Approximate adder with reduced error

Balasubramanian, Padmanabhan; Maskell, Douglas Leslie; Prasad, K.

2019

Balasubramanian, P., Maskell, D. L., & Prasad, K. (2019). Approximate adder with reduced error. 31st International Conference on Microelectronics (MIEL 2019), 293-296.
doi:10.1109/MIEL.2019.8889605

<https://hdl.handle.net/10356/144131>

<https://doi.org/10.1109/MIEL.2019.8889605>

© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at:
<https://doi.org/10.1109/MIEL.2019.8889605>

Downloaded on 26 Apr 2023 20:03:32 SGT

Approximate Adder with Reduced Error

P. Balasubramanian, D.L. Maskell, and K. Prasad

Abstract – A new approximate adder is proposed, which is suitable for FPGA- and ASIC-based implementations. Here, we consider an Artix-7 FPGA for the implementations using Vivado 2018.3. For 32-bit addition, the proposed approximate adder with an 8-bit least significant inaccurate sub-adder reports an improvement in the maximum frequency by 7.7% compared to the native accurate FPGA adder while consuming 22% fewer LUTs and 18.6% fewer registers. For 64-bit addition, the proposed approximate adder reports an increase in the maximum frequency by 9.1% than the accurate FPGA adder while consuming 11% fewer LUTs and 9.3% fewer registers. The power-delay product (PDP) is computed as the product of total on-chip power consumption and the minimum clock period. The proposed approximate adder achieves 14.7% and 9.3% reductions in PDP compared to the accurate FPGA adder for 32- and 64-bit additions respectively. Further, in comparison with a recent approximate adder presented in the literature, the proposed approximate adder reports a 40% reduction in the root mean square error (RMSE) while having the same design metrics.

I. INTRODUCTION

Approximate computing, also called imprecise or inaccurate computing, is considered to be a potential alternative to conventional accurate computing due to its ability to improve the speed, reduce the area, minimize the power and optimize the energy, while providing acceptably correct computation results [1]. In practical applications such as multimedia [2], low power graphics processing [3], big data analytics [4], artificial intelligence and machine learning [5], neuromorphic computing [6] etc. approximate computation results subject to a specified error bound are considered as acceptable. Approximate arithmetic circuits [7] is an important topic that has been receiving significant attention from the VLSI design community.

This paper proposes a new approximate adder that is suitable for FPGA- and ASIC-based implementations. Although several approximate adders have been presented in the literature many of them are unsuitable for a FPGA-based implementation since they incorporate custom logic which are inefficient to realize using a FPGA. This is because the fast carry chain embedded in the FPGA fabric would not be utilized in the FPGA implementation of an ASIC-oriented approximate adder, which would result in

less speed and consumption of more FPGA resources compared to the native accurate FPGA adder, as remarked in [8]. For an example, we considered a 32-bit FPGA implementation of the ASIC-oriented error tolerant approximate adder of [9]. This approximate adder reports a maximum frequency of 454.545MHz (i.e., a clock period of 2.2ns), and requires 53 LUTs and 97 registers (flip-flops) for implementation. These design metrics were estimated post-place and route on an Artix-7 FPGA (part: xca100tcsg324-3) using Xilinx Vivado 2018.3 with the synthesis strategy set as Flow_AreaOptimized_high and with a default implementation strategy. Compared to the FPGA implementation of the approximate adder of [9], the 32-bit native accurate FPGA adder reports a 4.8% increase in the maximum frequency (476.19MHz) and requires only 32 LUTs (i.e., 40% fewer LUTs) while requiring the same number of registers. This suggests that ASIC-oriented approximate adders may not be suitable for an efficient FPGA-based implementation.

Further, approximate adders incorporating dynamic approximation where the degree of approximation may be varied and/or accurate outputs may be produced on demand may not be suitable for an efficient FPGA-based implementation either. This is because of the introduction of extra circuitry such as error detection and correction and/or carry prediction and control, and the likely need for invoking multiple clock cycles etc.

II. ACCURATE AND APPROXIMATE ADDERS

The block schematics of accurate and approximate adders are shown in Fig. 1. In Fig. 1, assuming an N-bit adder, A_{N-1} to A_0 and B_{N-1} to B_0 represent the input bits, and SUM_N to SUM_0 represents the sum output bits which includes the carry overflow. All the approximate adders in Fig. 1 incorporate an inaccurate sub-adder, which is least significant, and an accurate sub-adder which is more significant. K denotes the size of the inaccurate sub-adder and (N-K) denotes the size of the accurate sub-adder. In Fig. 1, 'Adder1' represents the accurate adder, and 'Adder2' to 'Adder6' represent the approximate adders, with 'Adder6' being the proposed approximate adder. References to the corresponding literature are provided for the approximate adders shown in Fig. 1.

As mentioned earlier, Adder1, shown in Fig. 1(a), represents the accurate adder.

Adder2 [10] is realized by combining an (N-K)-bits accurate sub-adder with a K-bits inaccurate sub-adder. The inaccurate sub-adder uses K 2-input OR functions. The OR functions perform logical disjunction of the corresponding

P. Balasubramanian and D.L. Maskell are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, E-mails: {balasubramanian, asdouglas}@ntu.edu.sg

K. Prasad is with the Department of Electrical and Electronic Engineering, Auckland University of Technology, Auckland 1142, New Zealand, E-mail: krishnamachar.prasad@aut.ac.nz

Fig. 1. (a) Accurate adder, and (b) to (f) are approximate adders. (f) is the proposed approximate adder.

input bit-pairs to produce the respective sum output bits. Bit-wise addition is performed in parallel in the inaccurate sub-adder with no connections between the 2-input OR functions. The most significant input bit-pair of the inaccurate sub-adder is AND-ed and is given as the carry input for the accurate sub-adder.

Adder3 [11] is similar to Adder2, with the exception that there is no carry input given from the inaccurate sub-adder to the accurate sub-adder; rather, the carry input of the accurate sub-adder is set to 0.

Adder4 [8] is a modified version of Adder3, which additionally contains a 2:1 multiplexer (MUX) in the inaccurate sub-adder. The most significant input bit-pair of the inaccurate sub-adder is AND-ed and its output serves as the selection input for the MUX besides serving as the carry input for the accurate sub-adder. If the MUX selection input is 0, the most significant sum bit of the inaccurate sub-adder, i.e., SUM_{K-1} is the OR of A_{K-1} and B_{K-1} , i.e., $(A_{K-1} \mid B_{K-1})$. On the other hand, if the MUX selection input is 1, SUM_{K-1} would assume 0.

Adder5 [12] is a modified version of Adder2 in that only two 2-input OR functions are used to perform logical disjunction of the two most significant input bit-pairs in the inaccurate sub-adder to produce the sum bits SUM_{K-1} and SUM_{K-2} . The remaining less significant $(K-2)$ sum bits of the inaccurate sub-adder are connected to the supply (V_{dd}) resulting in the production of a constant 1 for the $(K-2)$ sum bits viz. SUM_{K-3} to SUM_0 .

Adder6 is the proposed approximate adder, which is a modified version of the approximate adder, Adder4. The logic corresponding to the sum bits SUM_{K-1} and SUM_{K-2} of Adder6 are the same as Adder4. However, a constant 1 is set for the remaining less significant sum bits (SUM_{K-3} to SUM_0) by connecting these to V_{dd} , which would reduce the FPGA resources needed for their implementation.

III. FPGA IMPLEMENTATION RESULTS AND ERROR CHARACTERISTICS

In [13], the size of the inaccurate sub-adder considered for a practical video encoding application is limited to 8-bits. Typically, the degree of approximation is restricted to a range of 7- to 9-bits in an approximate adder. Hence, we considered $K = 8$ for the approximate adders shown in Figs. 1(b) to 1(f). However, the size of the inaccurate sub-adder may be varied based on need depending on a target application since the approximate adder architectures shown in Fig. 1 are generic. It may be noted that as the degree of approximation is increased, the speed of an approximate adder will increase because the critical path delay will decrease.

32- and 64-bit accurate and approximate adders were implemented on an Artix-7 FPGA (part: xc7a100tcs324-3) using Vivado 2018.3. The synthesis strategy was set to Flow_AreaOptimized_high, and a default implementation strategy was used. All the adders were implemented with a pair of registers on the adder inputs and a register on the

adder output. The adders were successfully synthesized and placed and routed, and the design metrics estimated are given in Table I. The design metrics include minimum clock period (in ns), maximum operating frequency (in MHz), number of LUTs and flip-flops (FFs, also called registers), total on-chip power consumption and the adder power consumption in Watts. The total on-chip power and adder power consumption given in Table I are the default estimates reported by Vivado after placement and routing.

TABLE I
FPGA-BASED DESIGN METRICS OF 32- AND 64-BIT ACCURATE AND APPROXIMATE ADDERS (ESTIMATED AFTER PLACE-AND-ROUTE)

Adder Name	Clock Period	Max. Freq.	# LUTs	# FFs	Power (W)	
					Total	Adder
32-bit Adders						
Adder1	2.10	476.19	32	97	0.209	0.117
Adder2	1.94	515.46	28	97	0.200	0.109
Adder3	1.92	520.83	28	97	0.200	0.109
Adder4	1.94	515.46	28	97	0.201	0.110
Adder5	1.95	512.82	25	79	0.191	0.100
Adder6	1.95	512.82	25	79	0.192	0.101
64-bit Adders						
Adder1	2.89	346.02	64	193	0.264	0.173
Adder2	2.77	361.01	60	193	0.260	0.168
Adder3	2.77	361.01	60	193	0.258	0.167
Adder4	2.77	361.01	60	193	0.261	0.169
Adder5	2.65	377.36	57	175	0.261	0.169
Adder6	2.65	377.36	57	175	0.261	0.170

From Table I, it is seen that for both 32- and 64-bit additions, Adder5 and Adder6 consume identical and also the least FPGA resources in comparison with the accurate adder and other approximate adders. Adder1, which is the native accurate FPGA adder, requires 32 LUTs and 97 FFs for realizing 32-bit addition and 64 LUTs and 193 FFs for realizing 64-bit addition. Adder2 to Adder4 require the same number of FFs as Adder1. However, they require 4 LUTs less compared to Adder1 for realizing 32- and 64-bit additions. Adder5 and Adder6 require the least LUTs and FFs amongst all. This is mainly because the six least significant sum bits of the inaccurate sub-adder in Adder5 and Adder6 are connected to V_{dd} , and they do not consume the FPGA resources. In terms of the maximum operating frequency, Adder5 and Adder6 report increases over the accurate adder by 7.7% for 32-bit addition and 9.1% for 64-bit addition. Also, Adder5 and Adder6 report higher speed over other approximate adders for 64-bit addition.

In Table I, the total on-chip power includes the power consumed by the clock, logic, input-output (IO), signals, and the device static power component. It is noticed that the adder power consumption is approximately half of the total on-chip power consumption. We compute the power-delay product (PDP) as the product of total on-chip power

consumption and the clock period, and the PDP serves as a representative metric for low power. Based on the PDP calculations, it is noted that Adder5 and Adder6 report reductions in PDP compared to the accurate FPGA adder by 14.7% for 32-bit addition and 9.3% for 64-bit addition. From the combined perspectives of all the design metrics, it is observed that Adder5 and Adder6 are preferable to the accurate and other approximate adders.

We analyzed the error range of the approximate adders by considering a K-bit inaccurate sub-adder and computed the RMSE by assuming $K = 8$. The error characteristics are given in Table II. We compute the root mean square error (RMSE) for the approximate adders using (1) [8], as RMSE gives a relatively higher weight to larger errors, which is more likely to impact a practical application employing approximate computer arithmetic.

$$\text{RMSE} = \sqrt{\frac{1}{2^{2N}} \sum_{j=0}^{2^{2N}-1} e_j^2} = \sqrt{\sum_{\delta} e_{\delta}^2 \cdot P_{\delta}} \quad (1)$$

Where N is the adder bit-width, e is the error (i.e., the difference between the accurate and the approximate adder outputs), P is the probability of an error value occurring, and δ is the set of all error values.

TABLE II
ERROR RANGE AND RMSE OF APPROXIMATE ADDERS

Adder Legend	Error Range	RMSE
Adder1	Not applicable	Nil
Adder2	$-(2^{K-1}-1)$ to 2^{K-1}	64
Adder3	$-(2^K-1)$ to 0	90.33
Adder4	$-(2^{K-1}-1)$ to 0	45.08
Adder5	$-(2^{K-1}-1)$ to $(2^{K-1}+2^{K-2}-1)$	69.13
Adder6	$-(2^{K-1}-1)$ to $(2^{K-2}-1)$	41.31

Adder1 is accurate and hence its error characteristics are nil, by default. The error range mentioned in Table II gives an indication that Adder6 has a relatively better normal error distribution with respect to zero error compared to the other approximate adders. Moreover, Adder6 has a reduced RMSE compared to the other approximate adders. It was noted from Table I that Adder5 and Adder6 feature the same design metrics for 32- and 64-bit additions, but from Table II, it is seen that Adder6 has a 40% less RMSE compared to Adder5.

IV. CONCLUSIONS

This paper presented a new approximate adder which is suitable for FPGA- and ASIC-based implementations. The new approximate adder, in comparison with the native accurate FPGA adder, reports an increase in the maximum frequency while consuming fewer LUTs and registers for

32- and 64-bit additions. Also, comparisons with other approximate adders reveal that Adder6 has optimized design metrics and a reduced error characteristic.

ACKNOWLEDGEMENT

This research is funded by the Ministry of Education (MOE), Singapore under grant MOE2018-T2-2-024.

REFERENCES

- [1] K. Roy, and A. Raghunathan, "Approximate computing: an energy-efficient computing technique for error resilient applications", in *Proc. IEEE Computer Society Annual Symposium on VLSI ISVLSI 2015*, France, 2015, pp. 473-475.
- [2] M.A. Breuer, "Multi-media applications and imprecise computation", in *Proc. 8th Euromicro Conference on Digital System Design DSD 2005*, Portugal, 2005, pp. 2-7.
- [3] H. Zhang, M. Putic, and J. Lach, "Low power GPGPU computation with imprecise hardware", in *Proc. 51st Design Automation Conference DAC 2014*, USA, 2014, pp. 1-6.
- [4] R. Nair, "Big data needs approximate computing: technical perspective", *Comm. of the ACM*, 2015, vol. 58, pp. 104.
- [5] S.S. Sarwar, G. Srinivasan, B. Han, P. Wijesinghe, A. Jaiswal, P. Panda, A. Raghunathan, and K. Roy, "Energy efficient neural computing: a study of cross-layer approximations", *IEEE Journal of Emerging and Selected Topics in Circuits and Systems*, 2018, vol. 8, pp. 796-809.
- [6] P. Panda, A. Sengupta, S.S. Sarwar, G. Srinivasan, S. Venkataramani, A. Raghunathan, and K. Roy, "Cross-layer approximations for neuromorphic computing: from devices to circuits and systems", in *Proc. 53rd Design Automation Conference DAC 2016*, USA, 2016, pp. 1-6.
- [7] H. Jiang, C. Liu, L. Liu, F. Lombardi, and J. Han, "A review, classification, and comparative evaluation of approximate arithmetic circuits", *ACM Journal on Emerging Technologies in Computing Systems*, 2017, vol. 13, pp. 60:1-60:34.
- [8] P. Balasubramanian, and D. Maskell, "Hardware efficient approximate adder design", in *Proc. IEEE Region 10 Conference TENCON 2018*, South Korea, 2018, pp. 806-810.
- [9] N. Zhu, W.L. Goh, and K.S. Yeo, "An enhanced low-power high-speed adder for error-tolerant application", in *Proc. 12th International Symposium on Integrated Circuits ISIC 2009*, Singapore, 2009, pp. 69-72.
- [10] 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*, 2010, vol. 57, pp. 850-862.
- [11] P. Albicocco, G.C. Cardarilli, A. Nannarelli, M. Petricca, and M. Re, "Imprecise arithmetic for low power image processing", in *Proc. 46th Asilomar Conference on Signals, Systems and Computers*, USA, 2012, pp. 983-987.
- [12] A. Dalloo, A. Najafi, and A. Garcia-Ortiz, "Systematic design of an approximate adder: the optimizer lower part constant-OR adder", *IEEE Transactions on VLSI Systems*, 2018, vol. 26, pp. 1595-1599.
- [13] A. Raha, H. Jayakumar, and V. Raghunathan, "Input-based dynamic reconfiguration of approximate arithmetic units for video encoding", *IEEE Transactions on VLSI Systems*, 2016, vol. 24, no. 3, pp. 846-857.