QuAd: Design and Analysis of Quality-Area Optimal Low-Latency Approximate Adders

Muhammad Abdullah Hanif[1,2], Rehan Hafiz[2], Osman Hasan[3], Muhammad Shafique[1]
[1]Vienna University of Technology, Austria
[2]Information Technology University, Lahore, Pakistan
[3]National University of Sciences and Technology, Islamabad, Pakistan
{abdullah.hanif,rehan.hafiz}@itu.edu.pk, Osman.hasan@seecs.edu.pk, muhammad.shafique@tuwien.ac.at

ABSTRACT

Approximate circuits exploit error resilience property of applications to tradeoff computation quality (accuracy) for gaining advantage in terms of performance, power, and/or area. While state-of-the-art low-latency approximate adders provide an accuracy-area-latency configurable design space, the selection of a particular configuration from the design space is still manually done. In this paper, we analytically analyze different structural properties of low-latency approximate adders to formulate a new adder model, Quality-area optimal Low-Latency approximate Adder (QuAd). It provides an increased design space as compared to state-of-the-art, providing design points that require less logic area for the same accuracy, as compared to state-of-the-art approximate adders. Furthermore, based upon our mathematical analysis, we show that, provided a latency constraint, an adder configuration with the highest quality and lowest area requirement can effortlessly be selected from the whole design space of QuAd adder model, without requiring any optimization strategy or numerical simulation. Our experimental results validate the developed model and also the quality-area optimality of our optimal QuAd adder configuration. For functional verification and prototyping, we have used a Xilinx Virtex-6 FPGA. RTL/behavioral models and MATLAB equivalent scripts, of our proposed adder model are made open source, to facilitate further research and development.

1 INTRODUCTION AND RELATED WORK

Approximate Computing is an evolving computing paradigm that relies on trading-off the computational quality (accuracy) to provide new opportunities for improving the area, power, and performance efficiency of systems. Recent, investigations by Intel [1], IBM [2], Microsoft [3][4], and other research groups [5] have shown that there indeed exists a number of compute-intensive applications that can tolerate approximation errors while still producing outputs that are useful and of acceptable quality for the end-users[12]. Particularly, applications such as image/video/vision processing, machine/deep learning, big data analytics, recognition, web searches and signal processing, are either inherently prone to noise or are resilient to error because of the perceptual limitations of the end-users and hence are natural candidates for approximate computing.

Adders are one of the fundamental arithmetic units and have gained significant attention from the approximate computing community [6]-[11][16]. Carry computation typically forms the critical path for an N-bit two-operand adder. Most state-of-the-art low-latency approximate adders (such as, ACA [6][7], ETA, ETA-II, ETA-IIM [9], GDA [8], ESA [16], and GeAr [10]) rely on the observation that in most cases the longest carry propagation chain is less than the complete length (*N*) of the adder. Thus, the approximate designs reduce this critical path by employing multiple smaller disjoint or overlapping *L*-bit sub-adders (with *L*<*N*). Thereby achieves reduced latency at the cost of increased area (in case of overlapping sub-adders). Each sub-adder is composed of two types of bits, the Resultant bits (*R* bits) which produces sum bits that contribute to the final summation and Prediction bits (*P* bits) that utilized *P* previous bits for predicting carry for *R* bits. Only for the first sub-adder, all the bits are considered as *R* bits since carry-in is generally known. ACA-I [6] employed the use of multiple overlapping fixed-length sub-adders with *R=1.* ETAII [9] made use of the carry generated by the carry prediction unit of one previous sub-adder for predicting the carry-in of current sub-adder, thus *R=P*. ETAIIM [9] allowed the concatenation of carry prediction logic of any, but not the least significant sub-adder, to increase the accuracy. In ACA-II [7], the length of *R* for each sub-adder was set to half of sub-adder length, *L*. GDA [8] provided a configurable approximate adder that used multiple non-overlapping sub-adders of length *R* and used multiplexers for carry selection from either the previous sub-adder or from the carry-in prediction block. Thus, for any sub-adder the number of prediction bits were a multiple of *R*. GeAr [10] provided a unified design space by providing a configurable approximate adder along with its associated error probability model. GeAr adder model allows any combination for *R* and *P*, provided the length of sub-adder *(L = R+P)* is uniform throughout the adder and thereby covers many low-latency adders like ESA [16], ACA[6][7], etc. Furthermore, all sub-adders must have the same value for *R* and *P*. *The aforementioned condition limits the number of possible configurations that can be realized using the GeAr adder.*

Limitations of State-of-the-Art: It is noteworthy that in all of these approximate adders, their design imposes some specific restriction on the length of sub-adders, the number of sum bits that each sub-adder produces, and/or the number of carry prediction bits it utilizes for the generation of the sum bits. Due to these restrictions, the design space of such low-latency approximate adders overlooks several configurations, which may require lower logic area for the same accuracy, i.e., the real Pareto-optimal points in the design space.

Motivational Analysis: Consider an example where we would like to develop an 8-bit approximate adder and the maximum allowed latency is equivalent to that of a 7-bit Ripple Carry Adder (RCA) based sub-adder. Fig. 1 presents the values of the Accuracy-per-Area (denoted as Accuracy/Area) metric, for the complete design space, plotted against Accuracy, assuming

DAC '17, June 18-22, 2017, Austin, TX, USA
© 2017 ACM. ISBN 978-1-4503-4927-7/17/06 $15.00
DOI: http://dx.doi.org/10.1145/3061639.3062306

linear dependency of logic area on the sub-adder's length. Here, accuracy is computed as 1-Normalized Error Distance (NED).

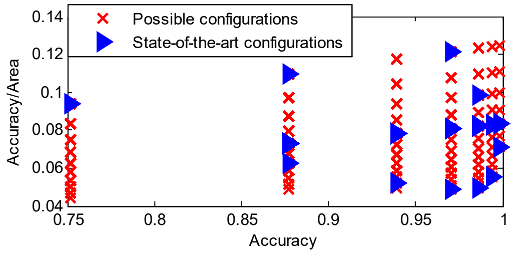


Figure 1. Design Space of an 8-bit approximate adder using sub-adders of length less than or equal to 7-bits

Fig. 1 provides the configurations supported by the state-of-the-art adders (represented by "►") including GeAr, ETA and GDA. Fig. 1 also provides adder configurations (represented by "x") that are still not supported by any of these prior adder configurations. This include combinations, such as sub-adders of varying sizes and/or other arbitrary combinations of *R* and *P* bits for each sub-adder. It can also be observed that these missing design configurations may provide design points that require less logic area while providing similar accuracy level as that of state-of-the-art.

Required: Thus, an adder model is required that is able to provide an enhanced design space, which includes the adder configuration with the *optimal* Accuracy-per-Area value for a particular latency constraint. Furthermore, the extended design space requires a technique to select an optimal configuration, which meets the latency requirement, provides maximum accuracy and still requires the least logic area.

Novel Contributions in a Nutshell: In this paper, we present QuAd: A Quality-Area optimal Low-Latency Approximate Adder model and its corresponding mathematical analysis that:

1. Allows using sub-adder units of varying lengths with arbitrary combinations of resultant and prediction bits. The extended design space not only includes all of the available low-latency adders but it further provides new Pareto-optimal design points.
2. Provides an analytic solution to find a quality-area optimal approximate adder configuration while assuming inputs to be independent and uniformly distributed, given a user-/designer-provided latency constraint.

2 QUAD: A QUALITY-AREA OPTIMAL LOW-LATENCY APPROXIMATE ADDER MODEL

For our QuAd model, each sub-adder can have any number of Prediction (P) and Resultant (R) bits, regardless of the number of P and R bits in other sub-adders. An N bit QuAd adder comprising k sub-adders is, therefore, defined using a resultant vector, $R_{vect} = [R_1, R_2, \ldots, R_k]$ and a prediction vector,$P_{vect} = [P_1, P_2, \ldots, P_k]$ where, R_i and P_i ($\forall\ i \in 1, 2, 3, \ldots, k$) represent the number of resultant and prediction bits in the i^{th} sub-adder, respectively. The resultant bits from each sub-adder constitute the N bit output of the adder, hence $N = \sum_{j=1}^{k} R_i$. Thus, the generic QuAd representation, $QuAd\{[R_1, R_2, \ldots, R_k], [P_1, P_2, \ldots, P_k]\}$ completely specifies any possible adder configuration. Fig. 2 shows a generic representation of an N-bit QuAd adder and a constituent i^{th} sub-adder unit. The first sub-adder does not require any prediction bits, i.e. $P_1 = 0$.

Unlike earlier adder models, R_i can take any value between 1 to $(N - \sum_{j=1}^{j=i-1} R_j)$. While P_i can theoretically take any value between 0 to $\sum_{j=1}^{i-1} R_j$, we propose to restrict P_i such that $P_i < R_{i-1} + P_{i-1}$, i.e., the number of prediction bits in i^{th} sub-adder should be less than the length of $i - 1^{th}$ sub-adder. While this may appear to be a restriction on P we show that the corresponding configurations with $P_i = R_{i-1} + P_{i-1}$ or $P_i > R_{i-1} + P_{i-1}$ require more area than the case when $P_i < R_{i-1} + P_{i-1}$ while providing equal/lower accuracy measure. Below, we compare the Probability Mass Function (PMF) of approximation error for these three possibilities in order to identify such sub-optimal configurations.

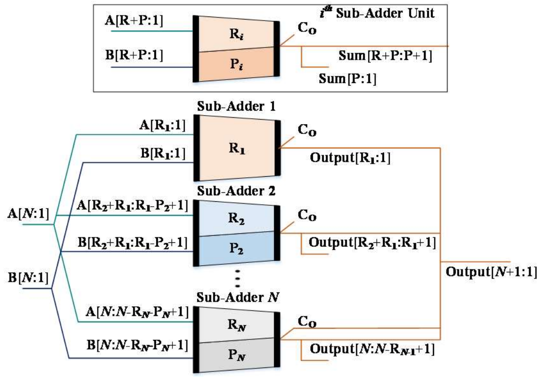


Figure 2. A generic *N*-bit QuAd adder. Each i^{th} QuAd sub-adder sums two L_i-bit numbers, where the first *P* bits of both the operands are used to predict carry-in for computing the sum of *R* significant bits.

1. $\boldsymbol{P_i = P_{i-1} + R_{i-1}}$: Fig. 3(a) provides a configuration in which $P_3 = R_2 + P_2$ and also provides its error PMF. The error PMF is defined as $P_E = P(E = e_w)$ where e_w is the error magnitude that can have any value between 0 (no error) to 2^N. For this configuration, the only possible error magnitude is $2^{R1} = 2^4$. This is because the error occurs only when $(R_1 - P_2)$ least significant bits generate a carry and P_2 or P_3 bits propagate it. Due to their overlapping structure, for P_3 to be in propagate mode, P_2 must be in propagate mode and hence the effective error magnitude for both the cases is 2^4.
2. $\boldsymbol{P_i < P_{i-1} + R_{i-1}}$: For each configuration having $P_i = P_{i-1} + R_{i-1}$ (for any $i \in 2,3, \ldots, k$), there exists an alternate configuration in which $i^{th}$and $i - 1^{th}$ sub-adders can be replaced by a single sub-adder having resultant and prediction bits equal to $R_i + R_{i-1}$ and P_{i-1}, respectively. The resultant configuration provides equal accuracy while utilizing lesser area. Fig. 3(b) provides such an alternative configuration for that of Fig. 3(a). Despite using lesser total bits for prediction, and hence requiring lesser logic area, the associated P_E is identical.
3. $\boldsymbol{P_i > P_{i-1} + R_{i-1}}$: Fig. 3(c) illustrates an instance where the propagation bits of the i^{th} sub-adder are extended even beyond the length of $i - 1^{th}$ sub-adder. In this particular case; another error term with a magnitude of $|2^{R1+R2} - 2^{R1}|$ is introduced in P_E due to the case when first $P_3 - R_2 + P_2$ bits of P_3 are in generate mode while the rest of its bits are in propagate mode.

The loss in accuracy experienced in the configuration of Fig. 3(c) as compared with that of Fig. 3(a), is due to the lower number of bits being used for the prediction of carry-in for R_2 bits. Therefore,

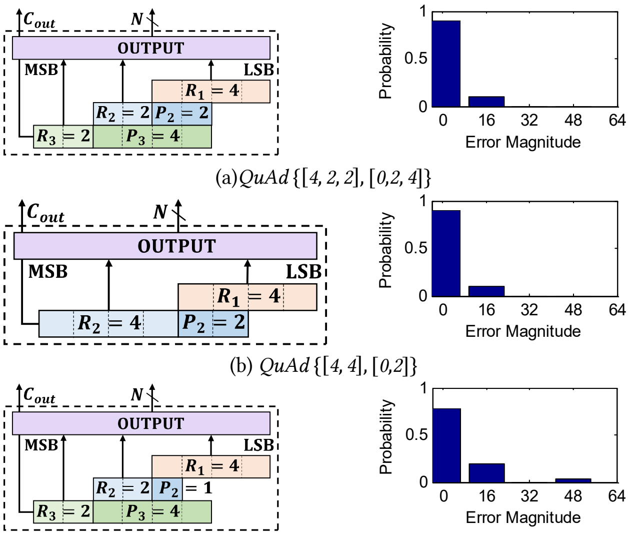
a. the configurations with $P_i = P_{i-1} + R_{i-1}$ provide better accuracy as compared to similar configurations with $P_i > P_{i-1} + R_{i-1}$; and

b. the configurations with $P_i < P_{i-1} + R_{i-1}$ provide better accuracy as compared to similar configurations with $P_i = P_{i-1} + R_{i-1}$, while requiring lesser logic area.

In summary, the configurations, with $P_i \geq P_{i-1} + R_{i-1}$ for any $i \in 2, 3, \ldots, k$, can be eliminated based on the fact that there is an alternative configuration (with $P_i < R_{i-1} + P_{i-1}$), which generates better/equivalent results while utilizing lesser amount of resources. We eventually propose that in QuAd, $P_i < R_{i-1} + P_{i-1}$.

2.1 Quality-Area Optimal QuAd adder configuration

Provided a latency constraint in terms of maximum allowed sub-adder length, L_{max}, we define, $QuAd_o(N, L_{max})$, i.e., an *N*-bit Quality-Area Optimal adder configuration that provides the highest accuracy while requiring least logic area, from among the complete QuAd design space. We use two metrics, i.e., Mean Square Error (MSE) and Mean Error Distance (MED), to measure the accuracy. The reasons of selecting these metrics are: (a) MSE is a standard quality metric for many multimedia applications [14] and is inversely proportional to Peak signal-to-noise ratio (PSNR), which is another widely used quality metric; and (b) MED is considered as an effective measure for computing the accuracy of a multi-bit approximate adder [13].



(a)*QuAd* {[*4, 2, 2*], [*0,2, 4*]}

(b) *QuAd* {[*4, 4*], [*0,2*]}

(c) *QuAd* {[*4, 2, 2*], [*0,1,4*]}

Figure 3. Architectural design of the three possible types of configurations along with their respective error PMFs.

In the following, we present two key properties and their proofs, which will facilitate our $QuAd_o(N, L_{max})$ adder design. It is noteworthy that these properties are also valid for state-of-the-art low-latency approximate adders since our QuAd's design space includes all of their configurations.

Property-I: For the most significant sub-adder, the configurations with the least number of *P* bits and maximum possible sub-adder length, provides lower values for MSE and MED.

Fig. 4(a) shows an *N*-bit low-latency approximate adder built using two sub-adders. The associated MED and MSE are given by:

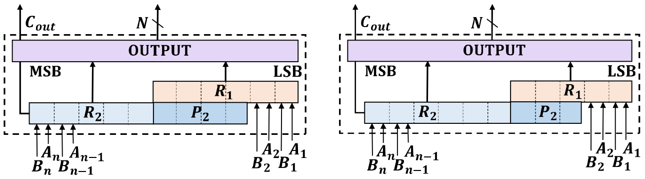
$$MED_1 = P[E]_1 * E_1$$
$$MSE_1 = P[E]_1 * (E_1)^2$$

Here, $P[E]_1$ is the probability of error while $E_1 = |Value_{observed} - Value_{true}|$ is the error magnitude. Subscript "1" refers to the configuration 1 of Fig. 4 for ease of reference. The error probability associated with this configuration can be defined as the probability with which $(R_1 - P_2)$ least significant bits generates a carry and rest of the bits corresponding to P_2 bits propagates it. Assuming, inputs with uniform distribution, the probability can mathematically be given as:

$$P[E]_1 = \rho[pr]^{P_2} * \textstyle\sum_{i=0}^{R_1 - P_2 - 1} \rho[gr] * \rho[pr]^i$$

Here, $\rho[gr] = P[a_i = 1 \,\&\, b_i = 1]$ and $\rho[pr] = P[(a_i = 1 \,\&\, b_i = 0) \; or \; (a_i = 0 \,\&\, b_i = 1)]$ defines the probability of carry generation and carry propagation respectively (a_i and b_i are the i^{th} bits of operands *A* and *B* respectively). The magnitude of error in the representative configuration of Fig. 4(a) is equivalent to the magnitude of carry-out from the first sub-adder and hence $E_1 = 2^{R1}$. Thus,

$$MED_1 = 2^{R_1} * \rho[pr]^{P_2} * \textstyle\sum_{i=0}^{R_1 - P_2 - 1} \rho[gr] * \rho[pr]^i$$
$$MS\ _1 = 2^{2R_1} * \rho[pr]^{P_2} * \textstyle\sum_{i=0}^{R_1 - P_2 - 1} \rho[gr] * \rho[pr]^i$$



(a) Configuration I: *QuAd*{[6,5], [0,4]}

(b) Configuration II: *QuAd*{[5,6], [0,3]}

Figure 4. Structural comparison of two alternate QuAd configurations

Now, if we amend the configuration of Fig. 4(a) by decreasing the *P* bits of most significant sub-adder by 1, while keeping its width fixed, we get the configuration of *QuAd*{[5,6], [0,3]} Fig. 4(b). The decreased number of *P* bits results in an increase in the probability of error while reducing the magnitude of the error by the same ratio for uniformly distributed inputs. The $P[E]$ and E for this 2nd configuration (in terms of the variables of 1st configuration) are given by:

$$P[E]_2 = \rho[pr]^{P_2 - 1} * \textstyle\sum_{i=0}^{R_1 - P_2 - 1} \rho[gr] * \rho[pr]^i$$
$$E_2 = 2^{R_1 - 1}$$

Similarly, the MED and MSE for this new configuration is given by:

$$ME\ _2 = 2^{R_1 - 1} * \rho[pr]^{P_2 - 1} * \textstyle\sum_{i=0}^{R_1 - P_2 - 1} \rho[gr] * \rho[pr]^i$$
$$= \frac{ME\ _1}{2 * \rho[pr]}$$
$$MSE_2 = 2^{2R_1 - 2} * \rho[pr]^{P_2 - 1} * \textstyle\sum_{i=0}^{R_1 - P_2 - 1} \rho[gr] * \rho[pr]^i$$
$$= \frac{1}{2}\left(\frac{MSE_1}{2 * \rho[pr]}\right)$$

Assuming a uniform input distribution, *MED₁* is equivalent to *MED₂* since $2 * \rho[pr] = 1$. However, the MSE of the altered configuration is half of $MS\ _1$. Similarly, If we keep on decreasing the *P* bits of the most significant adder, we keep on getting a configuration with lower MSE and same MED. Furthermore, lower prediction bits means lower overlap and hence lower area requirement as evident from Fig. 4. Also, using the aforementioned equations, it can be concluded that the configurations having larger most significant sub-adders show lower MSE and MED values. *Thus, a QuAd configuration with Pk = 0 and Rk=Lmax, shall provide the lowest MSE and MED with least area requirement.*

Property-II: MSE and MED are irrespective to the configuration of remaining (*N*-L_{max}) least significant bits.

The previous property dictated that, in case of an approximate adder composed of two sub-adders, minimum MSE and MED is achieved when a non-overlapping L_{max}-bit sub-adder is used at most significant location, i.e. R_k=L_{max} and P_k=0, while using an accurate sub-adder at the least significant location. Using the configurations of Fig. 5, we show that the decomposition of the least significant sub-adder into further non-overlapping sub-adders has no effect on the overall MSE and MED of the approximate adder.

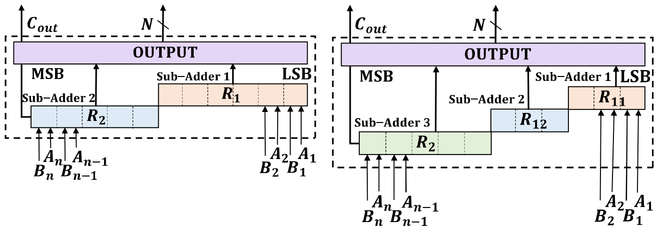
The *P*[*E*], *E*, MED and MSE for Configuration 3, Fig. 5(a), are given as:

$$P[E]_3 = \sum_{i=0}^{R_1-1} \rho[gr \ \ * \rho[pr \ ^{i}$$

$$E_3 = 2^{R_1}$$

$$MED_3 = 2^{R_1} * \sum_{i=0}^{R_1-1} \rho[gr] * \rho[pr \ ^{i}$$

$$MSE_3 = 2^{2R_1} * \sum_{i=0}^{R_1-1} \rho[gr \ \ * \rho[pr \ ^{i} \qquad (1)$$



(a) Configuration III: *QuAd* {[R_1, R_2], [0,0]}

(b) Configuration IV: *QuAd* {[R_{11}, R_{12}, R_2], [0,0,0]}

Figure 5. Structural comparison of two alternate QuAd configurations

Fig. 5(b) provides a case where the least significant sub-adder of Configuration 3 is sub-divided into two sub-adders, s1 and s2, with lengths equal to R_{11} and R_{12} bits, respectively. The probability with which these sub-adders induce error in the output can be given by:

$$P_{s1} = \sum_{i=0}^{R_{11}-1} \rho[gr \ \ * \rho[pr]^{i}$$

$$P_{s2} = \sum_{i=0}^{R_{12}-1} \rho[gr] * \rho[pr]^{i}$$

Note that, P_{s1} and P_{s2} is equal to the probability of carry-out of s1 and s2, respectively. Since, there are multiple sub-adders, there are multiple possible error magnitude values. The probability of these error magnitudes is provided in Table 1.

Table 1: Probability of error $P[E]_4$

Magnitude of Error	Probability
$2^{R_{11}}$	$P_{s1} - P_{s1} * P_{s2}$
$2^{R_{11}+R_{12}}$	$P_{s2} - P_{s1} * P_{s2}$
$2^{R_{11}} + 2^{R_{11}+R_{12}}$	$P_{s1} * P_{s2}$

The MED can be expressed using these probabilities as follows

$$MED_4 = (P_{s1} - P_{s1} * P_{s2}) * 2^{R_{11}} + \ (P_{s2} - P_{s1} * P_{s2}) * 2^{R_{11}+R_{12}} + \ (P_{s1} * P_{s2}) * (2^{R_{11}} + 2^{R_{11}+R_{12}})$$

$$MED_4 = \ P_{s1} * 2^{R_{11}} + P_{s2} * 2^{R_{11}+R_{12}} \qquad (2)$$

We multiply the first term in the above equation by $2^{R_{12}} * \rho[pr]^{R_{12}}$. This does not violate the equation since $2^{R_{12}} * \rho[pr]^{R_{12}}$ is equivalent to 1 for uniform distribution as $\rho[pr] = 0.5$. So,

$$MED_4 = P_{s1} * 2^{R_{11}} * \rho[pr]^{R_{12}} * 2^{R_{12}} + P_{s2} * 2^{R_{11}+R_{12}}$$
$$MED_4 = 2^{R_{11}+R_{12}} * (P_{s1} * \rho[pr]^{R_{12}} + P_{s2})$$

which is equivalent to the MED3, since $R_1 = R_{11} + R_{12}$ and,

$$\begin{aligned} &P_{s1} * \rho[pr]^{R_{12}} + P_{s2} \\ &= \rho[pr]^{R_{12}} * \sum_{i=0}^{R_{11}-1} \rho[gr] * \rho[pr]^{i} + \sum_{i=0}^{R_{12}-1} \rho[gr] * \rho[pr]^{i} \\ &= \sum_{i=R_{12}}^{R_{11}+R_{12}-1} \rho[gr] * \rho[pr]^{i} + \sum_{i=0}^{R_{12}-1} \rho[gr] * \rho[pr]^{i} \\ &= \sum_{i=0}^{R_1-1} \rho[gr] * \rho[pr]^{i} \qquad (3) \end{aligned}$$

Similarly, the MSE of Fig. 5(b) can be defined utilizing Table 1 as:

$$MSE_4 = (P_{s1} - P_{s1} * P_{s2}) * 2^{2R_{11}} + \ (P_{s2} - P_{s1} * P_{s2}) * 2^{2(R_{11}+R_{12})} + \ (P_{s1} * P_{s2}) * (2^{R_{11}} + 2^{R_{11}+R_{12}})^2 \qquad (4)$$

By expanding $(2^{R_{11}} + 2^{R_{11}+R_{12}})^2$

$$MSE_4 = \ P_{s2} * 2^{2(R_{11}+R_{12})} + P_{s1} * 2^{2R_{11}} + \ (P_{s1} * P_{s2}) * \ (2 * 2^{R_{11}} * 2^{R_{11}+R_{12}})$$

Taking $P_{s1} * \rho[pr]^{R_{12}} * 2^{2(R_{11}+R_{12})}$ common from the last two terms we get:

$$MSE_4 = P_{s2} * 2^{2(R_{11}+R_{12})} + P_{s1} * \rho[pr]^{R_{12}} * 2^{2(R_{11}+R_{12})} * (\frac{1}{\rho[pr]^{R_{12} * 2^{2R_{12}}}} + \frac{2 * P_{s2}}{\rho[pr]^{R_{12} * 2^{R_{12}}}})$$

As for the uniform input distribution $\rho[pr]^{R_{12}} * 2^{R_{12}} = 1$, the aforementioned equation can be simplified to:

$$MSE_4 = P_{s2} * 2^{2(R_{11}+R_{12})} + P_{s1} * \rho[pr]^{R_{12}} * 2^{2(R_{11}+R_{12})} * \left(\frac{1}{2^{R_{12}}} + 2 * P_{s2}\right) \qquad (5)$$

Now, for uniform distribution, P_{s2} can be expanded as:

$$P_{s2} = \sum_{i=0}^{R_{12}-1} \rho[gr] * \rho[pr]^{i} = \frac{1}{2^2}\left(1 + \frac{1}{2^1} + \frac{1}{2^2} + \cdots + \frac{1}{2^{R_{12}-1}}\right) \qquad (6)$$

Substituting P_{s2} from (6) in the last terms of (5), we get:

$$MSE_4 = P_{s2} * 2^{2(R_{11}+R_{12})} + P_{s1} * \rho[pr]^{R_{12}} * 2^{2(R_{11}+R_{12})} * (\frac{1}{2^{R_{12}}} + 2 * \frac{1}{2^2}\left(1 + \frac{1}{2^1} + \frac{1}{2^2} + \cdots + \frac{1}{2^{R_{12}-1}}\right)) \qquad (7)$$

Noting that, $\left(\frac{1}{2^{R_{12}}} + 2 * \frac{1}{2^2}\left(1 + \frac{1}{2^1} + \frac{1}{2^2} + \cdots + \frac{1}{2^{R_{12}-1}}\right)\right) = 1$, and further simplifying (7), we get:

$$MSE_4 = 2^{2(R_{11}+R_{12})} * (P_{s2} + P_{s1} * \rho[pr]^{R_{12}}) \qquad (8)$$

Substituting the 2nd term in (8) using (3), and noting that $R_1 = R_{11} + R_{12}$, we prove that (8) is equivalent to (1), therefore, the configurations of Fig. 5(a) and Fig. 5(b) have equal MSE and MED measures.

Summarizing, from property I, we know that, in order to get minimum MED and MSE while utilizing minimum area we can divide an adder into two non-overlapping sub-adders, i.e., the most significant sub-adder and least significant sub-adder, where the length of the most significant sub-adder should be equal to L_{max}. With the help of property-II, we can further sub-divide the least significant sub-adder into multiple sub-adders of length less than or equal to L_{max} without affecting the overall MED, MSE, and area. We further propose to use such a configuration that provides the least value for maximum error magnitude, $Ma \ \ _E$. For the case of configurations with disjoint non-overlapping sub adders, the maximum error occurs when carry is generated from all the *(k-1)* least significant sub-adders and is given by:

$$Max_E = \sum_{i=1}^{k-1} 2\hat{}(\sum_{j=1}^{i} R_j)$$

Max_E is minimum when there are fewest possible sub-adders of maximum length, placed at most significant locations, since,

$$\sum_{i=1}^{M-1} 2^\wedge(\sum_{j=1}^{i} R_j) < \sum_{i=1}^{N-1} 2^\wedge(\sum_{j=1}^{i} R_j)\ for\ any\ M < N \quad (9)$$

Therefore, we define our latency-constrained, quality-area optimal adder, $QuAd_o(N, L_{max})$ as:

$$QuAd_o(N, L_{max}) = QuAd\{[(N\%L_{max}), L_{max}.., L_{max}], [0, \ldots, 0]\}$$

Fig. 6 illustrates the $QuAd_o(N, L_{max})$ configuration for various values of N and L_{max}. As suggested by our mathematical analysis and later confirmed by our experimental results, $QuAd_o(N, L_{max})$ always provides a configuration with minimum MSE, MED and $Ma\ _E$ from among the complete QuAd design space.

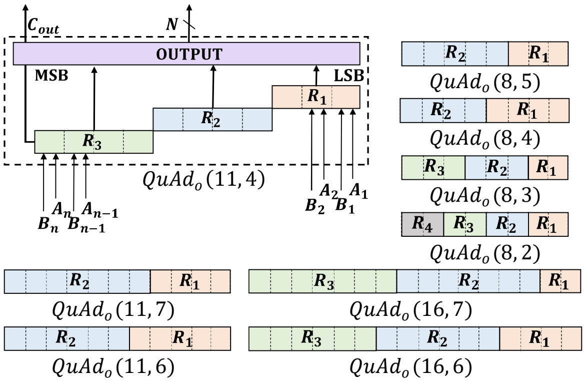


Figure 6: QuAd$_o$ configurations for 11, 16 and 8 bit approximate adders for multiple L_{max} values.

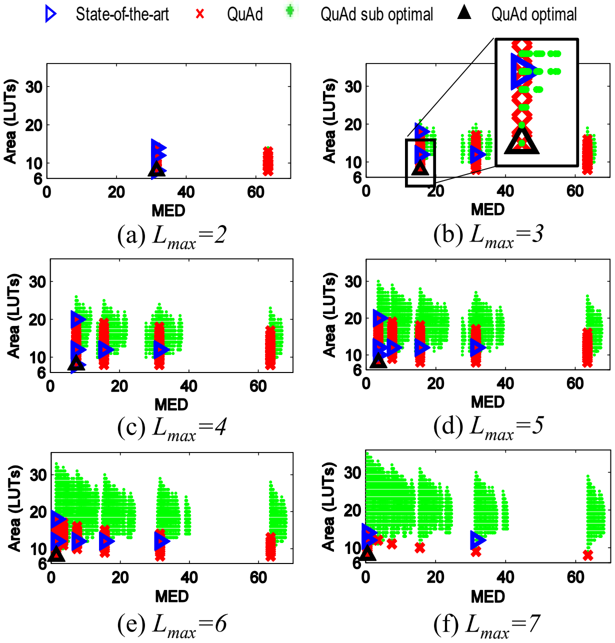
3 RESULTS AND DISCUSSION

We compare the extended design space of the proposed QuAd adder model to that of the state-of-the-art approximate adders. We show that our optimized $QuAd_o(N, L_{max})$ adder indeed provides a quality-area optimal adder configuration for a given latency constraint. The area and latency results for adder configurations are obtained by synthesizing their VERILOG models for XILINX Virtex 6 XC6VLX75T FPGA using Xilinx ISE. Sub-adders were implemented using the Ripple Carry Adder (RCA) since *current FPGAs use dedicated carry chains for their efficient implementation.* However, note that our QuAd model is not specific to any particular sub-adder implementation. Thus, unlike FPGAs if for an ASIC implementation an n-bit CLA is considered faster as compared to an RCA, sub-adder unit of the GeAr may comprise a CLA. Similarly, for FPGAs, LUT based fast adders such as [15]. Functional models for QuAd are also developed in MATLAB to compute accuracy values (MSE, MED) by exhaustive simulations and to evaluate performance in real applications. We have made these RTL and MATLAB implementations open source at https://sourceforge.net/projects/quad-code/ for reproducibility of results and to facilitate further research and development in this domain.

3.1 Design Space Coverage

Fig. 7 compares the quality (MED)-area design space of an 8-bit QuAd adder model to that of the combined design space of state-of-the-art adders, including GeAr [10], ETA-II [9], ETA-IIM [9], GDA [8], ACA [6][7], ACAA, and ESA [16] for various L_{max} values. Fig. 7 demonstrates that the QuAd adder not only covers the configurations of state-of-the-art low-latency adders but also provides further configurations that utilize lesser area while providing better/same error measures.

Fig. 7 further supports our mathematical analysis of Section II that for each L_{max} there exists QuAd configurations (with $P_i < R_{i-1} + P_{i-1}$) that require lesser area while providing lower MED as compared to the sub-optimal configurations of QuAd ($with\ P_i \geq R_{i-1} + P_{i-1}$).



(a) L_{max}=2 (b) L_{max}=3

(c) L_{max}=4 (d) L_{max}=5

(e) L_{max}=6 (f) L_{max}=7

Figure 7. Design space of an 8-bit QuAd adder for various L_{max} values. The case of $L_{max} = 1$ is not provided since that requires $R = 1$ for all sub-adders.

3.2 Quality-Area-Latency evaluation

Fig. 7 also provides the design points that relate to our optimal adder configuration (QuAd$_o$), for each possible value of L_{max}. It can be observed that the QuAd$_o$ configuration provides us with the minimum area and minimum MED for each case. As an example, $QuAd_o(8,5)$ and $QuAd_o(8,6)$ provide 20% (Fig.7(d)) and 33% (Fig.7(e)) area reduction, respectively, as compared to the best possible configuration provided by the state-of-the-art. Fig. 8 and Fig. 9 compare the *Latency (ns) vs Quality (MSE and MED) design space* of the aforementioned state-of-the-art adders with that of QuAd adder for an 8-bit low latency approximate addition. It can be observed that the QuAd$_o$ configuration always provides an adder configuration that provides the lowest latency for any value of MSE or MED.

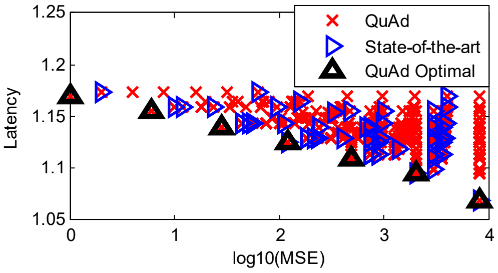


Figure 8. Latency (ns) vs log$_{10}$(MSE) for 8-bit QuAd configurations

$QuAd_o(N, L_{max})$ utilizes multiple non-overlapping sub-adders, each with length L_{max}, to provide a configuration that is optimal in terms of MSE, MED. Note that in Fig. 8 and Fig. 9 there are a few other configurations of QuAd adder that underlay the QuAd$_o$ configurations and provides the optimal MSE and MED for an approximate adder. However, it was shown using (9) that employing smaller sized non-overlapping sub-adders result in configurations that shall not be optimal in terms of Max_E.

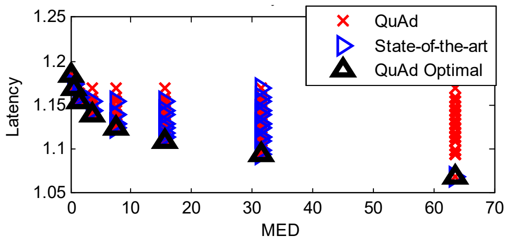


Figure 9. Latency (ns) vs MED for 8-bit QuAd configurations

To prove experimentally, we computed Max_E by performing exhaustive simulations for all the 8-bit configurations of QuAd that provided optimal MSE and MED values. These values are plotted in Fig.10 vs latency. It can be observed that for each possible latency value the L_{max}, QuAd$_o$ provides the minimum value of Max_E. Therefore, using proposed QuAd$_o$ we can select an approximate adder which is optimal in terms of MSE, MED, Max_E and also utilizing minimum possible area in case of FPGAs which is equivalent to that of an RCA.

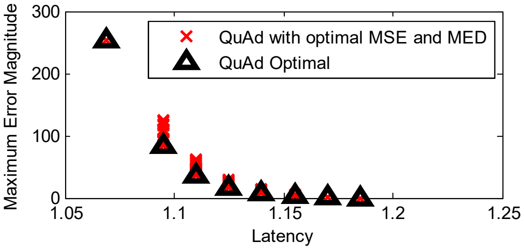


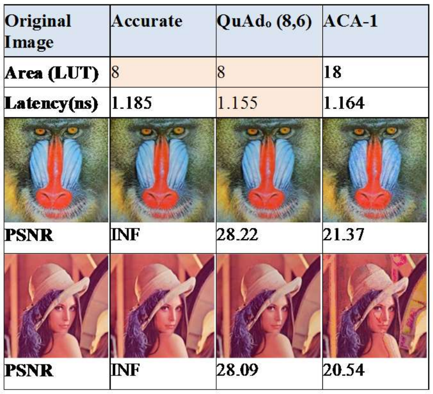
Figure 10. Max_E for all QuAd configurations with optimal MSE and MED plotted against various latency values (ns)

3.3 Performance Evaluation in a Real-World Application

To demonstrate the effectiveness of QuAd$_o$ adder in real-world applications, we present result of 4x4 Gaussian smoothing using *QuAd$_o$*(8,6) adder. The multiplication in Gaussian smoothing is implemented using precise components and the QuAd$_o$ adders are assumed to be connected in cascade. Fig. 11 demonstrates that the optimal *QuAd$_o$*(8,6) configuration provides an acceptable level of visual quality. It is noteworthy that QuAd$_o$ does not require any prediction (*P*) bit and still provides better PSNR as compared to ACA-I (*QuAd*{[6,1,1],[0, 5 5]}), that is a configuration with maximum overlap between the sub-adders for L_{max}=6. Since, QuAd$_o$ does not require any *P* bits, it has no associated area overhead as compared to an accurate adder and requires 8 LUTs for each *QuAd$_o$*(8,6) adder.

4 CONCLUSION

In this work, we proposed a highly configurable QuAd Adder model that offers a higher number of selectable configurations and provides better trade-off between performance and accuracy as compared to state-of-the-art low-latency adders for error resilient applications. Furthermore, utilizing the proposed method we can effortlessly select an optimal configuration, i.e. QuAd$_o$, for a specific latency constraint L_{max}. QuAd$_o$ is optimal in terms of MED, MSE, and Maximum error bound and unlike most of the low latency approximate adders, does not require overlapping sub-adders and hence reduces the latency without any area overhead. Experimental results demonstrate that, provided a latency constraint, QuAd$_o$ always provides an adder with the best quality-area metric.



Original Image	Accurate	QuAd$_o$ (8,6)	ACA-1
Area (LUT)	8	8	**18**
Latency(ns)	1.185	1.155	1.164
PSNR	**INF**	**28.22**	**21.37**
PSNR	**INF**	**28.09**	**20.54**

Figure 11. 4x4 Gaussian smoothing using QuAd adder with L_{max} = 6. Highlighted values corresponds to lowest area and latency.

REFERENCES

[1] A. K. Mishra, R. Barik, S. Paul, "iACT: A Software-Hardware Framework for Understanding the Scope of Approximate Computing", Workshop on Approximate Computing Across the System Stack (WACAS), 2014.

[2] R. Nair, "Big data needs approximate computing: technical perspective", ACM Communications, 58(1): 104, 2015.

[3] J. Bornholt, T. Mytkowicz, K. S. McKinley, "Uncertain<T>: Abstractions for Uncertain Hardware and Software", IEEE Micro 35(3): pp. 132-143, 2015.

[4] J. Bornholt, T. Mytkowicz, K. S. McKinley, "Uncertain: a first-order type for uncertain data", International conference on Architectural support for programming languages and operating systems (ASPLOS), pp. 51-66, 2014.

[5] S. Misailovic, M. Carbin, S. Achour, Z. Qi, M. C. Rinard, "Chisel: reliability and accuracy aware optimization of approximate computational kernels", International Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA), pp. 309-328, 2014.

[6] A. K. Verma, P. Brisk, P. Ienne, "Variable Latency Speculative Addition: A New Paradigm for Arithmetic Circuit Design". Design, Automation and Test in Europe (DATE), pp. 1250-1255, 2008.

[7] A. B. Kahng, S. Kang, "Accuracy-configurable adder for approximate arithmetic designs", Design Automation Conference (DAC), pp.820-825, 2012.

[8] R. Ye, T. Wang, F. Yuan, R. Kumar, Q. Xu, "On reconfigurationoriented approximate adder design and its application", International Conference on Computer-Aided Design (ICCAD), pp.48-54, 2013.

[9] N. Zhu, W.-L. Goh, K.-S. Yeo, "An enhanced low-power high-speed Adder for Error-Tolerant application", International Symposium on Integrated Circuits (ISIC), pp. 69-72, 2009.

[10] M. Shafique, W. Ahmad, R. Hafiz, J. Henkel, "A Low Latency Generic Accuracy Configurable Adder", IEEE/ACM Design Automation Conference (DAC), pp. 1-6, 2015.

[11] J. Miao, K. He, A. Gerstlauer, M. Orshansky, "Modeling and synthesis of quality-energy optimal approximate adders", International Conference on Computer Aided Design (ICCAD), pp. 728-735, 2012.

[12] M. Shafique, R. Hafiz, S. Rehman, W. El-Harouni, and J. Henkel, "Cross-Layer Approximate Computing: From Logic to Architectures", IEEE/ACM Design Automation Conference (DAC), pp. 1-6, 2016.

[13] J. Liang , J. Han , F. Lombardi, New Metrics for the Reliability of Approximate and Probabilistic Adders, IEEE Transactions on Computers, v.62 n.9, p.1760-1771, 2013

[14] R. C. Gonzalez, and R. E. Woods. "Digital image processing.", 3rded , pearson education , pp. 376, 2008.

[15] P. Zicari, and S. Perri. "A fast carry chain adder for Virtex-5 FPGAs." In MELECON 2010-2010 15th IEEE Mediterranean Electrotechnical Conference, pp. 304-308. IEEE, 2010.

[16] H. Jiang, J. Han, and F. Lombardi. "A comparative review and evaluation of approximate adders." In Proceedings of the 25th edition on Great Lakes Symposium on VLSI, pp. 343-348. ACM, 2015.