Energy-Quality Scalable Adders Based on Nonzeroing Bit Truncation

Fabio Frustaci, Stefania Perri, Pasquale Corsonello, and Massimo Alioto

***Abstract*— Approximate addition is a technique to trade off energy consumption and output quality in error-tolerant applications. In prior art, bit truncation has been explored as a lever to dynamically trade off energy and quality. In this brief, an innovative bit truncation strategy is proposed to achieve more graceful quality degradation compared to state-of-the-art truncation schemes. This translates into energy reduction at a given quality target. When applied to a ripple-carry adder, the proposed bit truncation approach improves quality by up to 8.5 dB in terms of peak signal-to-noise ratio, compared to traditional bit truncation. As a case study, the proposed approach was applied to a discrete cosine transform engine. In comparison with prior art, the proposed approach reduces energy by 20%, at insignificant delay and silicon area overhead.**

***Index Terms*— Adaptive precision, approximate computing, energy-quality scaling, error-tolerant systems, low-power design, VLSI.**

I. Introduction

Energy-quality scaling is emerging as a paradigm to dynamically reduce the consumption in applications that can naturally tolerate inaccuracies, such as multimedia, machine learning, digital signal processing, and wireless communications [1]–[6]. In this general area, approximate addition has been widely explored, and several approximate full adders have been proposed to approximate the least significant output bits (LSBs) [7]–[10]. Unfortunately, these techniques require the design of specialized standard cells, and their accuracy is statically set at design time, which is a significant limitation since it does not allow dynamic energy-quality scaling [3], [12]. Indeed, static assignment of the accuracy may either fail to meet higher output quality when temporarily required, or unnecessarily increase the energy in the common case. For this reason, adders should be employable in automated frameworks where the level of approximation is dynamically tuned based on the user demand [13], [14]. In [15], a dynamic energy-quality tradeoff was achieved by tuning the number of truncated LSBs at run time. In particular, a bit truncation scheme was introduced to inhibit the activity of the LSBs by zeroing the corresponding input bits. However, this approach was shown to suffer from ungraceful quality degradation at larger number of truncated bits [8], [11].

In this brief, a novel bit truncation approach is proposed to design dynamically energy-quality scalable adders with graceful degradation. As main idea, instead of being set to zero, the truncated input bits are set to the constant value that maximizes the output quality, while maintaining the same energy. The approach is validated in several adders in 28-nm fully depleted silicon-on-insulator (FDSOI) technology. Compared with conventional zeroing bit truncation [15], the proposed approach is shown to improve quality in terms of peak signal-to-noise ratio (PSNR) and mean error distance (MED) by up to 8.5 dB and 67%, respectively. More favorable energy-quality tradeoff is also observed over static approximate adders [7]–[10], with up to 64% energy reduction at isoquality when scaling down the voltage. As a case study, the proposed approach was also validated in a discrete cosine transform (DCT) engine, whose PSNR is shown to be improved by up to 6 dB over existing approaches at iso-energy.

This brief is organized as follows. Section II introduces the proposed bit truncation approach. Comparison with the existing approximate adders is presented in Section III. Section IV presents an approximate DCT engine as a case study. Finally, conclusions are drawn in Section V.

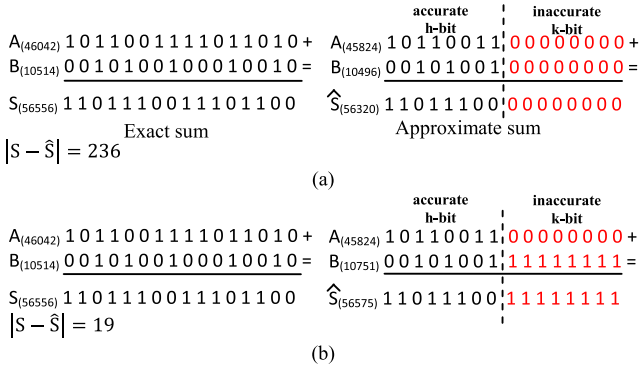


Fig. 1. Approximate addition. (a) Traditional bit truncation [15]. (b) Example of the proposed bit-truncation approach (among many equivalent configurations).

II. Proposed Bit Truncation Scheme

Approximate n-bit addition is generally performed by splitting it into an h-bit accurate and a k-bit inaccurate part, with $n = h + k$. In line with prior art, ripple-carry adders (RCAs) will be considered in the following, in view of their low energy consumption. The example in Fig. 1(a) explains how two unsigned n-bit operands are truncated by zeroing their k LSBs [15] for $n = 16$ and $k = 8$. Fig. 1 also illustrates how the approximate sum $\hat{S}$ differs from the exact sum S. The resulting output error $\hat{S} - S$ depends on k, and is equal to the sum of the errors associated with the two operands. When the operands are truncated by zeroing their k LSBs as in [15], the error in each operand is invariably nonpositive and ranges from $-(2^k - 1)$ to 0, and hence has negative mean value. Instead, a zero-mean error would be desirable since this would allow the compensation of errors with opposite sign. Among the possible choices of bit truncation schemes, the one that maximizes quality is derived in the following, using the PSNR as common quality metric [10].

Let us consider the addition of two unsigned n-bit operands $A = A_{n-1} \ldots A_0$ and $B = B_{n-1} \ldots B_0$, and the generic bit truncation scheme that sets k LSBs of the operands to nonzero constant values. The resulting error in A and B is, respectively, in the $[x, x+m]$ and $[y, y+m]$, with $m = 2^k - 1$ and $x, y \epsilon [-m; 0]$. The value of x and

Manuscript received July 9, 2018; revised October 8, 2018; accepted November 9, 2018. This work was supported by the Singapore Ministry of Education under Grant MOE2014-T2-1-161. *(Corresponding author: Pasquale Corsonello.)*

F. Frustaci and P. Corsonello are with the DIMES Department, University of Calabria, 87036 Rende, Italy (e-mail: f.frustaci@dimes.unical.it; p.corsonello@unical.it).

S. Perri is with the DIMEG Department, University of Calabria, 87036 Rende, Italy (e-mail: stefania.perri@unical.it).

M. Alioto is with the ECE Department, National University of Singapore, Singapore 117583 (e-mail: massimo.alioto@nus.edu.sg).

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

Digital Object Identifier 10.1109/TVLSI.2018.2881326

TABLE I
COMPARISON RESULTS IN TERMS OF PSNR AND MED

		PSNR			MED		
n	k	[15] (dB)	Proposed truncation (dB)	Δ_{PSNR} (dB)	[15] (a.u.)	Proposed truncation (a.u.)	Δ_{MED} (%)
32	16	101.7	110.1	+8.4	65535	21845	-66.7
	12	125.7	134.2	+8.5	4095	1365.3	-66.6
	8	149.8	158.3	+8.5	255	85.3	-66.5
16	8	53.54	61.7	+8.16	255	85.3	-66.5
	6	65.6	74	+8.4	63	21.3	-66.2
	4	78.1	86	+7.9	15	5.3	-64.7
8	4	29.9	37.9	+8	15	5.3	-64.7
	3	36.4	43.9	+7.5	7	2.6	-62.8
	2	43.5	50.2	+6.7	3	1.2	-60

y are equal to $A_{[k-1:0]} - m$ and $B_{[k-1:0]} - m$, respectively, being $A_{[k-1:0]}$ ($B_{[k-1:0]}$) the numerical value chosen to truncate the k LSBs of the operand A (B). The accurate portion of A and B (i.e., the first $n-k$ bits) can assume 2^h values, i.e., $0, 2^k, 2^k \cdot 2, \ldots, 2^k \cdot (2^h - 1)$. Considering that 2^{2n} distinct additions can be performed on A and B, the sum of the squared errors (sse) that is introduced in the output sum by the truncation in the k LSBS of A and B is

$$\text{sse} = \sum_{j=0}^{2^{2n}-1} |S_j - \hat{S}_j|^2. \tag{1}$$

Since each possible output error value across all possible input combinations occurs $2^{2 \cdot h}$ times,[1] and assuming that the k-bit LSB subwords in the operands are uniformly distributed, (1) can be rewritten as

$$\text{sse} = 2^{2h} \times \sum_{i=x}^{x+m} \left(\sum_{j=y}^{y+m} (i+j)^2 \right). \tag{2}$$

By definition, the PSNR is given by (3), where MAX is the maximum value of the sum [i.e., $2 \times (2^n - 1)$ for n-bit unsigned operands]

$$\text{PSNR} = 20 \log \frac{\text{MAX}}{\sqrt{\frac{\text{sse}}{2^{2n}}}}. \tag{3}$$

By equating the derivative of (2) and (3) with respect to x and y to zero, the minimum value of sse and the maximum PSNR are readily found to occur when $x + y = -m$. All (x, y) pairs satisfying the latter condition lead to the minimum error and hence best quality. From the definition of x and y, it follows that the optimum condition is equivalent to: $A_{[k-1:0]} = m - B_{[k-1:0]}$. Since the binary representation of m is a k-bit sequence of 1s, the optimum condition translates into the requirement that the ith LSBs of A and B(with $i < k$) are statically set to complementary values, i.e., $A_i = \bar{B}_i$.

As an example, the choice in Fig. 1(b) has $x = -m$ (i.e., k LSBs in A set to 0...0) and $y = 0$ (i.e., k LSBs in B set to 1...1), and hence satisfies the above condition, thus maximizing quality for a given k (i.e., energy).

On the contrary, the traditional bit truncation strategy in [15] leads to $x = y = -m$, which is clearly non-optimal in terms of quality. Detailed results are reported in Table I, which shows that the quality improvement achieved by the proposed approach over [15] ranges between 6.7 and 8.5 dB. A comparison in terms of the MED quality metric [10] is also reported in Table I. The proposed approach reduces MED by 60%–67% over [15] at same k, thus confirming its effectiveness. Similar analysis can be repeated for signed operands, with results being essentially the same as those obtained for unsigned numbers.

[1]Indeed, the number of possible pairs of truncated input operands (A, B) having the same error $(\text{err}_A, \text{err}_B)$ is $2^{2 \cdot h}$.

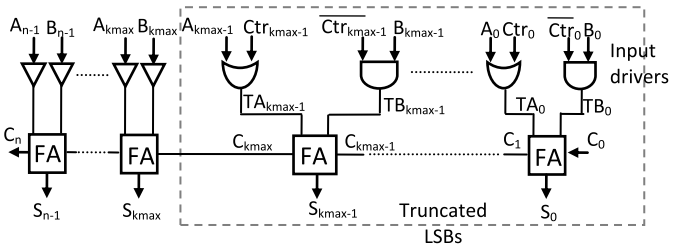


Fig. 2. Approximate RCA implementing the proposed bit truncation scheme.

The above bit truncation scheme can be easily implemented at circuit level in an n-bit RCA as shown in Fig. 2. No special cells are needed, as conventional full adders are used for all bit positions. OR and AND gates are employed as input buffers driving the full adders that receive $k_{\max}$ LSBs of the operands, being $k_{\max}$ the maximum number of LSBs that can be truncated. The control signals $\text{Ctr}_{\text{kmax}-1} \ldots \text{Ctr}_0$ allow dynamic adaptation of k (with $k < k_{\max}$), and hence of the energy-quality tradeoff at run-time. If signals Ctr_i (with $0 \leq i < k$) are set to 1, the inputs TA_i and TB_i of the ith full adder are, respectively, 0 and 1, thus making its carry output equal to $C_i = 0$. Once the truncation is applied, switching activity and dynamic energy are suppressed in all k full adders associated with the LSBs, regardless of the constant value used in the k LSBs.

In general, the above results need to be modified when considering different operations, such as subtraction. Indeed, the difference $A - B$ is computed by first evaluating the 2s complement of the subtrahend B, and then by adding it to the minuend A. For the subtraction operation, the output error E_{diff} is the difference between the errors E_A and E_B due to the truncation of A and B, respectively. Consequently, E_{diff} can be reduced by ensuring that E_A and E_B have the same sign, instead of being opposite as in the addition operation. More precisely, the quality of the output of an approximate subtractor is maximized by truncating k LSBs of the two operands by setting $A_i = B_i$ (with $0 \leq i < k$), by simple extension of the above results. It is worth noting that the traditional zeroing bit truncation scheme automatically satisfies such a criterion, as it sets $A_i = B_i = 0$ for $0 \leq i < k$. Hence, the bit truncation strategy in [15] is preferable when performing subtractions, not additions. When multiplication is considered, the energy of an approximate multiplier was found to strongly depend on the constant values chosen to truncate the k LSBs of the inputs. In other words, two different truncation schemes can lead to the same accuracy but to different energy consumptions, which makes the analysis much more complex. This will be analyzed in the future work.

III. ENERGY-QUALITY TRADEOFF AND SIMULATION RESULTS

The above bit truncation approach for addition was applied to a 16-bit approximate RCA with k ranging from 0 to 8. The adder structured as in Fig. 2 was synthesized with Cadence RTL Compiler, using a commercial 28-nm FDSOI standard cell library with regular-Vth transistors. Several adder configurations were examined, each of which was excited with 10 000 consecutive random additions. Cadence UltraSim circuit simulations were run to derive the average energy consumption per addition. During simulations, the number of truncated LSBs was dynamically configured by properly setting k and the Ctr_i signals. The energy contributions of the RCA and its input buffers were split by using two separate 1-V supply voltages. The 2-GHz operating frequency was targeted, and the adder outputs

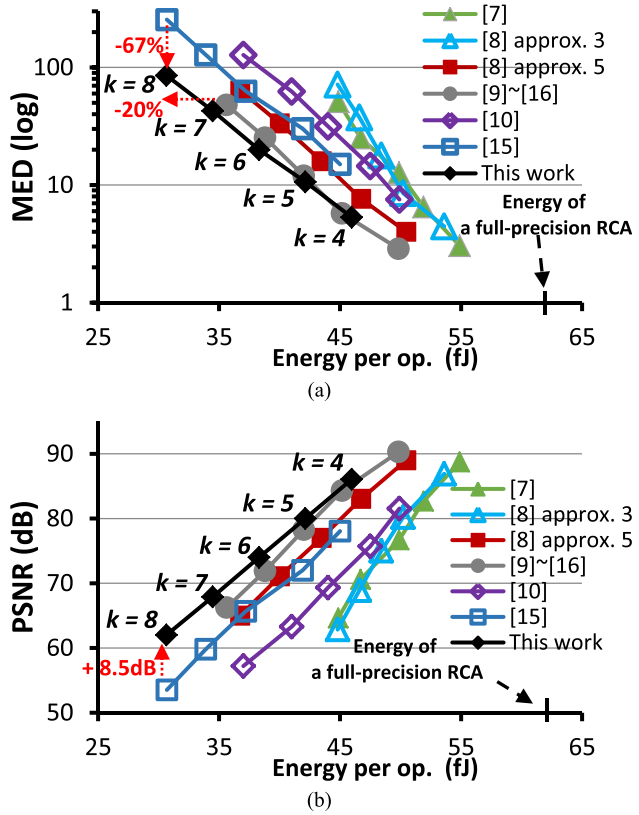


Fig. 3. Energy-quality tradeoff at fixed voltage (V_{DD} = 1 V). (a) MED versus energy. (b) PSNR versus energy.

were loaded with positive-edge triggered D flip-flops with minimum drive strength.

The proposed adder was compared to the existing approximate adders proposed in [7]–[10], [15], which were designed in the same technology. For the adder in [8], the two approximations Approximation 3 and Approximation 5 were considered. Fig. 3 depicts the energy-quality tradeoff achieved by the above 16-bit adders with k ranging from 4 to 8. Fig. 3(a) and (b) shows that the proposed bit truncation strategy improves quality by 67% in terms of MED and 8.5 dB in terms of PSNR, at iso-energy compared to [15]. Fig. 3(a) and (b) also shows that the quality improvement over [7] and [8] (Approximation 3) is even larger, with the PSNR being improved by more than 20 dB, and the MED being reduced by up to 86%. Only the technique [9] and its improved version in [16] achieve a quality that is comparable with the proposed adder, as shown in Fig. 3. However, [9] and [16] cannot be dynamically adjusted at run time, as opposed to the proposed approach. Moreover, the adders proposed in [9] and [16] dissipate up to 20% and 10% more energy than the proposed adder, respectively, at same k.

It is worth noting that the conventional zeroing bit truncation scheme could be implemented by statically setting the k LSBs of the sum to 0 to save k FAs and $2k$ input buffers, as opposed to zeroing the inputs. However, this choice would not provide any benefit with respect to the proposed approach in terms of both energy consumption and accuracy. In fact, such static zeroing bit truncation would clearly achieve the same accuracy and nearly the same energy as [15]. Similar considerations apply to [7]–[10], as static zeroing bit truncation at the output would lead to the same energy and quality in [7]–[10], while making dynamic configuration impossible. Table II summarizes delay and energy contribution of the input buffers and the addition circuits for different values of k. From Table II, the proposed adder consumes

TABLE II
COMPARISON RESULTS

k	Technique	Delay (ps)	Energy (fJ)		Area (um^2)	Avg. error (%)
			Buffers	Adder		
8	[7]	236	33.1	44.9	54.8	51.1
	[8] (appr. #3)	356	33.1	44.8	51.21	-1.06
	[8] (appr. #5)	250	25.9	36.7	37.53	-0.45
	[9]	251	33.1	35.6	42.01	-0.214
	[10]	236	33.1	37.1	55.74	127.45
	[15]	238	17.1	30.7	60.05*	254.9
	[16]	251	19.7	33.3	33.77	-7.98
	This work	240	17.6	30.7	60.05*	-0.001
4	[7]	330	33.1	54.9	56.12	3.02
	[8] (appr. #3)	394	33.1	53.6	54.32	-0.94
	[8] (appr. #5)	345	29.9	50.4	47.97	-0.498
	[9]	346	33.1	49.8	49.81	-0.246
	[10]	330	33.1	49.9	56.59	14.5
	[15]	335	25.9	45.1	60.05*	14.98
	[16]	346	28.6	49.1	48.14	-0.51
	This work	336	26.3	45.9	60.05*	-0.014
0	[15]	437	34.6	63.2	60.05*	0
	This work	438	34.9	63.5	60.05*	0
	Exact Adder	433	33.1	62.6	57.43	0

* Obtained under k_{max} = 8

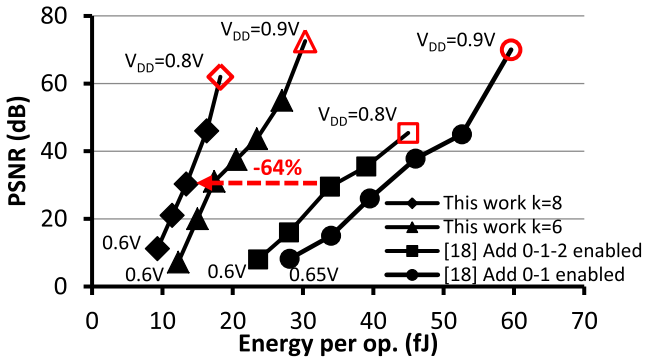
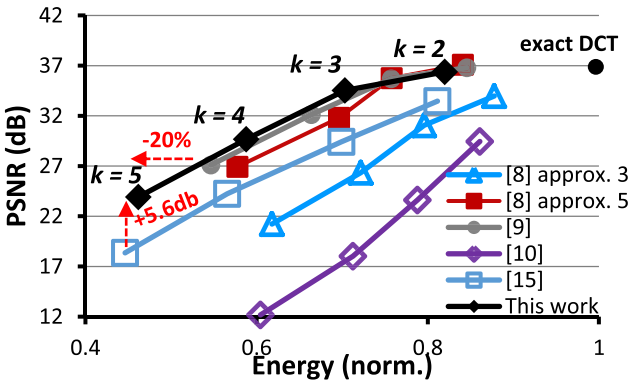
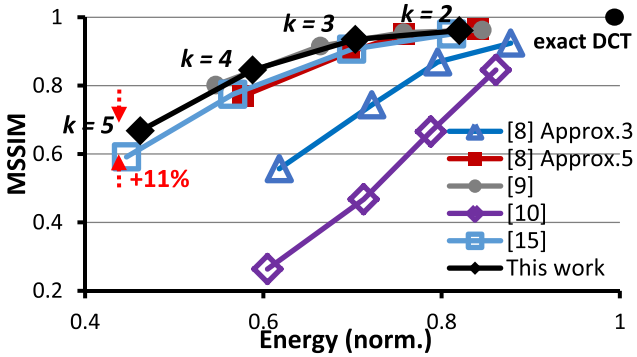


Fig. 4. Comparison of the proposed bit truncation scheme and the reconfigurable approximate RCA in [18], when voltage scaling is applied. The red dots refer to operation with no timing failures (no V_{DD} overscaling).

2X less energy than the exact 16-bit adder. The same energy reduction is achieved by [15], although at the cost of an 8.5-dB additional PSNR degradation. Table II also shows that the proposed bit truncation strategy and [15] permit to save the energy due to the buffers that are made inactive when k LSBs are truncated. On the contrary, all the input buffers used within the static approximate adders [7]–[10], [16] are always active, thus constantly dissipating dynamic energy. It is also worth noting that the OR and AND gates used in the proposed circuit as input buffers of the k_{max} LSBs introduce a negligible delay penalty (about 1%). Conversely, they cause a 4. The 5% area increases over the standard exact adder. The most area efficient approximate adder is the Approximation 5 [8], which exhibits a 34.6% lower area than the exact adder for k = 8. Finally, Table II shows that the proposed technique achieves the lowest average error. Very similar results were obtained for addition of 16-bit signed operands and 16-bit subtraction; hence, the related data are omitted accordingly. The proposed adder was also compared to the reconfigurable 16-bit adder in [18]. The latter dynamically adjusts the output quality and the energy by partitioning the adder into four subadders that can be adaptively disabled to shorten the critical path. This enables opportunities to scale down the supply voltage V_{DD} and hence energy, for a given frequency target. Fig. 4 depicts the energy-quality tradeoff obtained with different configurations and supply voltages, which was



(a)



(b)

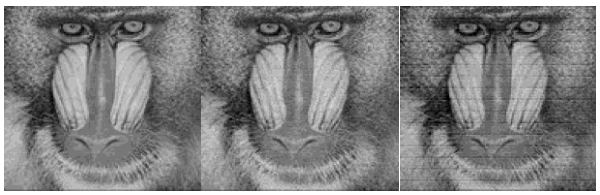
Fig. 5. Energy-quality tradeoff in DCT engine under different approximate adders. (a) PSNR and (b) MSSIM are evaluated as average across 64×64 image benchmarks.

quantified by using the Cadence UltraSim circuit simulator. In each curve, the highest voltage is the minimum that guarantees the 2-GHz frequency target. At lower V_{DD}, energy is reduced at the cost of occasional timing violations when the critical path is excited, thus further degrading the quality. Approximate outputs were compared to the exact results to generate the PSNR curves in Fig. 4. From Fig. 4, the proposed bit truncation scheme invariably consumes less energy than [18] at isovoltage. The energy saving is more pronounced at larger k, and grows up to 64% at PSNR $= 30$ dB (i.e., $k = 8$), corresponding to the low end of the range of acceptable quality in vision systems [6].

IV. Discrete Cosine Transform as a Case Study

To assess its effectiveness in more complex designs, the proposed bit truncation scheme was embedded in a complete DCT architecture processing 8×8 blocks of pixels. DCT is a fundamental operation in static and dynamic image compression [19], and is hence relevant to error-tolerant applications. DCT makes use of additions, subtractions, and multiplications. The latter were replaced by more energy-efficient balanced adder tree architectures, as described in [15]. The chosen DCT application allows estimating the effect of error accumulation across a series of approximate adders for each of the analyzed bit truncation schemes, which is expected to be essentially the same as a single adder.

The DCT engine was designed with Cadence RTL Compiler using the above technology. UltraSim circuit simulations were performed to evaluate the energy while processing four 8-bit grayscale benchmark images taken from the public database in [20]. To evaluate the quality in terms of PSNR, the exact inverse DCT was implemented in MATLAB, decompressing the results produced by DCT and



no approx.	proposed with k=4	[15] with k=4
PSNR=37.96dB	PSNR=30.41dB	PSNR=24.27dB

Fig. 6. Output images obtained from the DCT engines.

comparing them to the uncompressed image. Energy and PSNR were evaluated as the average across four 64×64 image benchmarks (Lena, Baseball, Tennis, Football), and the results are plotted in Fig. 5(a). Fig. 5(a) shows that the proposed bit truncation scheme improves PSNR by up to 5.6 dB compared to conventional zeroing bit truncation scheme [15], at same k. This improvement comes at no energy penalty.

Among the existing techniques with static accuracy, [9] is the most energy efficient as its energy-quality curve lies to the left of the other curves in Fig. 5(a). On the other hand, when compared to the proposed design, the approximate DCT designed with the technique [9] suffers from up to 20% larger energy consumption toward lower quality targets. From Fig. 5(a), the proposed scheme is able to cover a wider range of energy-quality tradeoffs. Most importantly, [9] is not able to dynamically adapt to the quality target. Hence, its design needs to constantly achieve the maximum required quality, which translates into energy penalties when the quality target can be reduced. Finally, it is interesting to observe that the proposed adder is also able to reduce the DCT energy by 2.2X, compared to a DCT designed with exact 16-bit adders. Similar energy reduction is achieved by [15], although at the cost of up to 6.5-dB PSNR degradation. As expected, the DCT engine exploiting the proposed scheme maintains essentially the same energy-quality benefits as the single adder. This confirms that the proposed technique consistently retains its effectiveness when applied to a series of addition operations, as occurs in DCT. The same results were obtained for four other 128×128 image benchmarks (Airplane, SmallGirl, Watchcolor and Baboon). As an example, Fig. 6 visually compares the image Baboon obtained with the proposed and traditional [15] bit truncation techniques for $k = 4$. The DCT architecture was also tested to compress the video benchmark backdoor [21] targeting pedestrian recognition. The quality adaptation of the DCT engine was explored by configuring the adders at higher accuracy ($k = 2$) in frames where motion was detected, and lower accuracy ($k = 5$) otherwise. Conversely, k was statically set to three for the DCT engines under the approximations in [9] and [16], to match the quality achieved by the proposed DCT engine at $k = 2$ for fair comparison. The results obtained for the complete 2000-frame video sequence show that the proposed technique reduces energy by 20% compared to [9] and [16], thanks to dynamic reconfiguration.

For completeness, the output quality of DCT was evaluated also in terms of the Mean Structural Similarity (MSSIM) metric [22], which is relevant to the applications where DCT is employed. As shown in Fig. 5(b), 12% better MSSIM was achieved by the proposed bit truncation scheme, compared to [15] at iso-energy. The same results were achieved with the scheme in [9], although it again covers a smaller range of energy-quality tradeoffs, and the quality target cannot be changed dynamically.

V. Conclusion

This brief proposed a novel nonzeroing bit truncation scheme for the design of energy-quality scalable adders. In contrast to

conventional zeroing bit truncation schemes, the proposed approach sets the inputs associated with the truncated digits equal to proper nonzero constant values that minimize the error on the output. The proposed technique is able to retain the dynamic energy-quality configurability of traditional reconfigurable approximate adders, while achieving more favorable energy-quality tradeoff. The proposed scheme increases the output quality by up to 8.5 dB in terms of PSNR, and 67% in terms of MED, at no energy penalty, compared to traditional zeroing bit truncation schemes.

When voltage scaling is applied, a 64% energy reduction has been observed compared to the approximate adder in [18], which specifically aims to extract energy savings from voltage scaling. More substantial benefits have been shown compared to existing approximate designs that do not allow dynamic energy-quality tradeoff. Finally, the benefits of the proposed scheme have been quantified in a DCT engine, and confirmed to be essentially the same as an individual adder.

REFERENCES

[1] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *Proc. ETS*, May 2013, pp. 1–6.

[2] M. Alioto, Ed., *Enabling the Internet of Things: From Integrated Circuits to Integrated Systems*. Cham, Switzerland: Springer, 2017.

[3] M. Alioto, "Energy-quality scalable adaptive VLSI circuits and systems beyond approximate computing," in *Proc. IEEE DATE*, Lausanne, Switzerland, Mar. 2017, pp. 127–132.

[4] M. Alioto, V. De, and A. Marongiu, "Guest editorial energy-quality scalable circuits and systems for sensing and computing: From approximate to communication-inspired and learning-based," *IEEE J. Trans. Emerg. Sel. Topics Circuits Syst.*, vol. 8, no. 3, pp. 361–368, Sep. 2018.

[5] D. Shin and S. K. Gupta, "Approximate logic synthesis for error tolerant applications," in *Proc. IEEE DATE*, Mar. 2010, pp. 957–960.

[6] F. Frustaci, M. Khayatzadeh, D. Blaauw, D. Sylvester, and M. Alioto, "SRAM for error-tolerant applications with dynamic energy-quality management in 28 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 50, no. 5, pp. 1310–1323, May 2015.

[7] N. Zhu, W. L. Goh, W. Zhang, K. S. Yeo, and Z. H. Kong, "Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 8, pp. 1225–1229, Aug. 2010.

[8] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 1, pp. 124–137, Jan. 2013.

[9] 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 Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[10] H. A. F. Almurib, T. N. Kumar, and F. Lombardi, "Inexact designs for approximate low power addition by cell replacement," in *Proc. IEEE DATE*, Mar. 2016, pp. 660–665.

[11] J. Miao, K. He, A. Gerstlauer, and M. Orshansky, "Modeling and synthesis of quality-energy optimal approximate adders," in *Proc. IEEE/ACM ICCAD*, Nov. 2012, pp. 728–735.

[12] F. Frustaci, D. Blaauw, D. Sylvester, and M. Alioto, "Approximate SRAMs with dynamic energy-quality management," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 6, pp. 2128–2141, Jun. 2016.

[13] S. Hashemi, R. I. Bahar, and S. Reda, "DRUM: A dynamic range unbiased multiplier for approximate applications," in *Proc. IEEE/ACM ICCAD*, Nov. 2015, pp. 418–425.

[14] M. Imani, D. Peroni, and T. Rosing, "CFPU: Configurable floating point multiplier for energy-efficient computing," in *Proc. IEEE/ACM DAC*, Jun. 2017, pp. 1–6.

[15] J. Park, J. H. Choi, and K. Roy, "Dynamic bit-width adaptation in DCT: An approach to trade off image quality and computation energy," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 5, pp. 787–793, May 2010.

[16] A. Dalloo, A. Najafi, and A. Garcia-Ortiz, "Systematic design of an approximate adder: The optimized lower part constant-OR Adder," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 8, pp. 1595–1599, Aug. 2018.

[17] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in *Proc. DAC*, Jul. 2012, pp. 820–825.

[18] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, "On reconfiguration-oriented approximate adder design and its application," in *Proc. IEEE/ACM ICCAD*, Nov. 2013, pp. 48–54.

[19] D. Gong, Y. He, and Z. Cao, "New cost-effective VLSI implementation of a 2-D discrete cosine transform and its inverse," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 4, pp. 405–415, Apr. 2004.

[20] *Public-Domain Test Images*. [Online]. Available: http://homepages.cae.wisc.edu/~ece533/images

[21] *Public-Domain Video Database for Testing Change Detection Algorithms*. [Online]. Available: http://www.changedetection.net/

[22] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.