Approximate Computing Based Low Power Image Processing Architecture for Intelligent Satellites

Zhixi Yang[1,2], Rong Lv[2(✉)], Xianbin Li[1], Jian Wang[1], and Jun Yang[2]

[1] National Innovation Institute of Defense Technology, Academy of Military Science, Beijing 100073, China
nudtyzx@163.com, lixianbincn@163.com, wangjian710108@126.com
[2] The 63rd Research Institute, National University of Defense Technology, Nanjing 210000, China
lvrong17@nudt.edu.cn, john323@163.com

Abstract. Approximate computing is an innovative circuit paradigm for lower power and real time image processing architecture within an intelligent satellite. Multiplication and addition are often fundamental functions for many image processing applications. Based on previous approximate compressor designs, a recursive type multiplier is first proposed. A reduced gate-level complexity full adder is then proposed. Extensive simulation results show that the proposed designs achieve significant reductions in area, power and delay compared with exact recursive multiplier and adders, as well as other approximate designs found in the technical literature. An image processing application is performed to further show that the performance of the proposed approximate designs for image processing achieves a very good accuracy (measured by the peak signal to noise ratio) as well as substantial reductions in power dissipation and delay.

Keywords: Approximate multiplier · Approximate adder · Low power · Real time · Image processing · Intelligent satellite

1 Introduction

The power consumption and processing speed are the major performance metrics for on-board image processing architectures within intelligent satellites. These architectures commonly rely on digital signal processing circuits and various design techniques are used to reduce their processing power. Approximate computing uses inexact or approximate processing circuits to produce meaningful results while provide an additional layer of power saving over conventional low-power design techniques. The simplified and approximate circuits operating at higher performance and/or lower power compared to their accurate logic counterparts [1], which has been extensively used for error resilient applications such as image processing [2].

Q. Wu et al. (Eds.): WiSATS 2020, LNICST 358, pp. 351–363, 2021.
https://doi.org/10.1007/978-3-030-69072-4_28

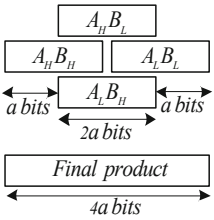

Fig. 1. Recursive multiplication by dividing multiplier and multiplicand into two parts.

Approximate circuits proposed starting from fundamental components such as multipliers and full adders [3,4]. A multiplier is designed either in the simple and popular array configuration [5,6] or in parallel column compression architecture [7,8]. The latter type of design is frequently used for high performance due to the shorter delay incurred by the compressors [9,10]. Different designs of a compressor lead to different architectures for the multiplier [11–14]. Several approximate schemes have been proposed as applicable to a multiplier [15–20]. In addition to these methods, several approximate compressors have been proposed for multiplication [21,22]. [23] uses the simple recursive multiplication technique which has also been used in this paper.

In terms of approximate adder design, [24] simplifies a single adder cell by removing transistors from a mirror adder (AMA). In this paper, multiplexer-based approximate adder designs are proposed and the multiplexer designs utilize transmission gates (TG) due to the low power dissipation than a conventional CMOS multiplexer.

An image processing application using previously proposed approximate compressors, multipliers [25] and TG based approximate adders are considered, the analysis and simulation results show that the proposed approximate designs for both the multiplier and adder are viable for approximate computing.

2 Proposed Approximate Designs

2.1 Approximate Multiplier Design

An alternative approach for designing a multiplier is to use recursive multiplication as shown in Fig. 1. Recursive multiplication splits input into two parts, i.e. $A_H(B_H)$ and $A_L(B_L)$; then perform multiplications as $A_H B_H$, $A_H B_L$, $A_L B_H$ and $A_L B_L$; finally four terms are added. The designs studied in this paper are shown in Table 1.

2.2 Approximate Adder Design

An alternative approach to implement complex logic is to use a logic network of switches, such as a multiplexer. Multiplexer logic can be accomplished by using either gate or transistor level designs. Conventional gate level multiplexers use

Table 1. Design features of approximate 8 × 8 multipliers.

Multiplier design	Features
M1-M3	Multipliers M1-M3 in [1]
M4-M6	ACCI1-ACCI3 [1] compressor based 4 × 4 multiplier for $A_L B_L$, $A_L B_H$, $A_H B_L$ only
Multiplier [2]	Lower 8 columns use compressors shown in [2]
Multiplier [3]	Inaccurate counter [3] based multiplier
Mul_Acc	Accurate compressor based recursive 8 × 8 multiplier

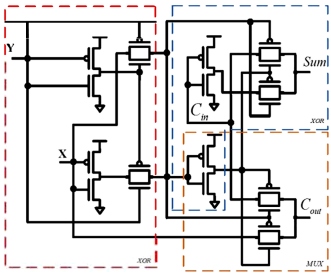

Fig. 2. TG based one bit accurate full adder.

complex gate structures, thus transistor level designs that include compounded logic and transmission gates are better alternatives.

[26] has shown that a TG based full adder exhibits good power and delay performance with a simpler circuit. Figure 2 shows an accurate TG-based full adder and this implementation is based on TGs and several inverters; the designs of approximate adders require either the removal of some modules, or changing signals for generating Sum or C_{out}. Moreover, in the proposed approximate adders, TG based multiplexers are frequently used due to the simple structure.

Figure 3 and Fig. 4 show the proposed approximate adders (denoted as APA1-2). The feature common is the modification of the first module XOR gate, so reducing the node capacitance and lowering the power dissipation by using only one conventional XOR gate for a reduction in delay and power.

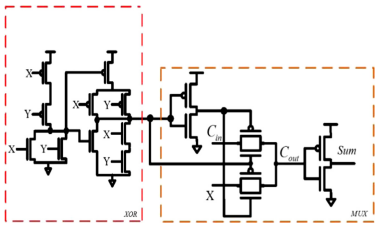

Fig. 3. TG based approximate adder APA1.

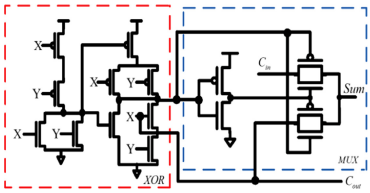

Fig. 4. TG based approximate adder APA2.

In one bit accurate adder, there are six cases when $Sum = \overline{C_{out}}$, thus an option is to keep C_{out} accurate and use an inverter after C_{out} to generate the Sum signal. Its logic expressions are given as:

$$Sum = \overline{(X \oplus Y)C_{in} + XY} \tag{1}$$

$$C_{out} = (X \oplus Y)C_{in} + XY \tag{2}$$

In APA1 the carry signal C_{out} is accurate while in APA2 C_{out} is modified. The error rate is four cases out of eight for combination of C_{out} and Sum, which is higher than APA1.

$$Sum = (X \overline{\oplus} Y)C_{in} + X\overline{Y} \tag{3}$$

$$C_{out} = X \tag{4}$$

3 Simulations Results

3.1 Metric for Approximate Design

Several metrics are introduced to quantify the effects of errors in an approximate design, including normalized error distance (NED) [2], pass rate (PR) [2] and accuracy of amplitude (Acc_{amp}) [22]. NED is defined as the mean absolute error over maximum value of the error. The pass rate is the ratio of the number of correct outputs over all outputs. Acc_{amp} is defined as:

$$Acc_{amp} = 1 - \frac{AbsoluteError}{|result_{correct}|} \tag{5}$$

$result_{correct}$ is the correct result for a certain input combination.

3.2 Approximate Compressor Based Recursive Multipliers

[25] has assessed both accuracy and electrical performance for M1-M3, hence in this subsection only recursive approximate multipliers using approximate compressors in [25] are simulated. The proposed approximate multipliers are simulated for an 8×8 recursive multiplication scheme. Delay, power consumption and area are investigated for these approximate designs compared to an exact multiplier using a recursive multiplication scheme with accurate compressors.

Table 2. Comparison of 8×8 multiplier for different approximate compressors based recursive multiplication.

Multiplier design	Power (uW)	Delay (ns)	Area (um^2)	NED (10^{-4})	PR (%)	Acc_{amp} (%)
M4	164.98	3.06	782.08	0.16	98.88	99.99
M5	165.67	2.92	775.84	1.56	89.7	99.86
M6	161.98	2.83	764.92	2.42	84.38	99.81
Mul_Acc	179.76	3.14	836.68	N/A	N/A	N/A

As discussed previously, $A_L B_L$, $A_H B_L$, $A_L B_H$ utilize approximate compressor based multipliers, while an accurate 4×4 multiplier is used for $A_H B_H$.

The proposed approximate multipliers are simulated for an 8×8 multiplication Dadda tree. Delay, power consumption and area are investigated for approximate designs compared with an exact multiplier. Table 2 shows the comparison for these circuit based metrics and accuracy analysis. M6 has the least values for power, area and delay while a poor accuracy.

3.3 TG Based Approximate Adder

In this subsection, comparison with AMAs is presented for single APA in terms of accuracy and power, delay. Simulation is performed using Cadences Ultrasim simulator in STMicroelectronics 65 nm process, for which 1.0 V is used as the standard supply voltage (Vdd). A load of four standard Inverters is utilized, but its energy consumption is uncounted for in the evaluation of all adders. Inputs are provided by independent voltage sources. Since some inputs drive the outputs, the energy provided by the input signals is included in the simulation results. A comparison of each approximate design and the accurate full adder is pursued with respect to power consumption, delay and power delay product (PDP). All possible 64 transitions for different input combinations are considered for power and delay measurement. The delay is calculated from the input C_{in} to output C_{out} since it would propagate within a RCA; the value in the table is largest value under 64 transitions. AMAs in [24] are also included as a comparison reference. The NED is obtained by using an 8-bit RCA with all APAs. The results are shown in Table 3.

In terms of power, APA2 has the least power. AMA3 has the lowest value with 2.992uW within AMAs. However, AMA3 has a larger MED/NED than APA2.

In terms of delay, since APA2 is using input as output, the carry in-carry out delay is considered to be zero. For AMA1-3, they have similar delay within which AMA1 has largest delay; APA1 has the largest delay as more than 0.2 ns.

In terms of PDP which is the product of delay and power. Surprisingly APA2 has the least value as 0. The reason is obvious that the carry in-carry out delay is 0, thus there is no carry chain for APA2. While for AMAs, AMA4 has the least value. In a general conclusion, AMA4 is a better design within AMA4 with acceptable MED/NED and PDP.

Table 3. Simulation results for various approximate adders. Metrics of NED/MED are obtained by using an 8 bit RCA; power and delay is obtained for single approximate adder design.

Adder design	MED	NED	C_{out} delay (ns)	Power (uW)	PDP (fJ)
APA1	90.55	0.3537	0.206	3.971	0.794
APA2	63.99	0.25	0	2.883	0
AMA1	17.39	0.068	0.138	3.423	0.475
AMA2	90.55	0.3537	0.133	3.230	0.431
AMA3	74.31	0.2895	0.125	2.992	0.375
AMA4	33.12	0.1294	0.071	3.525	0.252

3.4 Comprehensive Comparison

This section discusses a new metric for a fair comparison by considering power dissipation, area and delay as well as accuracy. The Power, delay and area product (PDAP) is defined as:

$$PDAP = power * area * delay \tag{6}$$

The normalized PDAP for a group of designs is then defined as:

$$NPDAP_i = PDAP_i / max\{PDAP_j\} \tag{7}$$

where j denotes the design in the group with the largest value of PDAP; the design group includes M1_4, M2_6, M3_6, M4-M6, and the designs in [21] and [22]. The Normalized NED is defined in a similar manner, i.e.,

$$NNED_i = NED_i / max\{NED_k\} \tag{8}$$

Note that $NNED_i$ and $NPDAP_i$ relate to the same design i but the normalization is different, because the maximum values can occur in different designs (i.e. j and k). In this paper, the design of [22] has the highest value of PDAP and the highest value of NED. Figure 5 shows the NNED vs. NPDAP plot for approximate multipliers; so the design with the smaller NNED and NPDAP is the better design among the comparison set according to Pareto Front, i.e. nearest to the origin in Fig. 5. The best design is M6 with desirable position from origin.

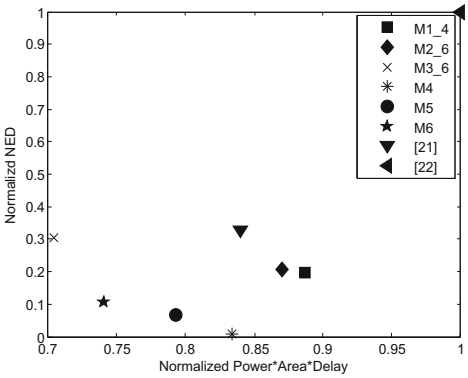

Fig. 5. Normalized NED vs. Normalized power*delay*area for approximate multiplier designs.

4 Image Processing Architecture Design

This section presents an image processing application using the proposed approximate multipliers, adders and approximate multipliers M1-M3 in [25]. An image sharpening algorithm is considered; image sharpening is functionally implemented in Verilog. The compressed image quality is measured by the peak signal noise ratio (PSNR).

The image sharpening is a Gaussian smoothing based image filter and the algorithm performs as [27]:

$$S(x,y) = \\ 2 \times I(x,y) - \frac{1}{273}\Sigma_{i=-2}^{2}\Sigma_{j=-2}^{2}G(i+3,j+3)I(x-i,y-i) \tag{9}$$

where I and S are original and processed image; G is a matrix given as:

$$G = \begin{pmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{pmatrix} \tag{10}$$

This algorithm is implemented in Verilog to build the whole architecture as shown in Fig. 6. Input and output signals are connected with registers. The components are identical apart from the multiplier and adder in Fig. 6.

Five images are selected for the sharpening algorithm executed using different approximate multipliers and adders; these images are selected, because they show features commonly found in multimedia applications; the corresponding PSNR and NED values are then measured.

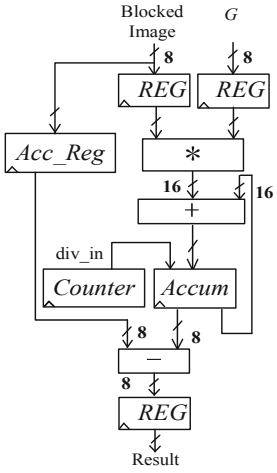

Fig. 6. Architecture of image sharpening algorithm.

4.1 Approximate Multiplier Based Image Sharpening

Two of the five processed images are shown in Figs. 7 and 8; also, the ranking of PSNR values are very close for those five images when processed by the same approximate design. When compared to the accurate results, all approximate designs (except the multiplier of [22]) produce images whose quality degradation is not perceived by human eyes. The images generated by the multiplier of [22] show areas of low quality (indicated by bold circles). For M4, for the example images, the PSNR is infinite, meaning no errors for M6 for the processed example images.

Table 4 also gives the NED and PSNR values for these two images. Table 4 shows that all approximate multipliers achieve PSNR values higher than 30dB; the approximate multipliers in [25] and proposed in this manuscript show a better image quality in terms of PSNR values than the two designs of [21] and [22]. The PSNRs of the output images generated by M2_6, M3_6 and M5, M6 reaches above 47dB, a high value that is acceptable by most applications.

4.2 Approximate Adder Based Image Sharpening

In this algorithm, addition is performed by the approximate adders proposed in this paper while for other operations, i.e., multiplication, subtraction and division all use accurate operations.

The sharpening algorithm executed using different approximate adders and the corresponding PSNR and NED values are then measured. For this algorithm, a 16 bit approximate adder is used, because the maximum possible sum is $255 * 273$, i.e. approximately $2^{16} - 1$. Case when lower 8 LSBs using approximate adders and higher MSBs using accurate adders are considered for the

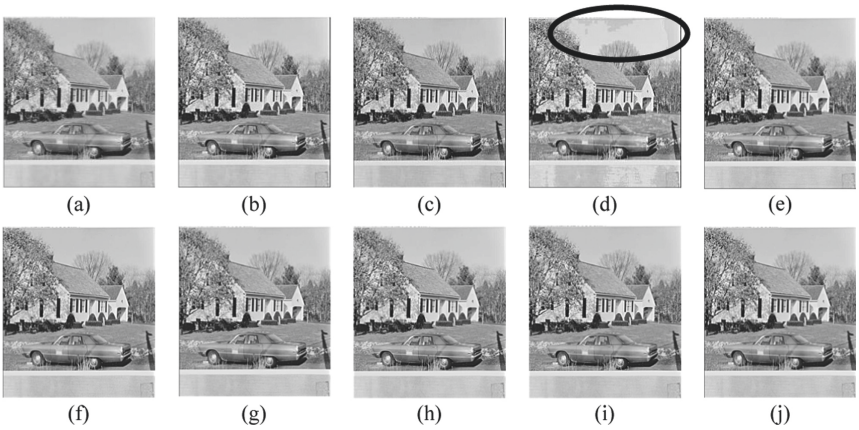

Fig. 7. Image sharpening results for Image 1: (a) original image, (b) using an accurate multiplier, (c) using multiplier of [21], (d) using multiplier of [22], (e) using M1_4, (f) using M2_6, (g) using M3_6, (h) using M4, (i) using M5, (j) using M6.

Fig. 8. Image sharpening results for Image 2: (a) original image, (b) using an accurate multiplier, (c) using multiplier of [21], (d) using multiplier of [22], (e) using M1_4, (f) using M2_6, (g) using M3_6, (h) using M4, (i) using M5, (j) using M6.

comparison. Table 5 shows the PSNR and NED for the example image, and it is easy to see when the approximate RCA with APAs has smaller NED, and then it would have better PSNR, i.e., image quality.

Table 4. Comparison using different Dadda tree multipliers based on inaccurate compressors for image processing.

	Image 1		Image 2	
Multiplier design	PSNR (dB)	NED (10^{-5})	PSNR (dB)	NED (10^{-5})
Multiplier [1]	35.61	5.18	35.97	4.90
Multiplier [2]	32.27	7.27	35.60	3.02
M1_4	66.06	0.029	66.17	0.029
M2_6	47.72	0.936	52.17	0.46
M3_6	47.98	0.892	52.31	0.44
M5	49.52	0.64	56.03	0.18
M6	49.35	0.68	54.84	0.23

Table 5. Comparison using different adders for image sharpening.

	Image 1		Image 2	
Multiplier design	PSNR (dB)	NED (10^{-4})	PSNR (dB)	NED (10^{-4})
APA1	45.31	0.136	45.39	0.132
APA2	43.09	0.17	43.37	0.16

5 Conclusion

Approximate computing is an innovative paradigm for error-resilient arithmetic circuits, because it offers significant advantages for design. In this paper, previously proposed compressor designs are utilized to design recursively based approximate multipliers and two transmission gate based approximate adders are proposed, which are assessed for both accuracy and electrical performance. The following conclusions are drawn from the extensive simulation presented in this paper:

(1) M4 has the least value of NED and best value of PS and Acc_{amp}, hence it has the best accuracy.
(2) By performing power delay and area product (PDAP) to indicate the utilization of resource and (PDAP vs. NED), M6 shows that it is a good candidate for approximate multiplication with lower power/area/delay and high accuracy requirements.
(3) When considering all metrics, generally M6 is the best approximate 8×8 multiplier.
(4) An image processing application by using design in [25] and proposed designs show that all proposed approximate multiplier design produce better quality images than the multipliers of [21] and [22] and M6 has the good image quality qualified by the PSNR while has the lower PDAP.

(5) The above proposed designs achieve lower power and faster processing speed than traditional image processing architecture, which can be potentially used in on-board in an intelligent satellite.

6 Discussion

This section discusses the relation of this paper to prior work. Compared to previously published work, this work extends the purely array based multiplier into recursively based multiplier. Recursive multiplier has flexible ability to adjust configuration of accurate and approximate computation, hence approximate multiplier with varying accuracy is easy to obtain. Moreover, only [23] has proposed recursive multiplier hence this work also extends this type of approximate multiplier designs. This paper also applies the designs to real image processing application compared to previous work [25]. The proposed image processing architecture could be used on-board within the satellites to achieve low power and high processing speed.

References

1. Han, J., Orshansky, M.: Approximate computing: an emerging paradigm for energy-efficient design. In: Proceedings of the ETS 2013: Proceedings of the 18th IEEE European Test Symposium, Avignon, France, May 2013, pp. 1–6 (2013)
2. Liang, J., Han, J., Lombardi, F.: New metrics for the reliability of approximate and probabilistic adders. IEEE Trans. Comput. **62**(9), 1760–1771 (2013)
3. Gupta, V., Mohapatra, D., Park, S.P., Raghunathan, A., Roy, K.: IMPACT: imprecise adders for low-power approximate computing. In: ISLPED 2011: Proceedings of the 17th IEEE/ACM International Symposium on Low-Power Electronics and Design, pp. 409–414 (2011)
4. Mahdiani, H.R., Ahmadi, A., Fakhraie, S.M., Lucas, C.: Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft computing applications. IEEE Trans. Circuits Syst. I Regul. Pap. **57**(4), 850–862 (2010)
5. Kang, S.M., Leblebici, Y.: CMOS Digital Integrated Circuits. McGraw-Hill, New York (1999). ISBN 0-07-116427-8
6. Rabaey, J.M., Chandrakasan, A.P.: Digital Integrated Circuits, 2nd edn. Prentice Hall, Upper Saddle River (2003). ISBN 0-13-090996-3
7. Wallace, C.S.: A suggestion for a fast multiplier. IEEE Trans. Electron. Comput. **EC–13**(1), 14–17 (1964)
8. Dadda, L.: Some schemes for parallel multipliers. Comput. Arithmetic **34**, 349–356 (1965)
9. Bickerstaff, K.C., Swartzlander, E.E., Schulte, M.J.: Analysis of column compression multipliers. In: The Proceedings of 15th IEEE Symposium on Computer Arithmetic, Vail, CO, USA, June 2001, pp. 33–39 (2001)
10. Gahlan, N.K., Shukla, P., Kaur, J.: Implementation of Wallace tree multiplier using compressor. Int. J. Comput. Technol. Appl. **3**(3), 1194–1199 (2012)
11. Ma, W., Li, S.: A new high compression compressor for large multiplier. In: ICSIT 2008: 9th International Conference on Solid-State and Integrated-Circuit Technology, Beijing, China, October 2008, pp. 1877–1880 (2008)

12. Kwon, O., Nowka, K., Swartzlander, E.E.: A16-bitx16-bit MAC design using fast 5:2 compressors. In: Proceedings of IEEE International Conference on Application-Specific Systems, Architectures, and Processors, Boston, MA, USA, July 2000, pp. 235–243 (2000)
13. Chang, C.H., Gu, J., Zhang, M.: Ultra low-voltage low-power CMOS 4–2 and 5–2 compressors for fast arithmetic circuits. IEEE Trans. Circuits Syst. I Regul. Pap. **51**(10), 1985–1997 (2004)
14. Prasad, K., Parhi, K.K.: Low-power 4–2 and 5–2 compressors. In: Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, November 2001, vol. 1, pp. 129–133 (2001)
15. Swartzlander, E.E.: Truncated multiplication with approximate rounding. In: Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, October 1999, vol. 2, pp. 1480–1483 (1999)
16. Schulte, M.J., Swartzlander, E.E.: Truncated multiplication with correction constant [for DSP]. In: Workshop on VLSI Signal Processing VI, Veldhoven, Netherlands, October 1993, pp. 388–396 (1993)
17. Kulkarni, P., Gupta, P., Ercegovac, M.: Trading accuracy for power with an underdesigned multiplier architecture. In: Proceedings of the VLSID 2011: The Proceedings of 24th International Conference on VLSI Design, Chennai, India, January 2011, pp. 346–351 (2011)
18. Liu, C., Han, J., Lombardi, F.: A low-power, high-performance approximate multiplier with configurable partial error recovery. In: Proceedings of the DATE 2014: The Proceedings of IEEE Design and Test in Europe (DATE) Conference, March 2014, p. 95 (2014)
19. Kyaw, K.Y., Goh, W.L., Yeo, K.S.: Low-power high-speed multiplier for error-tolerant application. In: 2010 IEEE International Conference of Electron Devices and Solid-State Circuits (EDSSC), Hong Kong, China, December 2010, pp. 1–4 (2010)
20. Lu, S.-L.: Speeding up processing with approximation circuits. Computer **37**(3), 67–73 (2004)
21. Momeni, A., Han, J., Montuschi, P., Lombardi, F.: Design and analysis of approximate compressors for multiplication. IEEE Trans. Comput. **64**(4), 984–994 (2015)
22. Lin, C.H., Lin, I.C.: High accuracy approximate multiplier with error correction. In: Proceedings of the ICCD 2013: The 2013 IEEE 31st International Conference on Computer Design (ICCD), Asheville, NC, USA, October 2013, pp. 33–38 (2013)
23. Bhardwaj, K., Mane, P.S., Henkel, J.: Power-and area-efficient Approximate Wallace Tree Multiplier for error-resilient systems. In: Symposium ISQED 2014: 15th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, USA, March 2014, pp. 263–269 (2014)
24. Gupta, V., Mohapatra, D., Park, S.P., Raghunathan, A., Roy, K.: IMPACT: imprecise adders for low-power approximate computing. In: Proceedings of the 17th IEEE/ACM International Symposium on Low-Power Electronics and Design, Fukuoka, Japan, August 2011, pp. 409–414 (2011)
25. Yang, Z., Han, J., Lombardi, F.: Approximate compressors for error-resilient multiplier design. In: Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), pp. 183–186 (2015)

26. Shams, A.M., Darwish, T.K., Bayoumi, M.A.: Performance analysis of low-power 1-bit CMOS full adder cells. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **10**(1), 20–29 (2002)
27. Lau, M.S., Ling, K.V., Chu, Y.C.: Energy-aware probabilistic multiplier: design and analysis. In: Proceedings of the 2009 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, Grenoble, France, October 2009, pp. 281–290 (2009)