses input operands truncated by p bits which is one bit higher than input-operand truncation bits of AT-2. Bit width of ALUTs associated with AT-1 is further reduced by p bits. The proposed structure-3 use ALUTs of sizes $[16 \times (w-7)]$, $[16 \times (w-5)]$, $[16 \times (w-3)]$, and $[16 \times (w-1)]$, respectively, for {LUT unit-1, LUT unit-2}, {LUT unit-3, LUT unit-4}, {LUT unit-5, LUT unit-6}, and {LUT unit-7, LUT unit-8}. The proposed structure-3 is obtained from the proposed structure-2 of Fig. 9 by replacing AT-2 by AT-1 and LUT units comprising ALUTs of appropriate sizes. Due to smaller width of ALUT, the proposed structure-3 offers higher bit saving than the proposed structure-2 but introduces higher error in the inner-product output as well.

TABLE V
ESTIMATES OF BIT-SAVING (FOR $N = 32$, $b = 8$, $w = 8$)

Structure	Components			Bit-saving			
	LUT	AT	SAT	LUT	AT	SAT	TBS
Structure$_{112}$	F	F	PT	0	0	0	0%
Structure$_{114}$	F	F	LA	0	0	0	0%
Structure$_{122}$	F	PT	PT	0	0	24	0.2%
Structure$_{164}$	F	LA	LA	0	0	24	0.2%
Structure$_{132}$	F	DT	PT	0	56	24	0.7%
Structure$_{142}$	F	A_1	PT	0	192	24	2%
Structure$_{143}$	F	A_1	AP	0	192	48	2.2%
Structure$_{152}$	F	A_2	PT	0	128	24	1.4%
Structure$_{153}$	F	A_2	A	0	128	48	1.6%
Structure$_{242}$	A_a	A_1	PT	3072	256	24	31%
Proposed-1	A_b	A_2	PT	2048	128	24	20%
Structure$_{223}$	A_c	PT	A	3072	192	48	30%
Proposed-2	A_e	A_2	A	5120	320	48	50%
Proposed-3	A_d	A_1	A	6144	384	48	60%

LEGEND: AT: adder-tree, SAT: shift-adder-tree, F: full-width, A: approximate, A_1: AT-1, A_2: AT-2, PT: post-truncated, DT: direct-truncated, LUT: look-up-table, TBS: Total bit-saving. A_a= LUT truncation by 2-bits, A_b= LUT truncation by 3-bits, A_c= LUT truncation by {6,4,2,0}-bits, A_d=LUT truncated by {9,7,5,3}-bits, A_e= LUT truncated by {8,6,4,2}-bits. TBS in (%) = (TBS/TB$_F$)$\times$100.

The proposed structures along and the other approximate structure based on ALUT offer higher bit saving than the structures based on FLUT, but the ALUT introduces a substantial amount of error into the AT output primarily due to missing carries originating from the truncated part of the LUT word. The AT output error magnitude further scaled in the WSAT due to shift addition. Error analysis and the bit saving offered by the proposed structures and the other approximate structures are presented in Section V.

V. ERROR ANALYSIS AND HARDWARE-TIME COMPLEXITIES

A. Estimate of Bit Saving

A DA-based structure comprises b LUT units, b ATs of size M each and one SAT of size b, where each LUT-unit is further composed of M LUTs, where $M = N/4$, N is the inner-product size, b is the input vector bit width. The depth and width of each accurate LUT are $[16 \times (w+2)]$, where w is the coefficient-vector bit-width. The accurate DA-based structure (structure$_{111}$) involves $[b\{16M(w+2) + (N/4)(w+2) + (w+p+2)\}]$ bits and calculates the output without any approximation. The fixed-width AT (post-truncation-type) offers a saving of (bp) bits in the WSAT, and the fixed-width AT (direct truncation-type) offers a saving of $(M-1)$ and (bp) bits in the AT stage and WSAT stage, respectively. We have extracted an inner-product structure from the adaptive FIR filter structure [17] which uses logic approximation and post-truncation at AT and SAT stages to obtain fixed-width output. The AT-1 offers a saving of $[16 \times$ (Mbp)] bits within the LUT stage, (Mp) bits within the AT stage and (bp) bits within the WSAT. Similarly, the AT-2 fixed-width AT offers a saving of $[16 \times$ (Mb)$(p-1)]$ bits within the LUT stage, $M(p-1)$ bits within the AT stage

TABLE VI
ERROR ESTIMATES OF PROPOSED APPROXIMATE STRUCTURES AND EXISTING STRUCTURES

Structure	Component type			BS	word-length $w = 8$			word-length $w = 12$			word-length $w = 16$		
	LUT	AT	SAT	(%)	AED	ARE (%)	AAC (%)	AED	ARE (%)	AAC (%)	AED	ARE (%)	AAC (%)
Structure_{112} [10]	F	F	PT	0	94.7	2.11	99.7950	94.52	0.086	99.9864	94.5	0.005	99.9992
Structure_{122} [10]	F	PT	PT	0.2	978.5	21.9	97.8844	981.1	1.01	99.8587	977.1	0.056	99.9922
Structure_{144} [17]	F	LA	LA	0.2	1105.8	22.6	97.1825	1116.2	1.04	99.8212	1113.1	0.068	99.9860
Structure_{132} [2]	F	DT	PT	0.7	1606.9	36.04	96.5102	1609.8	1.49	99.7672	1607.6	0.093	99.9871
Structure_{142}	F	A_1	PT	2.0	1452.2	45.03	96.6880	1455.4	1.35	99.7642	1458.6	0.092	99.9873
Structure_{143}	F	A_1	A	2.2	1740	49.09	96.2377	1751.5	1.53	99.7529	1754.8	0.100	99.9860
Structure_{152}	F	A_2	PT	1.4	685.9	19.38	98.5240	683.2	0.54	99.9016	698.7	0.043	99.9944
Structure_{153}	F	A_2	A	1.6	955.8	26.01	97.9437	947.9	0.74	99.8636	977.8	0.058	99.9922
Structure_{242}	A_a	A_1	PT	31.0	1429.4	18.16	96.9656	1467.3	1.57	99.8123	1510.3	0.18	99.9887
Proposed-1	A_b	A_2	PT	20.0	699.1	8.32	98.5472	700.1	0.85	99.9109	698.8	0.070	99.9947
Structure_{223}	A_c	PT	A	30.0	4233.9	48.98	91.2258	4208.8	4.99	99.4609	4225.4	0.55	99.9683
Proposed-2	A_e	A_2	A	50.0	1024.1	12.4	97.8372	996.5	1.32	99.8733	1038.2	0.13	99.9924
Proposed-3	A_d	A_1	A	60.0	3013.3	38.91	93.4707	2996.2	3.12	99.6195	2982.4	0.34	99.9776

and (bp) bits within the WSAT. Approximate WSAT offers a saving of $[b(b-2)/2]$ bits within the WSAT. Using these formulas, we have estimated bit saving offered by different approximate DA structures listed in Table IV using accurate and approximate components of LUT, AT, and WSAT. We have considered $N = 32$, $b = 8$, and $w = 8$ for all the structures. The full-width structure is considered the reference design for a comparison of bit saving. As shown in Table V, the existing inner-product structures (structure_{112} and structure_{122} [10], structure_{132} [2], structure_{142}) [17]) offer a marginal bit saving compared to the proposed structures. The proposed structure-1 and structure-2 offer nearly 20% and 50% higher bit saving over the structure of [10], which is significant.

B. Error Analysis

To evaluate the error performance, we have coded all the structures listed in Table V in MATLAB for $N = 32, b = 8$, and coefficient bit width $w = 8, 12, 16$. Inner-product outputs are produced for 10^5 different sets of randomly generated 32-word test-vectors comprising integers. The full-width structure comprising {accurate LUT, full-width AT, and full-width WSAT} is considered the reference design to estimate the error. The {AED, ARE, and AAC}[4] are estimated for the proposed structures and other approximate structures including existing structures. The estimated values are listed in Table VI. As shown in Table VI, AED changes marginally, whereas AAC is maximized for higher coefficient word sizes across all the designs.

As shown in Table VI, the existing structures based on post-truncation [10] and logic approximation [17] offers the lowest AED and MED, and higher AAC when compared with the proposed structures, but they offer a negligible amount of bit saving. The structure of [2] based on direct truncation offers marginally higher bit saving over the post-truncated structures of [10] and [17] but offers relatively higher AED and MED and lower AAC. The proposed structure-1 (comprising ALUT, AT-2, and post-truncation WSAT) has the highest AAC and offers nearly 20% higher bit saving over the existing structure [10] based on post-truncation. The proposed structure-2 (comprising ALUT, AT-2, and approximate-WSAT) has marginally less AAC and offers nearly 50% higher bit saving over the existing structures of [10]) and [2]. The proposed structure-3 (comprising ALUT, AT-1, and approximate-WSAT) offers marginally less AAC than the existing structures for higher coefficient word sizes and offers almost 60% higher bit saving which is the highest among the proposed structures. Based on error analysis and estimate of bit saving, we find that the proposed structure-1 is a conservative design that is more suitable for the smaller coefficient word-size implementations with higher accuracy. On the other hand, the proposed structure-2 and structure-3 are

[4]AED = avg(abs($y_{\text{fw}} - y_{\text{fx}}$)), ARE = avg[((ED)/$y_{\text{fw}}$) × 100], AAC = avg[(1 − ((ED)/y_{fw})) × 100] [21], where y_{fw} is the output of full-width structure and y_{fx} is the output of fixed-width structures.

TABLE VII
ESTIMATED PSNR VALUES OF FILTERED IMAGES

Structures	PSNR (in dB)		
	Pepper	Baboon	Barbara
Existing (Post-truncated) [10]	145.0	144.4	145.2
Proposed structure-1	158.0	160.7	160.4
Proposed structure-2	154.1	152.6	156.0
Proposed structure-3	152.0	152.1	153.0

TABLE VIII
COMPARISON OF HARDWARE COMPLEXITY (FOR $N = 32, b = 8$)

Structure	$w = 8$			$w = 12$		
	ROM	MUX	FAs	ROM	MUX	FAs
Structure [17]	0	7392	1543	0	10080	2056
Structure [10]	10240	0	718	14336	0	970
Structure [2]	10240	0	630	14336	0	882
Proposed-1	8192	0	518	12288	0	770
Proposed-2	5120	0	311	9216	0	535

TABLE IX
COMPARISON OF SYNTHESIS RESULTS OF PROPOSED PDA INNER-PRODUCT DESIGNS AND EXISTING DESIGNS

Design	Component types			word-length $w = 8$			word-length $w = 12$			word-length $w = 16$		
	LUT	AT	SAT	**CPD** (ns)	**Area** (um^2)	**ADP** (um^2.us)	**CPD** (ns)	**Area** (um^2)	**ADP** (um^2.us)	**CPD** (ns)	**Area** (um^2)	**ADP** (um^2.us)
$Structure_{111}$	AC	AC	AC	2.43	20105	48.9	2.73	26181	71.5	2.97	32193	95.6
$Structure_{122}$ [10]	AC	PT	PT	2.23	19805	44.2	2.53	25781	65.2	2.77	31793	88.1
$Structure_{132}$ [2]	AC	DT	PT	2.19	18831	41.2	2.45	24926	61.1	2.69	31542	84.8
Proposed Structure-1	AP_b	AP_2	PT	2.13	16544	35.2	2.37	22808	54.1	2.63	28.9	76.0
Proposed Structure-2	AP_e	AP_2	AP	1.77	9834	17.4	2.05	18167	37.2	2.31	25.5	58.9

relaxed designs which offer higher bit saving at the cost of some accuracy loss, where the proposed structure-2 is suitable for low as well as moderate coefficient word sizes, and the proposed structure-3 is suitable for higher coefficient word sizes.

To further study the error performance of the proposed structures and existing structures of [10] and [2], we have considered image filtering application. We have coded the proposed structures, existing full-width structure, and fixed-width structure based on post-truncation for realization of 32-tap low-pass FIR filter (passband cut-off frequency = 40 kHz, stopband cut-off frequency = 96 kHz, passband ripple = 0.00057565, and stopband ripple = 10^{-4}) and 16-bit coefficients. The low-pass filter is used for the realization of separable 2-D FIR for image filtering. The filtered image obtained from the full-width structure ($structure_{111}$) is considered the reference image for the estimation of the peak signal-to-noise ratio (PSNR). The estimated PSNR values are listed in Table VII. As shown in Table VII, the filtered images obtained from the proposed structure-1, structure-2, and structure-3 have, respectively, nearly 15, 10, and 7 dB higher PSNR than the filtered image obtained from the fixed-width DA structure based on post-truncation [10]. Note the proposed structures use fixed bias for error compensation, whereas the structure based on post-truncation does not use any such bias.

C. Hardware Complexity

We have estimated hardware complexity of the proposed structure and the existing structures, and the estimated values are listed in Table VIII for $N = 32, b = 8, w = 8$ and 12. As shown in Table VIII, the proposed structure-1 involves nearly less than half the ROM bits and FAs than those required by the structures of [2] and [10]. The proposed structure-1 involves marginally higher ROM bits against bit MUXEs required by the structure of [17] and involves nearly one-third of FAs than those of [17]. Interestingly, the proposed structure-2 involves less ROM bits against bit MUXEs and less than one-third of FAs than those required by the structure mentioned in [17].

D. Synthesis Result

The proposed structure-1, proposed structure-2, and the existing structures mentioned in [2] and [10] along with full-width structure are synthesized in a Synopsys DC using TSMC 65-nm CMOS standard cell library. We have considered inner-product size $N = 32$, 16-word LUT, input signal word length $b = 8$ for all the designs and synthesized the designs for different coefficient word lengths ($w = 8, 12, 16$) to study hardware and time complexities. The area and minimum cycle period (MCP) reported by DC are listed in Table IX. As shown in Table IX, the proposed structures offer higher area saving and a marginal delay saving than the existing structures for different coefficient bit widths. We have estimated ADP for all the structures, and the estimated values are also listed in Table IX. For coefficient bit width $w = 8$, the proposed structure-1 offers nearly 20% higher ADP saving and nearly 1% higher AAC over the structure mentioned in [10]. The proposed structure-2 offers nearly 61% ADP saving and marginally less AAC than the $structure_{122}$ mentioned in [10]. Interestingly, for higher coefficient bit width the relative ADP saving offered by the proposed structures is marginally reduced but AAC improves substantially and almost matches the AAC of the structure mentioned in [10], which has the highest AAC amongst the existing structures. The reduction in ADP saving for higher word sizes is mainly due to the bit saving offered by the proposed structures is independent of the variation of coefficient word size. Compared with the full-width structure, the proposed structure-2 offers nearly (64% and 48%) less ADP, for $w = 8$ and 12 with a marginal loss of AAC.

TABLE X
COMPARISON OF SYNTHESIS RESULTS OF FIR STRUCTURES

Designs	**MCP** (ns)	**Area** (um^2)	**Power** (mW)	**ADP** (um^2.us)
DA-FIR [10]	2.48	60081	12.11	149
DA-BFIR (block-size 4) [2]	2.52	221333	43.45	139
Proposed structure-1	3.2	33652	4.7	108
Proposed structure-2	2.9	29730	4.6	86

An FIR structure is obtained using the proposed inner-product structures by integrating a delay chain to the input signal. The proposed FIR structures are synthesized in a Synopsys DC using TSMC 65-nm CMOS standard cell library (for filter length $N = 32$, $b = 8$, and $w = 16$). The existing DA-based FIR structure of [10] and the DA-based block-FIR structure of [2] are also synthesized for the same filter length, input, and coefficient word sizes. The area, MCP, and power consumption reported by DC are listed in Table X.

As shown in Table X, the FIR structures based on the proposed inner-product structure-1 and structure-2 involve (27% and 42%) less ADP[5] compared to those of existing DA-based FIR structure of [10], respectively, and calculates outputs almost with the same accuracy as the existing structure. Compared with the DA-based block FIR structure of [2], the FIR structures based on the proposed inner-product structure-1 and structure-2 involve (25% and 38%) less ADP and calculates the output with higher accuracy.

VI. Conclusion

In this article, we have proposed three variants of approximate PDA-based structures based on an efficient truncation model. To achieve higher bit saving with relatively less truncation error, we have presented here a novel approach using approximate LUTs, ATs, and WSAT with truncated operands to obtain hardware-efficient fixed-width PDA-based inner-product structures. We find that the proposed structure using ALUT and approximate AT offers nearly 20% higher bit saving, 20% saving in ADP and offers relatively less truncation error than the existing structures. The proposed structure-2 using ALUT, ATs, and proposed SAT offers nearly 50% higher bit saving, 61% ADP saving and offers nearly the same accuracy compared to the existing DA-based structures [10]. Proposed structure-3 offers nearly 60% higher bit saving and calculates the outputs with almost the same or marginally less accuracy than the existing structure for higher coefficient word lengths. The FIR filters based on the proposed structure-1 and proposed structure-2, respectively, offer nearly 27% and 42% less ADP than the similar existing structures and calculate output with higher or nearly the same accuracy.

References

[1] K. K. Parhi, *VLSI Digital Signal Processing*. New York, NY, USA: Wiley, 1998.

[2] S. K. Singhal and B. K. Mohanty, "Efficient parallel architecture for fixed-coefficient and variable-coefficient FIR filters using distributed arithmetic," *J. Circuits, Syst., Comput.*, vol. 25, no. 7, Jul. 2016, Art. no. 1650073.

[3] S. White, "Applications of distributed arithmetic to digital signal processing: A tutorial review," *IEEE ASSP Mag.*, vol. 6, no. 3, pp. 4–19, Jul. 1989.

[4] D. Allred, H. Yoo, V. Krishnan, W. Huang, and D. Anderson, "LMS adaptive filters using distributed arithmetic for high throughput," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 7, pp. 1327–1337, Jul. 2005.

[5] S.-S. Jeng, H.-C. Lin, and S.-M. Chang, "FPGA implementation of FIR filter using M-bit parallel distributed arithmetic," in *Proc. IEEE Int. Symp. Circuits Syst.*, Sep. 2006, pp. 1–4.

[6] H.-C. Chen, J.-I. Guo, T.-S. Chang, and C.-W. Jen, "A memory-efficient realization of cyclic convolution and its application to discrete cosine transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 3, pp. 445–453, Mar. 2005.

[7] P. K. Meher, "Unified systolic-like architecture for DCT and DST using distributed arithmetic," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 12, pp. 2656–2663, Dec. 2006.

[8] P. Meher, "Hardware-efficient systolization of DA-based calculation of finite digital convolution," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 8, pp. 707–711, Aug. 2006.

[9] P. K. Meher, S. Chandrasekaran, and A. Amira, "FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3009–3017, Jul. 2008.

[10] P. K. Meher, "New approach to look-up-table design and memory-based realization of FIR digital filter," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 3, pp. 592–603, Mar. 2010.

[11] S. Y. Park and P. K. Meher, "Efficient FPGA and ASIC realizations of a DA-based reconfigurable FIR digital filter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 7, pp. 511–515, Jul. 2014.

[12] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressor," *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 984–994, Apr. 2015.

[13] S. Venkatachalam and S.-B. Ko, "Design of power and area efficient approximate multipliers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 5, pp. 1782–1786, May 2017.

[14] G. Zervakis, K. Tsoumanis, S. Xydis, D. Soudris, and K. Pekmestzi, "Design-efficient approximate multiplication circuits through partial product perforation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 10, pp. 3105–3117, Oct. 2016.

[15] S. Venkatachalam and S.-B. Ko, "Approximate sum-of-products designs based on distributed arithmetic," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 8, pp. 1604–1608, Aug. 2018.

[16] H. Jiang, J. Han, F. Qiao, and F. Lombardi, "Approximate radix-8 booth multipliers for low-power and high-performance operation," *IEEE Trans. Comput.*, vol. 65, no. 8, pp. 2638–2644, Aug. 2016.

[17] H. Jiang, L. Liu, P. P. Jonker, D. G. Elliott, F. Lombardi, and J. Han, "A high-performance and energy-efficient FIR adaptive filter using approximate distributed arithmetic circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 1, pp. 313–326, Jan. 2019.

[18] C.-Y. Li, Y.-H. Chen, T.-Y. Chang, and J.-N. Chen, "A probabilistic estimation bias circuit for fixed-width Booth multiplier and its DCT application," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 4, pp. 215–219, Apr. 2011.

[19] B. K. Mohanty and V. Tiwari, "Modified PEB formulation for hardware-efficient fixed-width booth multiplier," *Circuits Syst. Signal Process.*, vol. 33, no. 12, pp. 3981–3994, Dec. 2014.

[20] B. K. Mohanty, "Efficient design for fixed-width adder-tree," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 2, pp. 292–296, Feb. 2019.

[21] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," *IEEE Trans. Comput.*, vol. 62, no. 9, pp. 1760–1771, Sep. 2013.

[5] ADP = Area × MCP/block-size.

Basant Kumar Mohanty (Senior Member, IEEE) is currently associated with the Mukesh Patel School of Technology Management and Engineering (MPSTME), SVKM's NMIMS at Shirpur campus, Dhule, India, as a Professor and an Associate Dean (Research). His research interests include algorithm and architecture design for signal processing applications, approximate computation, reconfigurable computing, wireless communication, and healthcare.

Dr. Mohanty was a recipient of the Sydney R. Parker Best Paper Award and the M. N. S. Swamy Best Paper Award of *Circuits, Systems, and Signal Processing* in 2018.

Pramod Kumar Meher (Senior Member, IEEE) has served in senior research positions at Nanyang Technological University, Singapore, and the Institute for Infocomm Research, Singapore, from 2004 to 2016. He has contributed to more than 250 technical articles that include more than 85 articles in the IEEE journals. His research interests include design of dedicated and reconfigurable architectures pertaining to signal, image, and video processing, communication, bio-informatics, and intelligent computing.