Efficient Design for Fixed-Width Adder-Tree

Basant Kumar Mohanty, *Senior Member, IEEE*

***Abstract*—Conventionally, fixed-width adder-tree (AT) design is obtained from the full-width AT design by employing direct or post-truncation. In *direct-truncation*, one lower order bit of each adder output of full-width AT is post-truncated, and in case of *post-truncation*, $\{p\}$ lower order-bits of final-stage adder output are truncated, where $p = \lceil \log_2 N \rceil$ and N is the input-vector size. Both these methods does not provide an efficient design. In this paper, a novel scheme is presented to obtain fixed-width AT design using truncated input. A bias estimation formula based on probabilistic approach is presented to compensate the truncation error. The proposed fixed-width AT design for input-vector sizes 8 and 16 offers $(37\%, 23\%, 22\%)$ and $(51\%, 30\%, 27\%)$ area-delay product (ADP) saving for word-length sizes $(8, 12, 16)$, respectively, and calculates the output almost with the same accuracy as the post-truncated fixed-width AT which has the highest accuracy among the existing fixed-width AT. Further, we observed that Walsh-Hadamard transform based on the proposed fixed-width AT design reconstruct higher-texture images with higher peak signal to noise ratio (PSNR) and moderate-texture images with almost the same PSNR compared to those obtained using the existing AT designs. Besides, the proposed design creates an additional advantage to optimize other blocks appear at the upstream of the AT in a complex design.**

***Index Terms*—Adder, approximate design, arithmetic circuit**

I. Introduction

Low-power, area-efficient and high-performance computing systems are increasingly used in portable and mobile devices. For such applications, digital signal processing (DSP) algorithms are implemented in fixed-point VLSI systems. Adder-tree (AT) commonly used in parallel designs of inner-product computation and matrix-vector multiplication. Multiplier design also involves a shift-adder-tree (SAT) for accumulation of partial product bits. Word-length growth is a common problem encountered when multiplication and addition are performed in fixed-point arithmetic. The shape of the bit-matrix of SAT is different from the AT. Consequently, word-length grows in a different order in SAT and AT. Besides, there are few other bits also added in the SAT to take care of negative partial products of multiplier. Specific designs have been suggested for efficient realization of fixed-width multipliers with less truncation error [1]. However, the scheme used in fixed-width multiplier is not appropriate to develop a fixed-width AT design due to different shaped bit-matrix. The full-width AT (FL-AT) design produces $(w + p)$-bit output for every N-point input-vector, where $p = \log_2 N$. For the same size input-vector, the fixed-width AT (FX-AT) design produces w-bit output. Conventionally, FX-AT design is obtained from the FL-AT design by employing *direct* or *post-truncation*. In direct-truncation (DT), one lower order bit of each adder output of FL-AT is post-truncated, and in case post-truncation, $\{p\}$ lower order-bits of final adder output of FL-AT are truncated. In recent years, several schemes have been suggested for approximate computation of addition using ripple carry adder (RCA) to save critical path delay (CPD) and area [2]–[5]. The bio-inspired lower part OR adder is proposed based on approximate logic [2]. Four different types of approximate adder designs are proposed in [3]. An approximate 2-bit adder is proposed in [5] for approximate computation of triple multiplicand without carry propagation. These approximate designs can be used to implement RCA with less delay and area with some loss of accuracy. The approximate RCA design can be used to obtain fixed-width AT employing post-truncation. However, the approximate fixed-width AT (APX-FX-AT) does not offer an area-delay efficient design.

Bit-level optimization of FL-AT for multiple constant multiplication (MCM) is proposed to take advantage of shifting operation [6]. An efficient FL-AT design is proposed in [7] using the approximate adder of [3] for imprecise realization of Gaussian filter for image processing applications. We find that the optimized AT of [6] is specific to MCM based design and none of the existing design discusses the issues related to fixed-width implementation of AT. It is observed that direct-truncation and post-truncation methods does not provide an efficient FX-AT design. It is necessary to have a different approach for developing efficient FX-AT design which is currently missing in the literature. An efficient FX-AT design certainly help to improve the efficiency of dedicated VLSI systems implementing complex DSP algorithm. In this paper, we propose a scheme to develop an efficient FX-AT design with truncated input. Use of truncated input in FX-AT offers two fold advantages: (1) area and delay saving within the FX-AT due to reduction in adder-width (by p-bits), and (2) creates a scope to optimize other computing blocks appear at the upstream of AT in a complex design. However, the use of truncated input introduces a large amount of error in the FX-AT output which needs to be biased appropriately. The main contribution of the paper are:

- Use of truncated input in fixed-width AT design.
- Formula to estimate the bias for error compensation.

Two separate FX-AT designs with different accuracy and complexity are proposed using truncated input. The rest of the paper is organized as: proposed truncation scheme and error-compensation for fixed-width AT is presented in Section II. Hardware and time complexities are discussed in Section III. Conclusions given in Section IV.

II. Truncation and Bias Estimation Formula

To discuss the truncation scheme, bits of N-point input-vector are arranged in a $(N \times w)$ size matrix $\mathbf{A}$ such that the

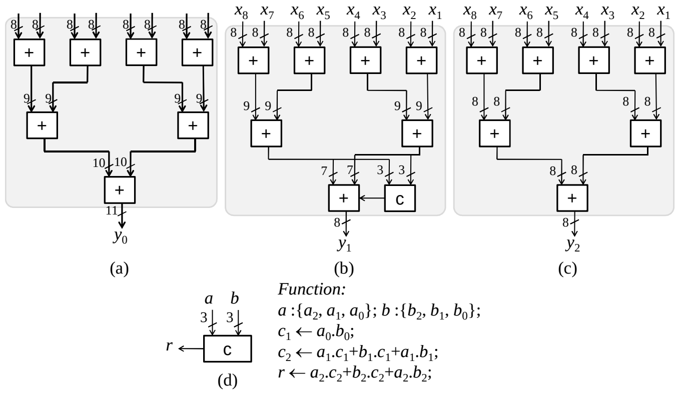
Manuscript submitted on Oct. 20, 2017, revised on Feb. 12, Apr.18, 2018.

B. K. Mohanty is with the Dept. of Electronics and Telecomn. Engineering, Mukesh Patel School of Technology Management and Engineering, SVKM's NMIMS (Deemed to be University), Shirpur Campus, Dhule, Maharashtra, India-425405, Email: bk.mohanti@gmail.com

i-th row of bits correspond to i-th component $\{x_i\}$ of input-vector $\mathbf{x}$ and w is the bit-width. The bit-matrix $\mathbf{A}$ for $N = 8$ and $w = 8$ is shown in Fig.1. Bits of $\mathbf{A}$ are added column-wise along with the carry bits to calculate one bit AT output and from w columns of $\mathbf{A}$, w lower order bits of FL-AT output $\{y\}$ are calculated. The remaining p upper bits of $\{y\}$ are calculated from the carry bits generated from the w-th column of $\mathbf{A}$. The structure of FL-AT for $N = 8$ and $w = 8$ is shown in Fig.2(a), where each adder is implemented using a RCA. Structure of FX-AT-PT and FX-AT-DT are shown in Fig.2(b) and Fig.2(c), respectively. It is observed that both FX-AT-PT and FX-AT-DT does not provide an area-delay efficient FX-AT design. To address this issue, a different type of truncation method and error compensation scheme is discussed here.

$$
\begin{array}{ccc|ccccc|ccc}
 & & & & & \text{MSP} & & & & \text{LSP} & \\
10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
\hline
 & & & x_{17} & x_{16} & x_{15} & x_{14} & x_{13} & x_{12} & x_{11} & x_{10} \\
 & & & x_{27} & x_{26} & x_{25} & x_{24} & x_{23} & x_{22} & x_{21} & x_{20} \\
 & & & x_{37} & x_{36} & x_{35} & x_{34} & x_{33} & x_{32} & x_{31} & x_{30} \\
 & & & x_{47} & x_{46} & x_{45} & x_{44} & x_{43} & x_{42} & x_{41} & x_{40} \\
 & & & x_{57} & x_{56} & x_{55} & x_{54} & x_{53} & x_{52} & x_{51} & x_{50} \\
 & & & x_{67} & x_{66} & x_{65} & x_{64} & x_{63} & x_{62} & x_{61} & x_{60} \\
 & & & x_{77} & x_{76} & x_{75} & x_{74} & x_{73} & x_{72} & x_{71} & x_{70} \\
 & & & x_{87} & x_{86} & x_{85} & x_{84} & x_{83} & x_{82} & x_{81} & x_{80} \\
\hline
y_0=\{y_{010} & y_{09} & y_{08} & y_{07} & y_{06} & y_{05} & y_{04} & y_{03} & y_{02} & y_{01} & y_{00}\,\} \\
y_1=\{y_{17} & y_{16} & y_{15} & y_{14} & y_{13} & y_{12} & y_{11} & y_{10} & 0 & 0 & 0\,\}
\end{array}
$$

Fig. 1. Input bit-matrix $\mathbf{A}$ of adder tree for $N = 8$ and $w = 8$. y_0 and y_1, respectively, represents the output of FL-AT and FX-AT-PT.



x_8 x_7 x_6 x_5 x_4 x_3 x_2 x_1
8 8
+
9 9
10 10
11
y_0
7 7 3 3
c
y_1
8 8
y_2
(a)
(b)
(c)
a b
3 3
r
(d)
Function:
a :{a_2, a_1, a_0}; b :{b_2, b_1, b_0};
$c_1 \leftarrow a_0.b_0$;
$c_2 \leftarrow a_1.c_1+b_1.c_1+a_1.b_1$;
$r \leftarrow a_2.c_2+b_2.c_2+a_2.b_2$;

Fig. 2. (a) Full-width adder-tree for $N = 8$ and $w = 8$. (b) Fixed-width post-truncated adder-tree (FX-AT-PT) for $N = 8$ and $w = 8$. (c) Fixed-width with 1-bit direct-truncated adder-tree (FX-AT-DT) for $N = 8$ and $w = 8$. (d) Function of carry-generator used in final stage of FX-AT-PT.

The bit-matrix $\mathbf{A}$ is partitioned into two parts named most significant part (MSP) and least significant part (LSP). The lower p columns of $\mathbf{A}$ forms the LSP and the upper $(w - p)$ columns forms the MSP. LSP columns are partially or fully truncated to design a fix-width AT which is discussed in the following section.

A. Truncated Fixed-width Adder-Tree

The MSP of input bit matrix is only considered to calculate the output of proposed truncated fixed-width AT (TFX-AT). Due to truncation of LSP, a significant amount of error is introduced in the AT output. A fixed bias for the truncated LSP is added to the MSP for error compensation. The fixed-bias is estimated using the probabilistic approach. The probability of each component of LSP ($x_{i,j}$, for $1 \leq i \leq N$ and $0 \leq j \leq p-1$, where $p = \log_2 N$) for binary '0' or '1' is $\left(\frac{1}{2}\right)$. The output of TFX-AT is expressed as:

$$y = MSP + 2^p \cdot \sigma \tag{1}$$

where, the term $(2^p.\sigma)$ corresponds to the LSP and (σ) represents the estimated bias. The expected value of the LSP is calculated using the formula:

$$E[LSP] = N\sum_{i=1}^{p}\left(\frac{1}{2}\right).2^{p-i} = \left(\frac{N}{2}\right)\cdot 2^p(1-2^{-p}) \tag{2}$$

The quantized value of σ is calculated as

$$\sigma = round\Big[\left(\frac{N}{2}\right)(1-2^{-p})\Big] \approx \left(\frac{N}{2}\right) \tag{3}$$

Using (3) the fixed-bias for $N = 8$ and 16 is found to be $\{\sigma$=4 and 8$\}$. The input bit-matrix of TFX-AT with fixed-bias is shown in Fig.3(a) for $N = 8$ and $w = 8$. The binary values of σ=4 is added to the least significant column of MSP for error-compensation. Structure of proposed TFX-AT with fixed-bias is shown in Fig.3(b).

B. Proposed Improved Truncated Fixed-width Adder-Tree

To estimate the bias of the truncated part more precisely the LSP of $\mathbf{A}$ is further partitioned into two parts as major-part and minor-part. The most significant column of LSP constitutes the major-part (MJP) and the remaining $(p - 1)$ lower order columns constitute the minor-part (MNP). The bias in this case is estimated using the MJP and MNP of LSP as:

$$\sigma = \sigma_{major} + \sigma_{minor} \tag{4}$$

where, σ_{major} and σ_{minor} are the estimated bias of MJP and MNP of LSP. σ_{major} is estimated accurately using the actual signal value of MJP where the σ_{minor} is estimated using the probabilistic approach. The quantized value of σ_{major} is estimated using the relation:

$$
\begin{aligned}
E[MJP] &= \Big\lfloor\sum_{i=1}^{N} 2^{p-1}x_{i,p-1}\Big\rfloor = 2^p\Big\lfloor\sum_{i=1}^{N}\left(\frac{x_{i,p-1}}{2}\right)\Big\rfloor \\
\sigma_{major} &= \Big\lfloor\sum_{i=1}^{N}\left(\frac{x_{i,p-1}}{2}\right)\Big\rfloor
\end{aligned} \tag{5}
$$

The quantized value of σ_{minor} is estimated as:

$$E[MNP] = \left(\frac{N}{4}\right)\cdot 2^p(1-2^{-p+1}) \tag{6a}$$

$$\sigma_{minor} = round\Big[\left(\frac{N}{4}\right)(1-2^{-p+1})\Big] \approx \left(\frac{N}{4}\right) \tag{6b}$$

The input bit-matrix of the proposed improved truncated fixed-width adder-tree (ITFX-AT) is shown in Fig.4 for $N = 8$ and $w = 8$. The logic-block corresponding to σ_{major} calculates the carry bits $\{c_0, c_1, c_2, c_3, c_4, c_5, c_6\}$. These carry bits and the fixed-bias corresponding to σ_{minor} are added to the least significant column of MSP. According to (6b), the

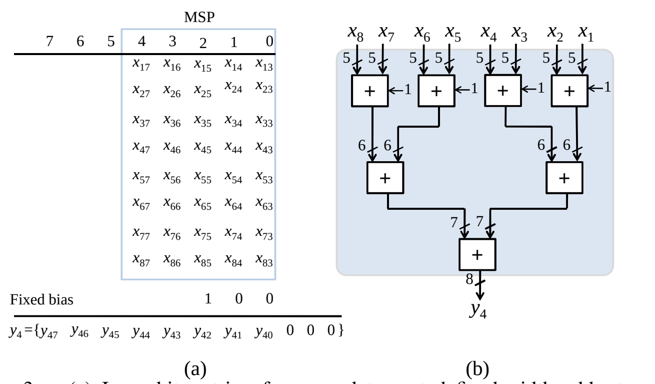


Fig. 3. (a) Input bit-matrix of proposed truncated fixed-width adder-tree (TFX-AT) with fixed-bias for error-compensation. (b) Structure of proposed TFX-AT design

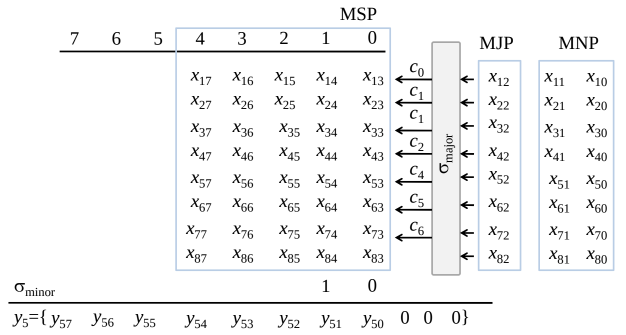


Fig. 4. Input bit-matrix of proposed improved truncated fixed-width adder-tree (ITFX-AT) for $N = 8, w = 8$. MJP and MNP represents the major part and minor part of LSP. $\{c_0, c_1, c_2, c_3, c_4, c_5, c_6\}$ represent the carry bits corresponding to the estimate of σ_{major}.

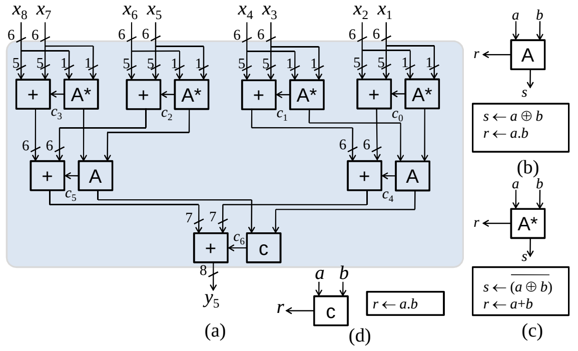


Fig. 5. (a) Structure of proposed ITFX-AT. (b) Logic function of half-adder (A). (c) Logic function of modified half-adder (A*). (d) Logic function of carry cell.

value of σ_{minor} for $N = 8$ is found to be 2. The structure of proposed ITFX-AT is shown in Fig.5(a). The seven half-adders (A) connected in a tree structure calculates the carry bits $\{c_0, c_1, c_2, c_3, c_4, c_5, c_6\}$ corresponding to σ_{major}. Out of these 7 half-adders, 4 half-adders of first tree level are replaced with four full-adders with fixed input-carry 1 to add the fixed-bias $(+4)$ to the MJP of LSP instead of least significant column of MSP. Full-adders with fixed input-carry '1' are further optimized into a modified half-adder (A*) comprising of a XNOR and OR-gate.

Approximate full-adders of [3] are considered to add the LSP of the input matrix for reducing the logic complexity and CPD of the FL-AT. The approximate adder (AXA) is

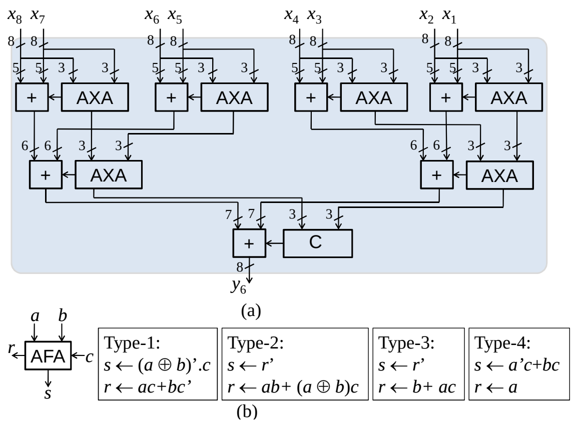


Fig. 6. (a) Structure of approximate fixed-width adder-tree (APX-FX-AT) using accurate RCA and 3-bit approximate (APX) RCA of [3] for $N = 8$ and $w = 8$. (b) Approximate full-adder (AFA) {Type-1, Type-2, Type-3 and Type-4} of [3].

implemented using the approximate full-adder (AFA) {Type-1, Type-2, Type-3 and Type-4} of [3]. Post-truncated approximate FX-AT (APX-FX-AT-PT) is obtained from approximate full-width adder-tree (APX-FL-AT) to study the performance of the proposed FX-AT designs. Note that APX-FL-AT-Type-4 is identical to the APX-AT of [7]. Structure of APX-FX-AT-PT is shown in Fig.6(a) using accurate RCA and 3-bit approximate RCA for $N = 8$ and $w = 8$.

III. Hardware-Time Complexities and Performance Study

Full-width AT calculates the output accurately and involves largest amount of logic complexity and CPD than the fixed-width AT. But, fixed-width AT produces output with some error. It is necessary to study the error performance of fixed-width AT designs along with their hardware and time complexities.

A. Error Performance

To study the error performance, we have coded the proposed fix-width AT designs (TFX-AT and ITFX-AT), existing APX-FX-AT-PT using approximate adders (Type-1, Type-2, Type-3 and Type-4 of [3]) of size 3-bit, conventional FX-AT-DT and FX-AT-PT in VHDL for $N = 8, 16$ and $w = 8, 12, 16$. Test-bench output are produced for 10,000 different sets of randomly generated 8-word and 16-word test vectors. The full-width AT is considered as the reference design to estimate the error. The {maximum error distance (MED), average error distance (AED), average accuracy (AAC)}[1] of proposed designs and existing designs are estimated and the values are listed in Table I. AED, MED and AAC changes marginally for higher word-length across all designs. In general, the error metric of all designs is almost independent of word-length size and largely depends on the input vector size. The FX-AT-PT has the lowest AED and MED among the existing

[1]MED = $\max(abs(y_0 - y_i))$, AED = $avg(abs(y_0 - y_i))$, AAC = $avg\left[\left(1 - \left(\frac{abs(ED)}{y_0}\right)\right) \times 100\right]$ [4], where, y_0 is the output of full-width AT and y_i is the output of fixed-width AT

TABLE I
ERROR ESTIMATES OF PROPOSED DESIGNS AND EXISTING DESIGNS

Designs	N	word-length $w = 8$			word-length $w = 12$			word-length $w = 16$		
		AED	MED	AAC (%)	AED	MED	AAC (%)	AED	MED	AAC (%)
FX-AT-PT	8	3.4510	7	99.6630	3.5068	7	99.9786	3.5082	7	99.9987
	16	7.2784	15	99.6446	7.5025	15	99.9771	7.4844	15	99.9986
FX-AT-DT	8	5.9686	12	99.4171	5.9821	12	99.9635	6.0149	12	99.9977
	16	16.2039	31	99.2088	16.0422	31	99.9511	16.0774	31	99.9969
APX-FX-AT-PT [3] (Type-1)	8	4.3118	18	99.5789	4.3542	23	99.9734	4.3516	19	99.9983
	16	8.002	36	99.6093	8.3669	39	99.9745	8.2562	35	99.9984
APX-FX-AT-PT [3] (Type-2)	8	4.3314	17	99.5770	4.4099	22	99.9731	4.4198	22	99.9983
	16	6.8510	33	99.6655	6.8241	33	99.9792	6.8774	33	99.9987
APX-FX-AT-PT [3] (Type-3)	8	4.6784	21	99.5431	4.8950	27	99.9701	4.8950	23	99.9982
	16	7.2882	27	99.6441	7.4220	35	99.9774	7.4505	36	99.9986
APX-FX-AT-PT [7] (Type-4) [3]	8	6.4341	21	99.3719	6.4238	26	99.9601	6.3927	29	99.9976
	16	12.2176	48	99.4034	12.4004	44	99.9622	12.3226	42	99.9977
Proposed TFX-AT	8	6.0176	22	99.4123	6.1677	26	99.9624	6.1167	26	99.9977
	16	16.3020	64	99.2040	16.1200	76	99.9508	16.1087	71	99.9969
Proposed ITFX-AT	8	3.3529	13	99.6726	3.4172	15	99.9792	3.4012	15	99.9987
	16	8.7078	34	99.5748	8.6520	40	99.9736	8.7064	40	99.9983

TABLE II
COMPARISON OF PSNR OF RECONSTRUCTED IMAGES OBTAINED THROUGH FORWARD AND INVERSE WALSH HADAMARD TRANSFORM IMPLEMENTED USING PROPOSED AND EXISTING ADDER-TREE DESIGNS: ($w = 8$ AND $N = 8$)

Designs	**Low-texture image PSNR (dB)**			**Moderate-texture image PSNR (dB)**			**Higher-texture image PSNR (dB)**		
	Lenna [8]	Barbara [8]	Grass bed [9]	Pepper [8]	Daisies flower [10]	Madison square [11]	Red pepper [12]	Flower bed-1 [13]	Flower bed-2 [13]
FX-AT-PT	102.21	101.36	101.37	89.49	63.21	55.97	80.76	62.5	73.4
FX-AT-DT	93.02	92.36	81.72	66.57	56.11	51.24	65.65	55.88	62.93
APX-FX-AT-PT-Type-1 [3]	93.02	92.36	81.72	66.57	56.11	51.24	65.65	55.88	62.93
APX-FX-AT-PT-Type-2 [3]	87.12	87.21	79.10	65.85	52.56	45.65	65.02	52.99	61.53
APX-FX-AT-PT-Type-3 [3]	87.12	87.21	79.10	65.85	52.56	45.65	65.02	52.99	61.53
APX-FX-AT-PT-Type-4 [3]	93.02	92.36	81.72	66.57	56.11	51.24	65.65	55.88	62.93
Proposed TFX-AT	76.96	76.01	76.23	75.84	53.96	47.05	77.20	59.49	64.11
Proposed ITFX-AT	90.25	89.14	89.35	88.87	60.01	53.44	90.17	64.11	77.42

TABLE III
COMPARISON OF SYNTHESIS RESULTS

Designs	N	**word-length** $w = 8$				**word-length** $w = 12$				**word-length** $w = 16$			
		CPD (ns)	**Area** (um^2)	**Power** (uW)	**ADP** (um^2.us)	**CPD** (ns)	**Area** (um^2)	**Power** (uW)	**ADP** (um^2.us)	**CPD** (ns)	**Area** (um^2)	**Power** (uW)	**ADP** (um^2.us)
FX-AT-PT	8	0.88	1333	379	1173	0.95	2093	571	1988	1.05	3082	733	3236
	16	1.07	2938	701	3144	1.21	4330	979	5239	1.31	6289	1271	8238
FX-AT-DT	8	0.85	1242	320	1056	0.89	2060	541	1833	1.01	2923	668	2952
	16	1.01	2707	616	2734	1.15	4053	887	4661	1.27	6125	1215	7779
APX-FX-AT-PT Type-1 [3]	8	0.87	1217	295	1059	0.99	1895	453	1876	1.09	2786	584	3036
	16	1.13	2350	440	2656	1.25	3907	822	4883	1.39	5679	986	7891
APX-FX-AT-PT Type-2 [3]	8	0.93	1080	254	1004	0.99	1843	479	1824	1.09	2943	639	3207
	16	1.13	2271	439	2566	1.25	3862	736	4827	1.37	5626	1004	7707
APX-FX-AT-PT Type-3 [3]	8	0.85	1004	264	853	0.97	1754	427	1701	1.07	2558	562	2737
	16	1.09	1881	399	2050	1.23	3409	694	4193	1.31	5366	1021	7029
APX-FX-AT-PT Type-4 [3], [7]	8	0.75	933	282	700	0.83	1746	493	1449	0.97	2616	651	2537
	16	0.89	1765	461	1571	1.07	3324	783	3556	1.17	5025	1095	5879
Proposed TFX-AT	8	0.73	860	257	628	0.83	1612	453	1338	0.97	2373	571	2301
	16	0.87	1444	397	1256	1.05	2994	715	3147	1.15	4742	1055	5453
Proposed ITFX-AT	8	0.75	984	295	738	0.85	1800	497	1530	0.99	2548	593	2522
	16	0.93	1650	448	1535	1.07	3427	837	3667	1.19	5040	1089	5997

designs whereas FX-AT-DT and APX-FX-AT-Type-4 has the highest AED and lowest AAC. The proposed TFX-AT has marginally higher AED than the FX-AT-DT and APX-FX-AT-Type-4. The proposed ITFX-AT has marginally lower/higher AED and nearly the same AAC when compared with FX-AT-PT for different word-length and vector sizes, but it has higher MED. This is mainly due to the overcompensation of σ_{minor} by the fixed-bias.

To further study the error performance of proposed and the existing AT designs on image processing application, we have considered 8-point forward and inverse Walsh-Hadamard transform (WHT) for processing images. We have coded 8-point forward and inverse WHT in VHDL using the proposed and existing AT designs. Considered images with different color and edge information. These images are grouped as low-texture, moderate-texture and higher-texture images for discussion purpose. Less color variation with less edge information refer to low-texture image, less color variation with more edge information refer to moderate-texture image, and more color variation with more edge information refer to higher-texture image. The pixel variation is different in each 8-point input data-vector of higher-texture, moderate-texture and low-texture images. Pixel variation is more in data-vectors of higher-texture images, relatively less in data-vectors of moderate-texture images and almost absent/small in data-vectors of low-texture images. The proposed TFX-AT and ITFX-AT designs use a fixed-bias for error-compensation. The fixed-bias is estimated using probabilistic approach. The fixed-bias compensates truncation error near accurately for input data-vector with more pixel variation while overcompensate the truncation error for input data-vector with small pixel variation. The existing FX-AT-PT design does not have this feature and its performance entirely depends on the magnitude of post-truncation error irrespective of pixel variation of input-vector. To observe this behavior, we have estimated PSNR of reconstructed image of WHT and the estimated values are listed in Table II. As shown in Table II, WHT based on the proposed design offers slightly degraded performance for low-texture images compared to those obtained using FX-AT-PT design mainly due to more overcompensation instances. However, WHT based on the proposed ITFX-AT reconstruct higher-texture images with higher PSNR and moderate-texture images with almost the same PSNR compared to those obtained using FX-AT-PT due to few overcompensation instances and more near accurate compensation instances.

B. Comparison of Synthesis Result

All the designs are synthesized in Synopsys Design Compiler using TSMC 65nm CMOS standard cell library. We have considered input vector-sizes $N = 8, 16$ and word-lengths $w = 8, 12, 16$. Area, CPD and power reported by the design compiler are listed in Table III for comparison. As shown in Table III, existing APX-FX-AT (Type-1 and Type-2) designs involve marginally less area and less CPD than the FX-AT-PT and calculate the output with marginally lower accuracy. Compared with FX-AT-PT, proposed ITFX-AT offers higher area and delay saving for higher size input-vectors. We have estimated ADP[2] of all the designs, and the estimated values are also listed in Table III. For word-length $(8, 12, 16)$, the proposed ITFX-AT offers $(37\%, 23\%, 22\%)$ and $(51\%, 30\%, 27\%)$ ADP saving over the FX-AT-PT for $N = 8$ and 16, respectively, and calculate outputs with almost the same accuracy of FX-AT-PT.

IV. Conclusion

In this paper, a novel scheme is presented to obtain fixed-width AT design using truncated input. A bias estimation formula based on probabilistic approach is presented to compensate the truncation error. Based on the proposed scheme, two separate fixed-width AT designs are derived. Both the proposed designs offer a substantial amount of area and CPD saving over the existing fixed-width AT designs. For vector-sizes 8 and 16, the proposed ITFX-AT offers $(37\%, 23\%, 22\%)$ and $(51\%, 30\%, 27\%)$ ADP saving for word-length sizes $(8, 12, 16)$, respectively, and calculates the output almost with the same accuracy as the post-truncated fixed-width AT which has the highest accuracy among the existing fixed-width AT. Further, we observed that Walsh-Hadamard transform based on proposed adder design reconstruct higher-texture images with higher PSNR and moderate-texture images with almost the same PSNR compared to those obtained from the existing fixed-width adder designs. Besides, the use of truncated input samples in fixed-width AT design is an interesting feature which creates an additional advantage to optimize other blocks appear at the upstream of the AT in a complex design.

References

[1] B. K. Mohanty and V. Tiwari, ”Modified probabilistic estimation bias formulation for hardware efficient fixed-width Booth multiplier”, *Circuits, Systems and Signal Processing, Springer*, vol.33, no.12, pp. 3981–3994, Dec., 2014,

[2] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie and C. Lucas, ”Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications”, *IEEE Transactions on Circuits and Systems-I, Regular Papers*, vol. 57, no. 4, pp. 850–862, Apr.2010.

[3] V. Gupta, D. Mahapatra, A. Raghunathan, and K. Roy, ”Low-power digital signal processing using approximate adders”, *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 32, no. 1, pp. 124–137, Jan.2013.

[4] J. Liang, J. Han and F. Lombardi, ”New metrics for the reliability of approximation and probabilistic adders”, *IEEE Transactions on Computers*, vol. 62, no. 9, pp. 1760–1771, Sept.2013.

[5] H. Jiang, J. Han, F. Qiao and F. Lambardi, ”Approximate radix-8 Booth multipliers for low-power and high-performance operations”, *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2638–2644, Aug.2016.

[6] Y. Pan and P. K. Meher, ”Bit-level optimization of adder-trees for multiplie constant multiplications for effcient FIR filter implementation”, *IEEE Transactions on Circuits and Systems-I, Regular Papers*, vol. 61, no. 2, pp. 455–462, Feb.2014.

[7] R. O. Julio, L. B. Soares, E. A. C. Costa and S. Bampi, ”Energy-efficient Gaussian filter for image processing using approximate adder”, *In. Proc. International Conference on Electronics, Circuits and Systems, 2015*, pp. 450-453, 2015.

[8] https://homepages.cae.wisc.edu/ ece533/images/

[9] http://seamless-pixels.blogspot.in/2014/07/grass-2-turf-lawn-green-ground-field.html

[10] https://www.decoist.com/2013-02-28/flower-beds/

[11] http://www.cosasexclusivas.com/2014/06/daily-overview-el-planeta-tierra-visto.html

[12] https://healthtipsfr.blogspot.com/2017/04/blog-post_40.html

[13] http://www.ugaoo.com/knowledge-center/how-to-design-a-flower-bed/

[2]ADP=area$\times$ CPD