Approximate Multipliers With Dynamic Truncation for Energy Reduction via Graceful Quality Degradation

Fabio Frustaci, *Member, IEEE*, Stefania Perri, *Senior Member, IEEE*, Pasquale Corsonello, *Member, IEEE*, and Massimo Alioto, *Fellow, IEEE*

***Abstract*—In approximate operators, dynamic truncation allows trading off energy and quality of computation at runtime. Although it exploits the specificity of the data being processed, its significant energy overhead over simple static truncation fundamentally limits its energy benefits. This brief describes a simple and efficient design methodology that reduces the energy consumption of dynamically truncated multipliers, based on a smart mapping of the partial products. A configurable hardware correction strategy is also proposed to enable graceful quality degradation, as well as more aggressive energy reduction at a given quality. When applied to Wallace multipliers, the proposed approach achieves quality, in terms of Mean Error Distance, up to 11× higher than the conventional dynamic truncation, at the same energy. In the case study of Discrete Cosine Transform compression, the proposed approximate multiplier reaches image qualities by 15-35% better, compared to prior art.**

***Index Terms*—Energy-quality scaling, approximate computing, multiplier, low-power design, VLSI.**

I. Introduction

ENERGY-QUALITY scaling has emerged as a general approach to the design of low-energy systems in applications that are inherently error resilient [1], [2]. In this general application domain, approximate multipliers have been widely explored as fundamental building block in signal processing and machine learning. Among other approaches, approximate compressors in the partial product reduction (PPR) stage were proposed in [3]–[8], replacing exact compressors based on half-adders (HAs) and full-adders (FAs). The static truncation technique [9], [10] inhibits the hardware resources needed to compute some of the least significant bits (LSBs) of the result, but the quality is inflexibly set at design time and it cannot leverage the specificity of the data being processed. On the contrary, dynamic truncation exploits such specificity and hence enables further energy savings, when the quality target varies over time [11]. As shown in [12], the number of truncated LSBs can be dynamically set at runtime by substituting the 2-input AND gates used in the partial product generation (PPG) stage with 3-input AND gates. As a downside, a hardware correction of the approximate result, and hence achieving graceful quality degradation, is difficult [12], as opposed to static truncation, in which the number of truncated LSBs is fixed. The output quality can be tuned also exploiting either a dual-quality compressor [13] or the dynamic perforation with rounding technique [14].

This brief introduces a novel methodology to design multipliers with dynamic energy-quality trade-off at runtime, improving the energy efficiency with respect to conventional dynamic truncations. The proposed approach is based on a partial product mapping that linearly reduces the energy according to the portion of unused LSBs, and a low-complexity hardware correction strategy in the PPG stage. When applied to the design of Wallace multipliers in a 28nm UTBB-FDSOI technology, the proposed technique exhibits an output accuracy by 11×, 74× and 10× compared to [12], [13], and [14] at iso-energy. As an example of application, the proposed approach was applied to image compression through Discrete Cosine Transform (DCT). Results show that the proposed dynamic truncation strategy increases the compressed image quality by up to 15% and 35% at iso-energy, compared to [12] and [13].

II. Background on Static and Dynamic Truncation

Static truncation simplifies a multiplier by pruning the hardware resources required to compute N_T LSBs of the result. Being $a_{(n-1:0)}$ and $b_{(n-1:0)}$ the two n-bit multiplication operands, the least significant N_T columns of compressors in the PPR stage can be avoided altogether, as depicted in Fig. 1a for the case $n = 8$ and $N_T = 8$. Analogously, the partial products $a_i \cdot b_j$ in the PPG stage with $i, j = 0 \cdots N_T - 1$ and $i + j < N_T$ do not need to be implemented. On the other hand, the dynamic truncation technique in [12] sets the number of truncated columns at runtime in the $[0, k_{max}]$ range, as in Fig. 1b. The 2-input AND gates in the PPG stage are substituted by 3-input AND gates computing $a_i \cdot b_j \cdot t_h$, with $h = i+j$ and $0 \leq h \leq k_{max}$, where the control signal t_h drives all the AND gates in the h-th column. To truncate N_T columns, t_h is

Manuscript received April 8, 2020; revised May 6, 2020; accepted May 24, 2020. Date of publication June 1, 2020; date of current version November 24, 2020. This brief was recommended by Associate Editor A. J. Acosta. *(Corresponding author: Pasquale Corsonello.)*

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

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

Massimo Alioto is with the Electrical and Computer Engineering Department, National University of Singapore (e-mail: massimo.alioto@nus.edu.sg).

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

Digital Object Identifier 10.1109/TCSII.2020.2999131

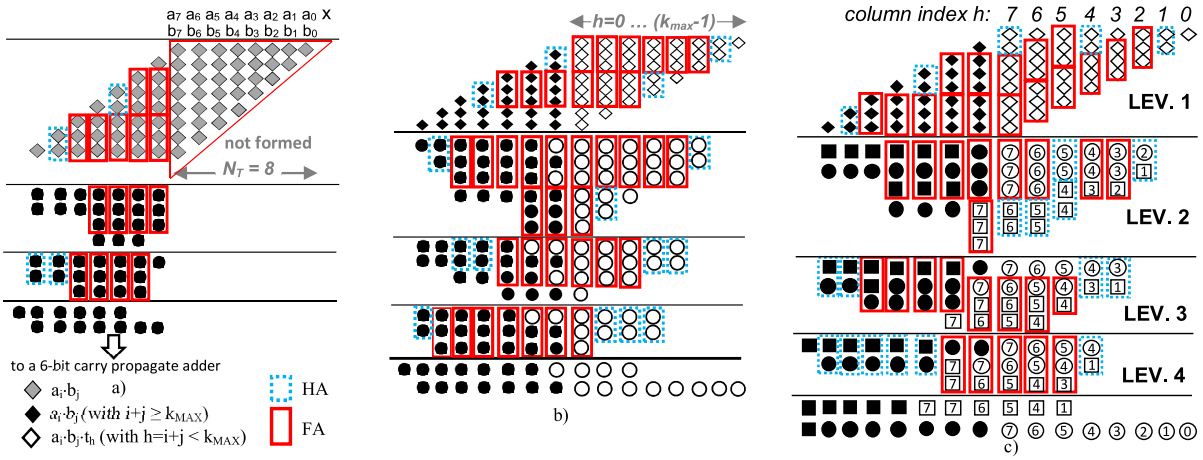

Fig. 1. The 8x8 unsigned Wallace multiplier: a) static truncation (NT = 8); b) conventional dynamic truncation [12]; c) proposed dynamic truncation.

set as in (1).

$$t_h = \begin{cases} 0, & \text{for } 0 \le h < N_T \\ 1, & \text{for } N_T \le h < k_{max}. \end{cases} \quad (1)$$

In this way, all the AND gates evaluating the partial products in the least N_T columns produce a logic 0 as desired, gating data processed by the compressors belonging to the same columns in the PPR stage and keeping their sum and carry outputs at 0 to save the related dynamic energy (see empty dots in Fig. 1b).

III. Proposed Partial Product Mapping Methodology

As shown in Fig. 1b, dynamic truncation multiplier has a higher number of active compressors and a deeper PPR stage, thus entailing an extra energy consumption compared to its statically truncated counterpart. Such energy overhead is well-known to be proportional to N_T [12], and is hence largest when the lowest precision is adopted, thus reducing the net energy savings coming from precision down-scaling under dynamic truncation. A more efficient partial product mapping methodology is discussed in the next subsection.

A. Low-Overhead Partial Product Mapping Methodology

The methodology proposed to design the PPR stage starts from the statically truncated architecture with $N_T = k_{max}$, as this defines the PPR structure with the minimum complexity at the lowest targeted precision. Then, extra compressors are progressively added to scale up the precision in an incremental fashion, minimizing the overhead while moving towards lower N_T (i.e., higher precision). Fig. 1c schematizes a multiplier designed in this way when $n = 8$ and $k_{max} = 8$. The carry and sum bits computed in each column of the PPR stage are depicted with squares and circles, respectively. To apply the proposed strategy efficiently, the following steps are performed orderly:

a) the first k_{max} columns within the 1st level of the PPR stage are mapped greedily: higher-order compressors (FAs) are used first, and lower-order compressors (HAs) are used afterwards;

b) in the subsequent levels, each carry or sum bit is labelled with the number i if it is data-gated for a truncation configuration N_T satisfying: $i \le N_T - 1$ (with $i < k_{max}$). Sum and carry bits are then sorted in descending order on the basis of their labels;

c) the bits belonging to the same column in the PPR stages are grouped in the ascending order. Three consecutive bits with the same label are grouped using a FA; conversely, if only two consecutive bits with the same label are encountered they are grouped with a HA; the remaining consecutive bits are grouped by using either a FA or a HA, depending on how many they are;

d) an isolated bit with the highest label is left ungrouped (e.g., the bit with label 7 in the 3rd level of Fig. 1c).

If the step c) leads to a deeper PPR stage, then FAs should be used. As an example, in the 4th level of Fig. 1c, one FA groups the two bits with label 7. Indeed, the use of a HA would require an extra level in the PPR stage.

B. Simulation Results and Comparison With Prior Art

The above described approach was applied to design an unsigned 8×8 Wallace Multiplier, which was compared to the conventional counterparts [12], [13], and [14]. In order to avoid the impact of a synthesizer tool and of the technology library, the schemes of Fig. 1 have been characterized using as compressors the minimum strength regular-Vth FA and HA standard cells of a 28-nm FDSOI 1V technology. For the statically truncated multiplier, different versions were designed with $N_T = 0$ and N_T varying between 5 and 8. Analogously, the dynamically truncated multipliers were designed adopting $k_{max} = 8$ to achieve the same precision scaling range of the statically truncated multipliers. All the multipliers use a ripple-carry (RCA) as the final carry propagate adder, for the sake of low energy consumption.

The designs were simulated at circuit level with the Cadence Ultrasim simulator, and their average energy per operation was evaluated over 10,000 consecutive random multiplications performed at the 1-GHz clock frequency. To define a realistic circuit environment, the outputs pins were loaded by D flip-flops with minimum drive strength. The quality metric Mean Error Distance (MED) [15] has been evaluated over uniformly distributed random inputs. The obtained results demonstrated that the MED values vary by less than 1% when the input dataset increases from 10k to 100k. As depicted in Fig. 2 (the results with the proposed correction strategy will be discussed in the next section), the dynamic truncation in [12]

TABLE I
COMPARISON RESULTS

	static truncation					conventional dynamic truncation [12]					proposed dynamic truncation					proposed dynamic truncation with correction				
	8×8					8×8					8×8					8x8				
N_T	**8**	**7**	**6**	**5**	**0**	**8**	**7**	**6**	**5**	**0**	**8**	**7**	**6**	**5**	**0**	**8**	**7**	**6**	**5**	**0**
Delay (ps)	507	587	659	698	777	627	680	685	728	800	564	674	725	766	830	640	710	725	770	831
Energy (fJ)	85	113	142	166	214	106	139	164	184	216	96	128	154	178	213	108	135	161	183	213
Area (um^2)	77	100	129	148	190	197	197	197	197	197	203	203	203	203	203	203	203	203	203	203
MED	452	192	80	32	0	452	192	80	32	0	452	192	80	32	0	202	97	44	19	0
MRED (%)	10	5.2	2.67	1.28	0	10	5.2	2.67	1.28	0	10	5.2	2.67	1.28	0	18	8.2	3.2	1.0	0
Err. Rate (%)	98.8	97.3	94.5	89.8	0	98.8	97.3	94.5	89.8	0	98.8	97.3	94.5	89.8	0	100	100	100	100	0
	16×16					16×16					16×16					16x16				
N_T	**16**	**14**	**12**	**10**	**0**	**16**	**14**	**12**	**10**	**0**	**16**	**14**	**12**	**10**	**0**	**16**	**14**	**12**	**10**	**0**
Delay (ps)	1007	1104	1197	1274	1539	1085	1184	1297	1363	1554	1011	1154	1269	1346	1570	1120	1176	1289	1367	1570
Energy (fJ)	478	624	737	833	1040	580	710	827	925	1110	504	635	749	840	1060	522	642	752	856	1060
Area (um^2)	500	591	664	727	827	852	852	852	852	852	873	873	873	873	873	873	873	873	873	873
MED	2.4E+5	5.4E+4	1.1E+4	2308	0	2.4E+5	5.4E+4	1.1E+4	2308	0	2.4E+5	5.4E+4	1.1E+4	2308	0	7.6E+4	1.8E+4	4136	993	0
MRED (%)	0.21	0.06	0.016	3.7E-3	0	0.21	0.06	0.016	3.7E-3	0	0.21	0.06	0.016	3.7E-3	0	0.33	0.05	0.011	2.2E-3	0
Err. Rate (%)	99.9	99.9	99.8	99.4	0	99.9	99.9	99.8	99.4	0	99.9	99.9	99.8	99.4	0	100	100	100	100	0

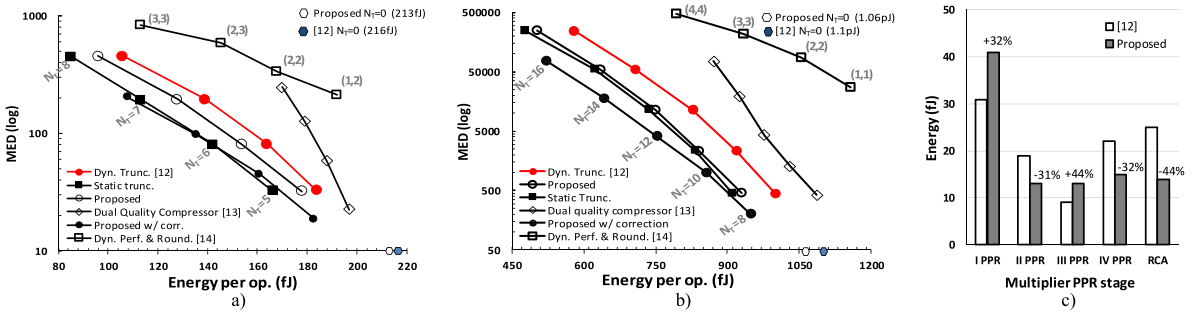

Fig. 2. Energy-quality trade-off obtained for a) 8x8 and b) 16x16 unsigned multipliers; c) Energy breakdown in the PPR stage of the 8 × 8 unsigned multiplier.

shows significant energy overhead from 10.8% ($N_T = 5$) to 24.7% ($N_T = 8$), compared to the static counterpart at equal output accuracy. Similarly, [12] invariably has a lower output accuracy at iso-energy, showing a MED increase of up to 2.4 × ($N_T = 7$) compared to static truncation. Compared to [12], the proposed strategy reduces the energy consumption by up to 9.5% ($N_T = 8$) and its maximum energy overhead compared to static truncation is reduced to only 13% ($-1.9\times$ with respect to [12]) ($N_T = 8$). It is worth noting that the higher the approximation (i.e., higher N_T), the higher the energy saving. This is because the percentage of compressors with a zero activity over the total number of employed compressors decreases as N_T is reduced, accordingly to the mapping strategy described in Section III-A.

The first N_T columns of the PPR stage of multiplier [13] have been implemented with dual-quality compressors, with N_T varying in the same range as for [12]. The dual-quality compressors use power gating and tristate buffers as output isolators, operating as described in this brief. From Fig. 2, it can be seen that [13] has a higher output accuracy for the same N_T value. Nevertheless, the proposed approach achieves an output accuracy that is up to $4\times$ higher in terms of MED at iso-energy. Moreover, for the same output accuracy, the energy of the proposed approach is reduced by up to 25% ($N_T = 7$). This is mostly due to the suppression of the extra tristate buffers required by the dual-quality compressors [13]. It is worth noting that the technique in [13] has a narrower energy trade-off range. Indeed, when N_T increases from 5 to 8, energy decreases by only 14%, whereas it decreases by up to 46% in the proposed approach. Finally, as described in this brief, input multiplexers have been used to set the number of perforated partial products and rounded bits (k, m) of the multiplier [14], assuring the best energy-quality trade-off achievable for such a multiplier. The latter always shows the worst energy-quality trade-off. At the parity of the achieved quality, it dissipates ∼50% more energy than the proposed design.

To better understand the energy benefits offered by the proposed design, Fig. 2c shows the energy breakdown observed through all the levels of the PPR stage under $N_T = 8$. Due to the step a) described in Section III-A, the first level of the PPR stage employs a higher number of compressors, and therefore it dissipates 32% more energy. Nevertheless, due to the reduced switching activity of their internal nodes, the levels II, IV and the last one dissipate 31%, 32% and 44% lower energy.

Results obtained for a 16 × 16 unsigned Wallace multiplier at the 600-MHz clock frequency are reported in Fig. 2b, which shows how higher bit-widths make the benefits offered by the proposed design methodology more pronounced. The MED values evaluated over 10k, 100k and 1M uniformly distributed random inputs vary by less than 1%. Compared to [12] and [13], the MED achieved by the proposed design is $4.9\times$ and $42\times$ lower, respectively, at iso-energy. Also, the energy reduction over [12] and [13], at iso-quality, grows up to 13% and 27%, whereas the maximum energy overhead reached with $N_T = 16$ over the static truncation is only 5.4%. For lower values of N_T, the proposed design approach exhibits an energy that is essentially the same as the static truncation counterpart. On the other hand, the energy overhead of

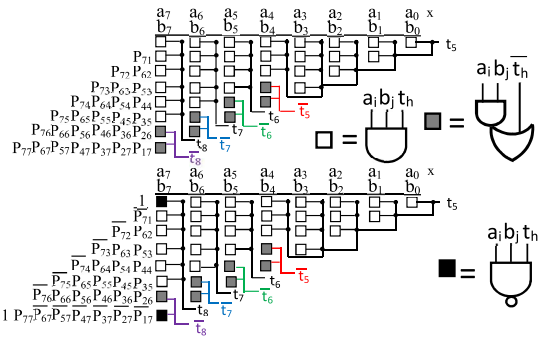

Fig. 3. Correction strategy for the 8 × 8 multiplier: a) unsigned; b) signed.

conventional dynamic truncation over the static counterpart is up to 19%.

The behaviours of the statically truncated, the conventional dynamically truncated, and the proposed dynamically truncated multipliers are summarized in Table I in terms of area, delay, energy and approximation, detailed as the MED, the Mean Relative Error Distance (MRED) and the Error Rate. As expected, static truncation leads to the lowest area, energy and delay, although it prohibits any precision flexibility. The delay of the proposed design increases only by up to 11% and 0.4% when 8- and 16-bit operands are processed, respectively. On the other hand, the delay penalty experienced by the 8×8 and 16×16 multipliers designed as in [12] are up to 23.6% and 7.7%, respectively. The proposed approach expectedly leads to a slightly higher area compared to [12]. As an example, for $n = 8$, it requires 1 less FA and 9 more HAs, with an area increase of only 3%. Interestingly, for $N_T = 8$, the proposed multiplier is 10% faster than [12] since the carry propagation path of the final RCA is reduced by 2 FAs, as it can be inferred by Fig. 1.

Further comparisons at the parity of MRED demonstrated that, with respect to [13], the 8 × 8 multiplier designed as proposed here dissipates up to 36% less energy. Even better behaviour is achieved in comparison to [14]. Indeed, the novel 16×16 multiplier exhibits up to 38% lower energy dissipation.

Finally, for purposes of iso-delay analysis, the Cadence Genus tool has been used to synthesize the full precision 8×8 (16×16) multipliers with the timing constraint of 500ps (1ns). The obtained results demonstrated that for $N_T = 8$ ($N_T = 16$), the proposed approach reduces the energy consumption by 21% (16%) with respect to the scheme [12].

IV. Dynamic Truncation Error Compensation

As a further improvement over [12], a simple hardware strategy for error compensation is introduced in this section. Table II reports the mean error (ME) of the multiplier for different N_T values, as obtained through exhaustive behavioural simulations. The ME can be considered as an offset to add to the result in order to increase the output accuracy. The MED obtained adding this offset (MED_wo) is significantly improved with respect to the MED of Table I. Clearly, such an offset depends on the value of N_T, thus adding its exact value results in a complex circuit modification. To achieve this at reduced circuit complexity, we propose to add an approximate value of the correction offset directly within the PPG stage in an incremental fashion, as follows. The required correction offset for a given N_T is incrementally approximated to

TABLE II
Mean Error and Approximate Correction Constant

N_T	ME	MED_wo	cc^{N_T}(dec)	cc^{N_T} (bin)	MED_wao
8×8 UNSIGNED MULTIPLIER					
6	80	42	96=**64**+cc^6	1100000	44
7	192	92	224=**128**+cc^7	11100000	97
8	452	199	480=**256**+cc^8	111100000	202
8×8 SIGNED MULTIPLIER					
6	80	42	96=**64**+cc^6	1100000	44
7	192	92	224=**128**+cc^7	11100000	97
8	319	184	352=**128**+cc^8	101100000	188
16×16 UNSIGNED MULTIPLIER					
N_T	ME	MED_wo	cc^{N_T}(dec)	cc^{N_T} (hex)	MED_wao
12	11290	4065	11648=**9216**+cc^{10}	2D80	4099
14	53920	17827	48512=**36864**+cc^{12}	BD80	17890
16	245800	76502	245120=**196608**+cc^{14}	3BD80	76540
16×16 SIGNED MULTIPLIER					
12	11240	4052	11648=**9216**+cc^{10}	2D80	4092
14	53150	17658	48512=**36864**+cc^{12}	BD80	17760
16	212800	73920	212352=**163840**+cc^{14}	33D80	73953

the value cc^{N_T} ensuring that $cc^{N_T+1} = cc^{N_T} + x$, with x being a constant whose binary value contains 1's only for the bit positions equal to or larger than $N_T - 1$. In this way, the same gates used to add the offset for a value of N_T are still used in the offset addition for the case $N_T + 1$. The value of x that best approximates the exact mean error is highlighted in bold in Table II, for signed and unsigned 8×8 and 16×16 multipliers. The above error correction strategy is readily integrated into the PPG stage, as illustrated in Fig. 3 for the 8×8 multiplier. Some 3-inputs AND gates are substituted with AND-OR gates driven by $\overline{t_h}$. In this way, when N_T columns are truncated and each signal t_h is set as in (1), the outputs of the AND-OR gates in the h-th column in Fig. 3 (with $h < N_T$) are set to logic 1 regardless of the inputs, thus enabling the addition of cc^{N_T+1}. It is worth noting that such a strategy reflects the proposed incremental offset addition described above, which results in minimal circuit modification and no extra control signals. Table II shows that the MED value (MED_wao), achieved by adding the approximate constant cc^{N_T}, increases at most by 5.4%, compared to MED_wo, thus demonstrating the benefits of the proposed approximate correction strategy. The MED overhead due to the approximation is lower for the 16 × 16 multiplier with respect to the 8 × 8 multiplier. This is due to the intrinsic higher MED value shown by the 16 × 16 multiplier. Fig. 2b shows the energy-quality trade-off obtained when the proposed correction strategy is applied to unsigned multipliers. For 16-bit (8-bit) inputs, a remarkable MED reduction by a factor of 11 × (2.2×) is observed for the case $N_T = 10(N_T = 7)$ at iso-energy, whereas, for the case $N_T = 16(N_T = 8)$, the energy is reduced by 26.2% (22%) at iso-quality. When the proposed truncation and correction approaches are jointly introduced, greater advantages are achieved. Compared to [13], the proposed techniques improve the output quality by up to 70× at iso-energy, and reduce energy by up to 40%. Slightly better results were obtained for signed multipliers (not shown for the sake of brevity), where the joint adoption of the above proposed techniques improves the output quality by up to 125× over [13].

Beyond Wallace multipliers, the above techniques clearly apply to other multiplier tree structures. As an example, the above analysis was reiterated for the popular Dadda trees designed with 4:2 compressors. In prior art adopting dual-quality 4:2 compressors, the DQ4:2C_2 compressors were

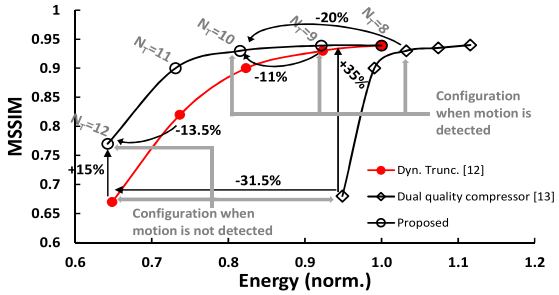

Fig. 4. Energy-quality tradeoff for the DCT application.

found to have the most favourable energy-quality trade-off [13]. Compared to this solution, the joint adoption of the proposed truncation and correction to the 8×8 multiplier substantially improves the MED by up to $7\times$ at iso-energy, and reduces the energy by up to 33% at iso-quality. Finally, the area increase compared to [12] is less than 5%.

V. Case Study: Discrete Cosine Transform

As a representative use case, the approximate multipliers based on the proposed truncation and correction techniques were explored in the context of the Discrete Cosine Transform (DCT), as a fundamental building block for video compression [16]. In frames where motion is (is not) detected, a higher (lower) image quality can be tolerated thus a multiplier configuration with a low (high) value of N_T can be enabled. Considering 8-bit input images, signed multipliers with 12-bit inputs were found to be required. The resulting energy-quality trade-off is depicted in Fig. 4 for the *backdoor* benchmark [17], chosen as a typical example of real video sequence taken from surveillance cameras (an external circuit is assumed to switch the truncation configuration). Quality was measured in terms of the Mean Structural Similarity (MSSIM) metric, which is relevant to the applications where DCT is employed [18]. In approximate DCT designs, the MSSIM target is dynamically modified to higher (lower) quality in frames where motion is (is not) detected. From Fig. 4, the proposed dynamic truncation and error compensation approaches always achieve a higher MSSIM than [12] for any value of N_T, or lower energy at a given MSSIM target. For example, under a typical MSSIM target of 0.93 in frames where motion is detected (i.e., 7% quality loss with respect to the uncompressed image), N_T needs to be set to 9 for [12], 10 for [13], and 10 for the proposed approach. Under such N_T values, Fig. 4 shows how the proposed approach reduces energy by 11% and 20%, compared to [12] and [13], respectively. On the other hand, in frames where motion is not detected, N_T needs to be set to 11 for [12], 12 for [13] and 12 for the proposed approach. As reported in Fig. 4, when motion is not detected, the proposed dynamic truncation and error compensation approaches reduce energy by 13.5% and 31.5% compared to [12] and [13], respectively. When frames with and without detected motion are mixed as in the 2,000-frame *backdoor* video sequence, the proposed technique is able to achieve an overall energy reduction of 12% and 26% compared to [12] and [13], respectively.

VI. Conclusion

This brief describes a strategy to improve the energy efficiency of the dynamic truncation technique in designing energy-quality scalable multipliers. The proposed approach is based on a smart mapping of the partial products, which reduces the energy dissipation of the portion of the multiplier that is not truncated. Moreover, a simple yet effective hardware correction strategy is described, which is configurable accordingly to the truncation configuration. Compared to traditional dynamic truncation, the proposed scheme increases the output quality by up to $11\times$ in terms of MED. Finally, the benefits of the proposed scheme have been quantified in the DCT application, showing an energy reduction of up to 13.5%.

References

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

[2] M. Alioto, "Energy-quality scalable adaptive VLSI circuits and systems beyond approximate computing," in *Proc. IEEE Design Autom. Test Europe Conf. Exhibit. (DATE)*, Lausanne, Switzerland, Mar. 2017, pp. 127–132.

[3] D. Esposito, A. G. M. Strollo, E. Napoli, D. De Caro, and N. Petra, "Approximate multipliers based on new approximate compressors," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 12, pp. 4169–4182, Dec. 2018.

[4] I. Qiqieh, R. Shafik, G. Tarawneh, D. Sokolov, S. Das, and A. Yakovlev, "Significant-driven logic compression for energy-efficient multiplier design," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 8, no. 3, pp. 417–430, Sep. 2018.

[5] H. Jiang, C. Liu, F. Lombardi, and J. Han, "Low-power approximate unsigned multipliers with configurable error recovery," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 1, pp. 189–202, Jan. 2019.

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

[7] 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.

[8] R. Marimuthu, Y. E. Rezinold, and P. S. Mallik, "Design and analysis of multiplier using approximate 15–4 compressor," *IEEE Access*, vol. 5, pp. 1027–1036, 2017.

[9] E. J. King and E. E. Swartzlander, Jr., "Data-dependent truncation scheme for parallel-multipliers," in *Proc. 31st Asilomar Conf. Signals Syst. Comput.*, vol. 2, Nov. 1997, pp. 1178–1182.

[10] C.-H. Chang and R. K. Satzoda, "A low error and high performace multiplexer-based truncated multiplier," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 12, pp. 1767–1771, Dec. 2010.

[11] A. Raha, H. Jayakumar, and V. Raghunathan, "Input-based dynamic reconfiguration of approximate arithmetic units for video encoding," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 3, pp. 846–857, Mar. 2016.

[12] M. de la G. Solaz, W. Han, and R. Conway, "A flexible low power DSP with a programmable truncated multiplier," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 11, pp. 2555–2568, Nov. 2012.

[13] O. Akbary, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Dual-quality 4:2 compressor for utilizing in dynamic accuracy configurable multipliers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 4, pp. 1352–1361, Apr. 2017.

[14] V. Leon, G. Zervakis, S. Xydis, D. Soudris, and K. Pekmestzi, "Walking through the energy-error Pareto Frontier of approximate multipliers," *IEEE Micro*, vol. 38, no. 4, pp. 40–49, Jul./Aug. 2018.

[15] H. A. F. Almurib, T. N. Kumar, and F. Lombardi, "Inexact designs for approximate low power addition by cell replacement," in *Proc. IEEE Design Autom. Test Europe Conf. Exhibit. (DATE)*, Dresden, Germany, Mar. 2016, pp. 660–665.

[16] D. Gong, Y. He, and Z. Go, "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.

[17] *Public-Domain Video Database for Testing Change Detection Algorithms*. Accessed: May 17, 2019. [Online]. Available: http://www.changedetection.net/

[18] 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.