Journal of Circuits, Systems, and Computers

LPQ-SAM: A Low Power Quality Scalable Approximate Multiplier

Sumbal Iqbal

Department of Electrical Engineering, International Islamic University (IIU), and School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST)
Islamabad, Pakistan
sumbal.iqbal@iiu.edu.pk

Osman Hasan

School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST)
Islamabad, Pakistan
osman.hasan@seecs.edu.pk

Rehan Hafiz

Information Technology University (ITU)
Lahore, Pakistan
rehan.hafiz@itu.edu.pk

Zeshan Aslam Khan

Department of Electrical Engineering, International Islamic University (IIU)
Islamabad, Pakistan
zeeshan.aslam@iiu.edu.pk

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Approximate computing allows compromising accuracy to attain energy and performance efficient designs. However, the accuracy requirements of many applications change on runtime and it has been often observed that traditional approximate hardware tends to either provide unacceptable results or leads to an unnecessary computational effort. Quality scalable configurations can overcome these limitations . With the same motivation, we propose a low power quality scalable approximate multiplier (LPQ-SAM) in this paper. This low power multiplier has various accuracy reconfigurable modes, including an accurate one, and thus it can be used for both error-resilient and exact applications. LPQ-SAM is exhaustively tested for different error metrics and it has been observed that in the approximate mode, it provides up to 19% and 55% power reduction compared to the exact Booth and Wallace multipliers, respectively. For illustration purposes, we demonstrated the effectiveness of LPQ-SAM on a real-time application, i.e, image masking.

Keywords: Approximate Multiplier; Reconfigurable Computing; Image Processing; Qual-

ity scalable.

1. Introduction

Approximate computing has recently emerged as an energy-efficient choice for designing error-resilient digital systems[1]. An efficient system can be built by making suitable multiplier for that system. In accurate computation ,compromise on the bounds of accuracy (by bargaining accuracy and output quality) provides the new possibilities for improvement in area, power and performance. Recent research [2,3,4,5,6] has revealed that there are a number of error resilient applications of different domains that can compromise on accuracy and still give beneficial outputs. Such applications are: multimedia, data mining, communication and recognition etc. Approximate computing (in software and hardware), has been considered as a new approach to save area and power, as well as increasing performance, however at the cost of accuracy. There are many applications that do not require full accuracy in computation. For such applications, we can achieve reduction in area, power and delay, and hence achieve better energy efficiency by using approximate computing. Many applications, such as multimedia, data mining and recognition, are inherently error tolerant and thus can exploit the benefits of approximate computing[9,1,6,10]. However, in most of the real-world applications, the degree of error resilience varies significantly at runtime. Thus, traditional approximate hardware with fixed wired design or static approximation may not provide required quality.

Reconfigurable computing allows the system to adapt to the requirements by updating its functionality at runtime [11] and thus allows us to achieve maximum flexibility along with performance[12]. Thus, reconfigurable computing can be leveraged to cater for the above-mentioned problem in approximate computing. The foremost requirement in this regard would be to have quality-effort configurable basic approximate arithmetic blocks, such as adders, multipliers. However, most of the existing approximate adders and multipliers do not support this feature. Recently, a couple of reconfigurable adder circuits (e.g., [13,14])have been proposed and have shown quite promising results. However, to the best of our knowledge, no reconfigurable multiplier has been proposed in the literature so far. In this paper, we propose a low power quality scalable approximate multiplier (LPQ-SAM) that allows the reconfigurability of its accuracy. LPQ-SAM is designed on the principles of adaptable approximation. Some of the main contribution of LPQ-SAM are:

(i) A power efficient multiplier design, LPQ-SAM, that supports various approximated modes for dynamic accuracy configurability based on the modified booth multiplier, which decreases the partial products by 50%.
(ii) LPQ-SAM provides an average mean accuracy of 97% when we tested for different error metrics. Furthermore, its effectiveness was demonstrated for an image processing application.
(iii) Using different modes while providing a power reduction of 19.6% as compared to the Booth multiplier and 55% as compared to the exact Wallace tree multiplier.

The rest of the paper is organized as follows: Section 2 provides a detailed literature review of reconfigurable and approximate multipliers. Section 3 explains the architecture of LPQ-SAM. Section 4 discusses the experimental results for LPQ-SAM for the pixel-by-pixel multiplication example. Finally, Section 5 concludes the paper.

2. Related Work

Adders and multipliers are basic building blocks of many circuits and thus a lot of efforts has been done to make them efficient.Multipliers use adder trees for summing the partial products. Therefore, designing approximate forms of these blocks has received major research interest [4,7,8,24,34,30,32,33]. Approximate adders truncate the carry propagation chain for reducing the critical-path [4,7,24,34] or exclude carry computation and circuit fragments to save power [35,36,37]. To minimize the length of carry prediction, GDA [12] combines several sub-adder units in overlapping manner, thus gaining a significant area or power overhead. Therefore, these adders cannot be used for designing low-power approximate multipliers. Parallel multipliers are found to be the most performance efficient ones and they can be classified as tree and array multipliers. Wallace tree multiplier is a high speed multiplier but it has a high structural irregularity[17]. Baugh-Wooley[18,19] is an array multiplier that executes signed multiplication but it is not appropriate for large size operands. Booth radix-2 multiplier [20] is used for signed binary numbers with no correction phases for signed terms. But it does not provide an efficient solution for alternate zeros and ones leading to area and speed limitations. The modified booth multiplier (radix-4 multiplier) allows decreasing the number of partial products by fifty percent [15]. Consequently, it increases speed, decreases power consumption and also saves multiplier layout area. Its regular structure makes it quite suitable for reconfiguration as well. Parallel multipliers are found to be the most performance efficient ones and they can be classified as tree and array multipliers. Wallace tree multiplier is a high speed multiplier but it has a high structural irregularity[17]. Baugh-Wooley[18,19] is an array multiplier that executes signed multiplication but it is not appropriate for large size operands. Booth radix-2 multiplier [20] is used for signed binary numbers with no correction phases for signed terms. But it does not provide an efficient solution for alternate zeros and ones leading to area and speed limitations. The modified booth multiplier (radix-4 multiplier) allows decreasing the number of partial products by fifty percent [15]. Consequently, it increases speed, decreases power consumption and also saves multiplier layout area. Its regular structure makes it quite suitable for reconfiguration as well.

Most of the research in the approximate computing is focused on optimizing approximate adders [21,9,22,23,24,25] and there has been comparatively less work done in the design of approximate multipliers. Typically, multiplication comprises of three levels that can be approximated, i.e, generation and accumulation of partial products and carry propagation. Thus, approximate multipliers have been designed based on functional approximation, Voltage over scaling, shortening of the carry

chain with error configurability and truncation. Recently for Fog computation, QoS maximation of applications having approximate computation has been addressed. The focus was reusability of end devices[38]. The complexity of QoS-constrained network lifetime optimization in battery powered IoT for approximate computation real-time assignments has been tackled[39]

The Under designed Multiplier (UDM) [26] is an approximate 2x2 multiplier block in which only one entry of the K-Map is approximated. It can be used to design larger multipliers using 2x2 multipliers. In this multiplier, the error is introduced only in the generation of partial products and the accumulation is done accurately. Approximated Compressor based Multipliers [27] present four different designs of 8x8 multipliers. They primarily use the recursive multiplication technique to exploit the partial product reduction via approximated compressors. This design has been reported to exhibit better characteristics in terms of performance but at the cost of a larger area. Recently, Liu et al. [28] proposed an approximate multiplier consisting of fast approximate adders, which cut the carry propagation chain and thus consumes less power. However, this multiplier has a comparatively higher error rate. Bhardwaj et al. [16] presented a power and area efficient approximate Wallace tree multiplier based on a bit width aware algorithm, which consists of a novel carry-prediction technique. This multiplier uses four sub multipliers for performing a 16x16 multiplication and approximation is applied only on three sub multipliers. For operands of size greater than 10, the mean accuracy of this multiplier is reported to be 99.965% with a momentous decrease in power and area. The programmable truncated multiplier [29] undergoes the truncation of partial products. This design achieves a significant power reduction dynamically by disabling the partial products columns. Momeni et al. [27] proposed two further types of approximated multipliers. Type 0 provides more accuracy but with less power savings. While Type 1 has more power savings with less accuracy compared to Type 0. Power consumption was reported to be reduced by 58% by compromising the accuracy. The Error-Tolerant Multiplier (ETM) [30] is a low power and high speed multiplier. It is divided into a multiplication part for the MSBs and a non-multiplication part for the LSBs and a control block is used to control the approximate or accurate computation.

A recent trend in approximate arithmetic block design is to move towards flexible designs with controllable accuracy-performance features. Run time adaptation can be very useful in this domain and with the same motivation a couple of accuracy scalable approximate adders have been recently proposed[13,14]. Despite the promising results shown by these reconfigurable adder designs, to the best of our knowledge, these adders do not provide run-time accuracy adaptation/configurability. In this paper, we present a low power quality scalable approximate multiplier (LPQ-SAM). The architecture of LPQ-SAM is described in the next section.

3. LPQ-SAM Design

The proposed LPQ-SAM has the capability of accuracy configurability. Fig. 1 depicts the design methodology of LPQ-SAM and its main components are described below:

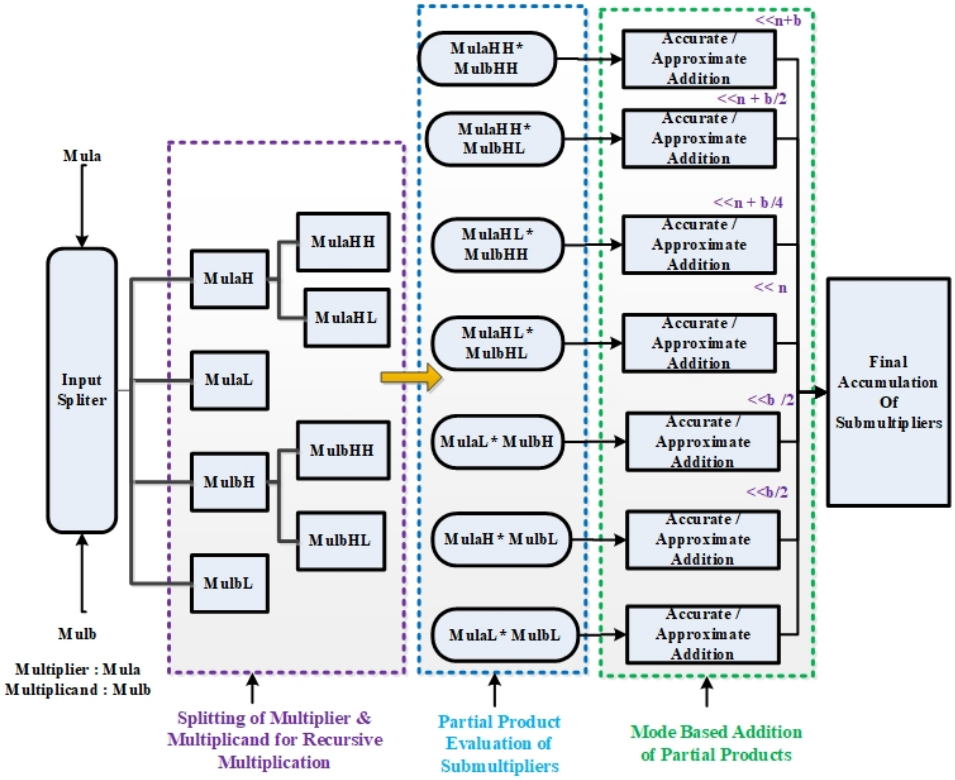


Fig. 1: Proposed Approach

3.1. *Recursive Multiplication*

In this step, the two multiplier operands (multiplier and multiplicand) are subdivided into smaller operands. The main motivation of this step is to obtain the same multiplication result but with smaller multipliers. Consider the example of a multiplier with 16-bit operands. We divide both multiplier and multiplicand into three parts, i.e., the least-significant 8 bits (Mul_{LSB}) and the most-significant 8 bits are further divided into most significant 4-bits ($Mul_{MSBhigh}$) and least significant 4-bits (Mul_{MSBlow}) as shown in Fig. 2.

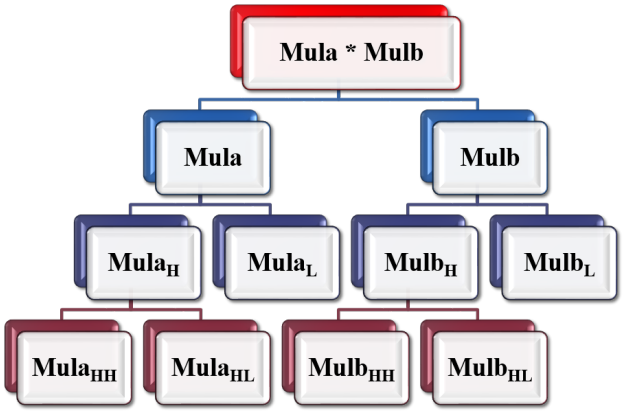


Fig. 2: Recursive Multiplication

3.2. *Mode Selection*

Thus, MulA = [$Mul_{MSBhigh}$, Mul_{MSBlow}, Mul_{LSB}]. Similarly the multiplicand is also divided. So instead of a 16*16 multiplication, we can achieve the same result via three 8*8 and four 4*4 multiplications. The main benefit of this kind of recursive multiplication method is that the smaller multiplications can be executed in parallel and then can be accumulated in the final stage to provide the desired result.

One of the distinguishing features of the proposed LPQ-SAM is the availability of having different accuracy levels, which can be selected on runtime based on the encountered application. After the multiplication step, the mode selection allows to select the level of accuracy of the multiplication result. The modes supported by LPQ-SAM are shown in Table 1. The first mode corresponds to the exact answer, which is computed recursively. This mode is for applications that are not error tolerant. It is important to note that the recursive multiplication would lead to a faster computation compared to the traditional multiplier. In other modes, i.e., 2-7, some of the sub multipliers are chosen to provide approximate results. These modes provide different levels of accuracy depending on the amount of approximation introduced and thus allow us to obtain accuracy configurability. Mode 2 exhibits the least approximation as only one sub multiplier, i.e., $Mula_L Mulb_L$ is approximated. Mode 7 has the maximum approximation as seven sub multipliers ($Mula_L Mulb_L$,$Mula_L Mulb_H$, $Mula_H Mulb_L$, $Mula_{HL} Mulb_{HL}$, $Mula_{HL} Mulb_{HH}$ and $Mula_{HH} Mulb_{HL}$) are approximated. The other modes are between these two extremes as shown in Table 1.

3.3. *Accumulation of Partial Products*

After the mode selection, the partial products of sub-multipliers are accumulated. The accumulation of partial products can also be approximated or can be done in

an exact manner. Table 1 provides a brief overview of these computations based on the selected mode.

Table 1: Approximate Modes

Modes	$Mula_{HH}$*$Mulb_{HH}$	$Mula_{HH}$*$Mulb_{HL}$	$Mula_{HL}$*$Mulb_{HH}$	$Mula_{HL}$*$Mulb_{HL}$	$Mula_{H}$*$Mulb_{L}$	$Mula_{L}$*$Mulb_{H}$	$Mula_{L}$*$Mulb_{L}$
0	Accurate	Accurate	Accurate	Accurate	Accurate	Accurate	Accurate
1	Accurate	Accurate	Accurate	Accurate	Accurate	Accurate	Approx.
2	Accurate	Accurate	Accurate	Accurate	Accurate	Approx.	Approx.
3	Accurate	Accurate	Accurate	Accurate	Approx.	Approx.	Approx.
4	Accurate	Accurate	Accurate	Approx.	Approx.	Approx.	Approx.
5	Accurate	Accurate	Approx.	Approx.	Approx.	Approx.	Approx.
6	Accurate	Approx.	Approx.	Approx.	Approx.	Approx.	Approx.
7	Approx.	Approx.	Approx.	Approx.	Approx.	Approx.	Approx.

If an approximated mode is selected then the addition of the considered partial product (evaluated using modified booth multiplier) are approximated. We use the method proposed in [16] for the middle bits. For example, consider a multiplier of 2n x2n (Both operands of the multiplier are of size 2n) then the addition of partial products will be done in such a way that n bits (LSBs) of the final result will be set to zero directly. While the next n bits of result will be set to 1^{16}.

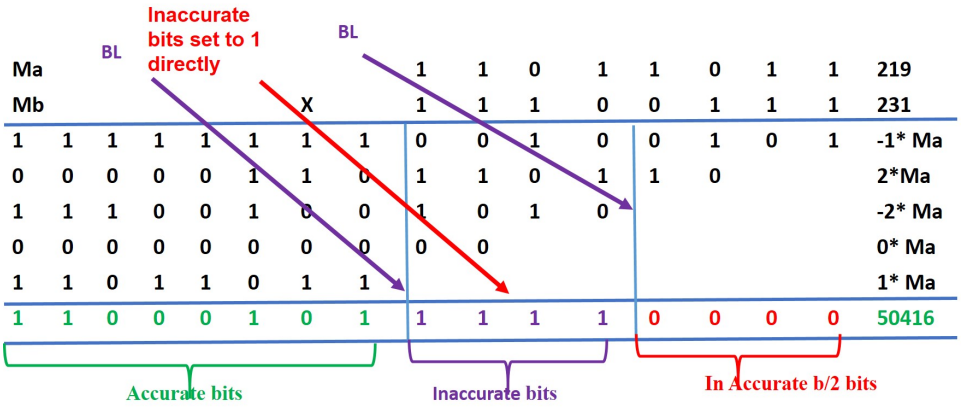


Fig. 3: Approximate Computation.

Fig. 3 provides an illustrative example of approximate computation. In this example, we obtain the partial products using the modified booth multiplier Radix-4. If an approximation mode is selected then we apply two vertical break levels where the lower $n/2$ bits are truncated, i.e, directly set to zero. Whereas, the next break level $n/2$ bits are directly set to 1^{16} and the higher n bits are obtained using an accurate adder.

The addition of partial products is done using an accurate adder when the accurate mode is selected. In this mode, the break levels are disabled through which the approximation was done as shown in Fig. 4.

Ma								**1**	**1**	**0**	**1**	**1**	**0**	**1**	**1**	**219**
Mb						**X**		**1**	**1**	**1**	**0**	**0**	**1**	**1**	**1**	**231**
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**-1* Ma**
0	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**1**	**1**	**0**			**2*Ma**
1	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**0**					**-2* Ma**
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**							**0* Ma**
1	**1**	**0**	**1**	**1**	**0**	**1**	**1**									**1* Ma**
1	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**1**	**0**	**1**	**50589**

Fig. 4: Accurate Computation.

3.4. *Final Accumulation of Sub Multipliers Results*

As described above, if the approximate mode is selected then the two break levels are applied on the partial products and the addition is done on partial products accordingly. Finally, we have the result of each sub multiplier. The last step is to add all these results to obtain the final multiplication result as illustrated in Fig. 5.

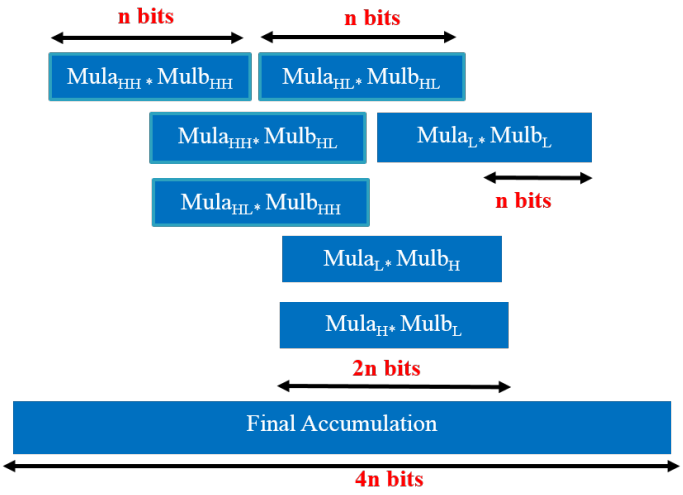


Fig. 5: Final Accumulation.

4. Results and Discussions

We analyzed the performance of LPQ-SAM for several commonly used error metrics for approximate circuits. For the power consumption comparison, we have implemented the 16 x 16 multiplier in Verilog HDL and synthesized the design using the XPower Analyzer in Xilinx 14.3. We have demonstrated the efficiency enhancement of our design by comparing it with both accurate and approximate multipliers. Moreover, we also used LPQ-SAM for a pixel-by-pixel image multiplication to illustrate its usefulness for real-world problems.

4.1. *Performance Evaluation*

4.1.1. *Error Metrics*

We simulated LPQ-SAM for one hundred thousand inputs in Matlab to evaluate its accuracy in terms of the following metrics [31].

- **Maximum Error** is the largest value of error incurred among all of the tested combinations.
- **Minimum Error** is the smallest value of error calculated among all of the tested combinations.
- **Error Distance (ED)**

$$ErrorDistance = M_{approx} - M \tag{1}$$

  where M_{approx} and M represent the approximated multiplication result and accurate multiplication result, respectively.
- **Mean Error** is calculated by accumulating all of the error distances and then dividing the result by the total number of error distance observations.
- **Relative Error Distance (RED)** is calculated by dividing the ED with the accurate multiplication result:

$$\frac{Error\ Distance}{Accurate\ Multiplication\ Result} \tag{2}$$

- **Accuracy** is calculated by subtracting RED from 1.

$$Accuracy = 1 - RED \tag{3}$$

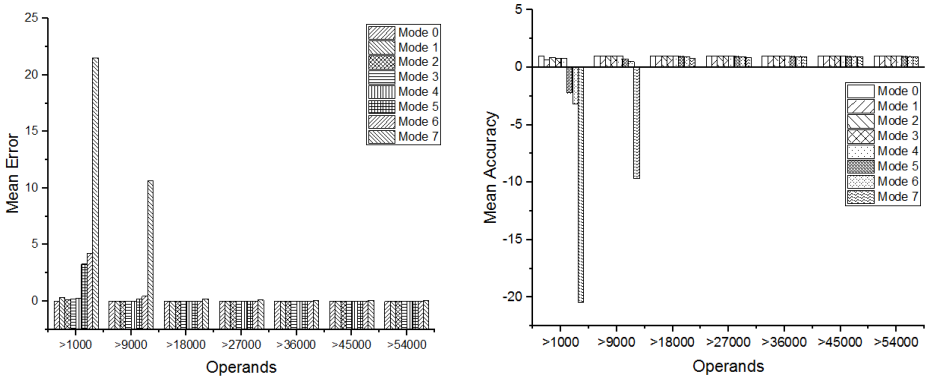
- **Average Accuracy** is the average of the accuracies calculated for all the test combinations.
- **Maximum Accuracy** is the largest accuracy value calculated while considering all the tested combinations.
- **Minimum Accuracy** is the smallest accuracy value calculated while considering the tested combinations.

The Table2 summarizes our simulation results of LPQ-SAM while considering all the available accuracy modes with operand values greater than 100. It can be observed that the accuracy reduces from Mode 0 to 7.

Table 2: Modes Results

Mode	Max Error	Min Error	Average Error Distance	Mean Relative Error	Max. Accuracy	Average Accuracy
0	0	0	0	0	1	1
1	637	-65499	955.7909	0.10813	1	0.8919
2	168494	-16813426	341966.1441	2.03265	0.9999991	0.85214
3	272710	-33606369	1400501	21.477	0.99999	0.83326
4	1232025	-49918273	3133297.089	33.9537	0.999999	0.784
5	16854302	-294165818	18984369.73	75.1	0.999944	0.625
6	30441830	-542249476	31824526	19.9354	0.99999067	0.4332
7	4114654617	-4228677408	359309611.1	1480.278	0.999968137	-3.499

Figs. 6a and 6b shows the mean error and accuracy comparison of the eight modes of LPQ-SAM for the operand range >1000. It has been observed that both

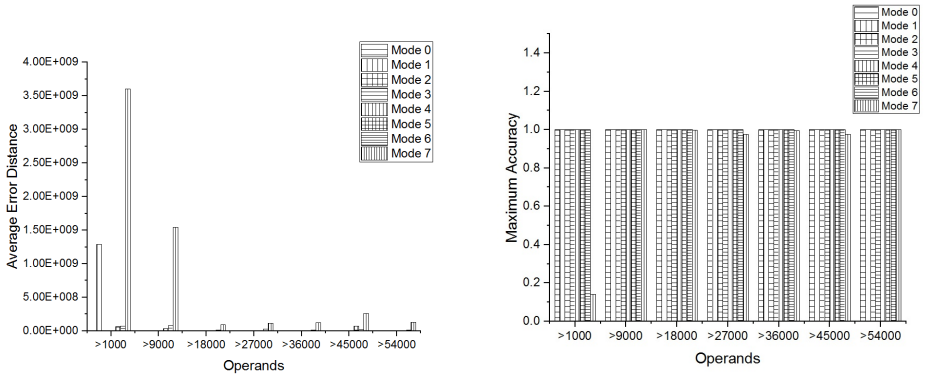


(a) Mean Error (b) Mean Accuracy

Fig. 6: Mean Error and Accuracy

mean error and accuracy remain the same as operand range increases for all modes. This depicts that LPQ-SAM barely compromises the accuracy of the results for all the approximate modes when the operand size >9000. Similarly, Figs. 7a and 7b shows the average error distance and maximum accuracy comparison of the eight modes of LPQ-SAM.

Table 3 shows that the accuracy of Mode 1 is 67.7% to 99.9%. It is important to note that Mode 1 is the least approximated mode in which only the last eight bits of the 16x16 multiplication result is approximated. In this mode out of seven sub multiplier, only one multiplier i.e. $Mula_L$* $Mulb_L$ is approximated.



(a) Average Error Distance (b) Maximum Accuracy

Fig. 7: Average Error Distance and Maximum Accuracy

Table 3: Mode 1 Accuracy Evaluation

Accuracy Metrics \ Operands Range	>1000	>9000	>18000	>27000	>36000	>45000	>54000
Max Error	5997448303	654	570	613	633	629	614
Min Error	-65464	-65442	-65480	-65496	-65480	-65470	-65496
Average Error Distance	1288209441	1009.04	1206.688462	603.2523	1154.584615	1659.106	1005.77
Mean Error	0.322310337	1.98E-06	2.55E-06	5.95E-07	6.82E-07	6.97E-07	2.83E-07
Average Accuracy	0.67768	0.999998019	0.999997449	0.9999994	0.99999931	0.999999303	0.99999971
Max Accuracy	1	1	1	1	1	1	1

The partial product of each multiplier is obtained using Modified booth multiplier. But accumulation depends on mode selected. In this mode only accumulation of $Mula_L$* $Mulb_L$ is approximated. Maximum accuracy achieved for this mode is 1. For operands >1000, the mean error is 0.322 .As the operand size increases the mean error is reduced. For operands >9000 the mean error is 1.9E-6 which is a significant drop.For operands >18000 there is small reduction in mean error. For operands >1000, the average accuracy is 67.6% .As the operand size increases the average accuracy is increased for operands >9000 the average accuracy is 99.999% which is a significant increase . For operands >18000 there is small increase in average accuracy i.e 99.9999%. Mode 2 is the second least approximated mode and its average accuracy ranges from 87.6 % to 99.9% as show in Table 4. In this mode two sub multipliers i.e $Mula_L$* $Mulb_L$, $Mula_H$ * $Mulb_L$are approximated.

The partial products of these sub multipliers will be evaluated using modified booth multiplier .While accumulation of these two sub multipliers will be approximated. Accumulation of other five multipliers will be accurate.The mean error for

Table 4: Mode 2 Accuracy Evaluation

Accuracy Metrics \ Operands Range	>1000	>9000	>18000	>27000	>36000	>45000	>54000
Max Error	157492	163633	144986	156563	152618	151083	168494
Min Error	-16813426	-16807077	-89468	-126810	-95034	-95889	-114338
Average Error Distance	979515.33	143779.45	37203.35231	40560.53846	40093.84	39734.77077	38113.01224
Mean Error	0.1231	0.000885907	7.56898E-05	4.11095E-05	2.46053E-05	1.63168E-05	1.07446E-05
Average Accuracy	0.876899	0.999114	0.99992431	0.99995889	0.999975395	0.999983683	0.999989255
Max Accuracy	0.99999	0.99999747	0.99999981	0.999839629	0.999999989	0.999999991	0.99999999

this mode ranges from 0.123 to 1.07446 * 10^{-5}. For operands >1000, the mean error is 0.1231.As the operand size increases the mean error is reduced for operands >9000 the mean error is 0.000885 which is a noteworthy drop. For operands >18000 mean error is 7.56E-05 than for operands >27000, there is a great reduction in mean error. For operands >1000, the average accuracy is 87.6% .As the operand size increases the average accuracy is increased for >9000 the average accuracy is 99.914% which is a significant increase. For operand >18000 there is small increase in average accuracy i.e. 99.999%.

Mode 3 represents the case when three out of seven sub multipliers are approximated. Mode 3 is the approximated mode in which approximation is increased as compared to Mode 2. In this mode three sub multipliers out of seven are approximated. Partial Products of all these sub multipliers are computed using Modified booth recording. But the accumulation of three sub multipliers will be approximated i.e. $Mula_L$* $Mulb_L$, $Mula_H$ * $Mulb_L$ and $Mula_L$* $Mulb_H$. This mode is also tested for random generated numbers. Average accuracy for this mode is 81.9% to 99.9% as show in Table 5.Mean error is 0.2009 to 2.46* 10^{-5}. For operands >1000, the mean error is 0.20.As the operand size increases the mean error is reduced for operands >9000 the mean error is 0.003 which is a significant drop. For operands >18000 mean error is 0.0004 than for operand >27000, there is a significant reduction in mean error. For operands >1000, the average accuracy is 81.9% .As the operand size increases the average accuracy is increased for operands >9000 the average accuracy is 99.6% which is a major increase. For operands >18000 there is small increase in average accuracy i.e. 99.95%.As this mode has more approximation than mode 2 thats why its mean error is more and accuracy is less as compared to mode2.

Tables 6 – 9 show the accuracy evaluation results of approximated modes such as Mode 4 to 7 in detail.

4.1.2. *Power Consumption Analysis*

We present the power analysis results of LPQ-SAM and their comparison with the power dissipation for some commonly used ideal, approximate and reconfigurable multipliers in Fig. 8. The comparison shows that LPQ-SAM exhibits a lower power consumption compared to Wallace, Booth and Modified Booth multipliers. More-

Table 5: Mode 3 Accuracy Evaluation.

Accuracy Metrics \ Operands Range	>1000	>9000	>18000	>27000	>36000	>45000	>54000
Max Error	242698	268277	267903	256098	240846	224871	272710
Min Error	-33529720	-16793152	-16820131	-16827370	-16785980	-16744310	-16773921
Average Error Distance	1954858.855	594574.79	234637.3554	122190.2631	137179.8577	87883.48615	83549.41385
Mean Error	0.200914724	0.003810451	0.000438972	0.000126862	8.17E-05	3.66E-05	2.46E-05
Average Accuracy	0.8192857	0.996189549	0.999561028	0.999873138	0.99991832	0.999963425	0.999975371
Max Accuracy	0.99995795	0.99999336	0.99999685	0.99999973	0.999999985	0.999999969	0.99999995

Table 6: Mode 4 Accuracy Evaluation

Accuracy Metrics \ Operands Range	>1000	>9000	>18000	>27000	>36000	>45000	>54000
Max Error	1081290	1094048	1164955	1039883	1232025	1118138	1208886
Min Error	-33695804	-32915589	-17564360	-17613098	-32412752	-32651017	-17122626
Average Error Distance	2656442.656	1521997.775	856515.3323	1161568.32	948797.9646	1204227.028	973955.0162
Mean Error	0.250322486	0.009166087	0.001704178	0.001192264	5.90E-04	5.01E-04	2.74E-04
Average Accuracy	0.749677514	0.990833913	0.998295822	0.998807736	0.999410191	0.999498574	0.999725623
Max Accuracy	0.999957792	0.999999695	0.999998984	0.999999169	0.999999881	0.999999995	0.999999831

Table 7: Mode 5 Accuracy Evaluation.

Accuracy Metrics \ Operands Range	>1000	>9000	>18000	>27000	>36000	>45000	>54000
Max Error	11246718	11242936	12357833	10114536	15700632	13874536	16499058
Min Error	-294165818	-276070959	-34939600	-29298701	-30315600	-30110472	-29656053
Average Error Distance	61187504.07	35735692.78	6001142.838	7156469.872	6224778.495	71315196.44	6951526.356
Mean Error	3.271484173	0.222956276	0.012280198	0.007188905	3.85E-03	2.92E-03	1.94E-03
Average Accuracy	-2.2714	0.777043724	0.987719802	0.99811095	0.996152032	0.997084406	0.998057933
Max Accuracy	0.999919089	0.999994026	0.99999944	0.999988549	0.999997585	0.999998465	0.999999234

over, it can also be seen that LPQ-SAM has a lower power consumption compared to all the other reconfigurable and approximate multipliers [21,9,17,26]. It is important to note that LPQ-SAM power dissipation includes the dissipation for Mode 0, which provides the exact answers.

4.1.3. *Real-time Image Multiplication Application (Pixel-by-Pixel)*

To demonstrate the effectiveness of our multiplier we have used it for the multiplication of two 1D binary images (159x158). The multiplication of these images,i.e., Image 1 and Image 2, using an accurate multiplier is shown in Fig. 9. We used

Table 8: Mode 6 Accuracy Evaluation.

Accuracy Metrics \ Operands Range	>1000	>9000	>18000	>27000	>36000	>45000	>54000
Max Error	5731572	20089272	26912972	25979578	30441830	24037136	25932461
Min Error	-540564780	-542249476	-282390632	-279985600	-276058672	-277078996	-272949396
Average Error Distance	73462693.13	79054457.03	14772608.31	27172801.91	15373919.62	19926516.52	12497899.64
Mean Error	4.233833	0.458817394	0.028451296	0.028693465	9.37E-03	8.36E-03	4.38E-03
Average Accuracy	-3.233833	0.541182606	0.971548704	0.971306535	0.990629943	0.991639532	0.995615611
Max Accuracy	0.999295468	0.999968137	0.99999747	0.99972724	0.99991359	0.999981664	0.99998663

Table 9: Mode 7 Accuracy Evaluation

Accuracy Metrics \ Operands Range	>1000	>9000	>18000	>27000	>36000	>45000	>54000
Max Error	55883302	228546920	51758437	100926824	265253436	4114654617	216482270
Min Error	-4228677408	-4186615631	-416559120	-471809719	-226949486	-210599584	-411567231
Average Error Distance	3599104010	1546840879	96528548.17	115726276	125085475.4	262745104.2	126186526.1
Mean Error	21.51406819	10.6807323	0.209468387	0.11596611	7.58E-02	6.96E-02	5.12E-02
Average Accuracy	-20.5140682	-9.680732303	0.790531613	0.88403389	0.92417899	0.930353601	0.948790557
Max Accuracy	0.141780746	0.999968137	0.996877994	0.97489029	0.994392163	0.975127828	0.999967252

various modes of LPQ-SAM for the same exercise and the results are given in Fig. 10. It is interesting to see that the results for Modes 0, 1, 2 and 3 are very similar to the ones for the accurate multiplier. The results for Modes 4 to 6 are not up to mark as their accuracy is less than 75%. The results are completely unacceptable in the case of this example for Mode 7. This research work enables the idea of designing a runtime accurate reconfigurable approximate multiplier. The proposed design multiplier combines the benefits of approximate and reconfigurable computing. There are many applications which do not need full exact computation. For such applications, this design is proposed for different accuracy levels. Each level can be selected using a mode. There are total eight modes out of which seven modes have different accuracies. Mode 0 has 100% accuracy. By sacrificing accuracy, power efficiency is achieved. Modified Booth multiplier is used for multiplication as it produced less number of partial products. Approximation is applied while adding these partial products. Approximation and accuracy levels are selected through mode selection. An accurate mode having 100% accuracy is available in the proposed design for such applications which do not tolerate loss in accuracy. The proposed multiplier is compared with two exact multipliers i.e. Wallace and modified booth. We also compared it with approximate and reconfigurable multipliers stated in literature a shown in 8.

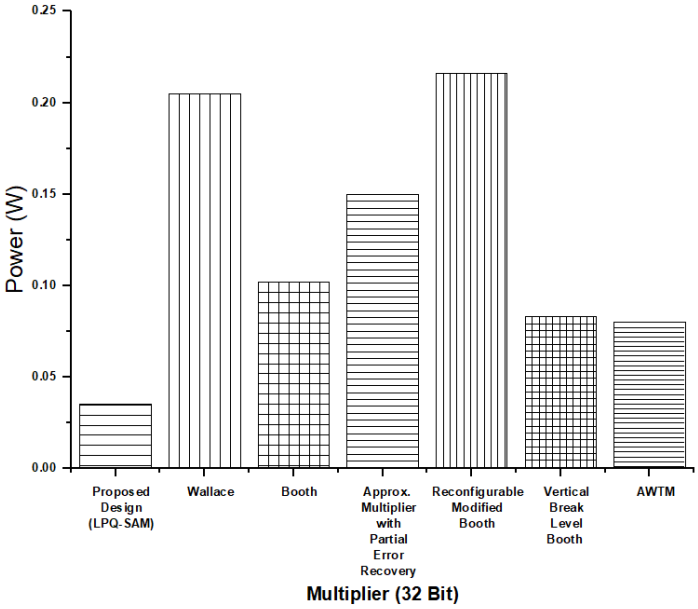


Fig. 8: Multipliers Power Consumption Comparison

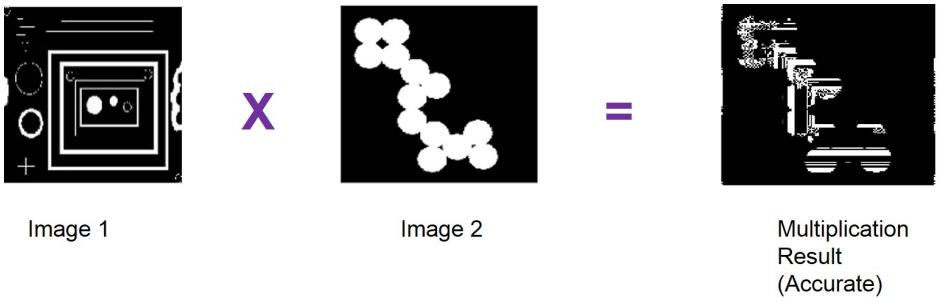


Fig. 9: Pixel By Pixel Accurate Multiplication

5. Conclusion

This paper presents a low power quality scalable approximate multiplier LPQ-SAM. The main distinguishing feature of LPQ-SAM is the ability to leverage upon the benefits of both approximate and reconfigurable computing. There are many applications which do not need full exact computations and for these applications, LPQ-SAM provides the flexibility to be configured in 8 different models with different levels of accuracy. The modes can be updated on runtime and thus allow

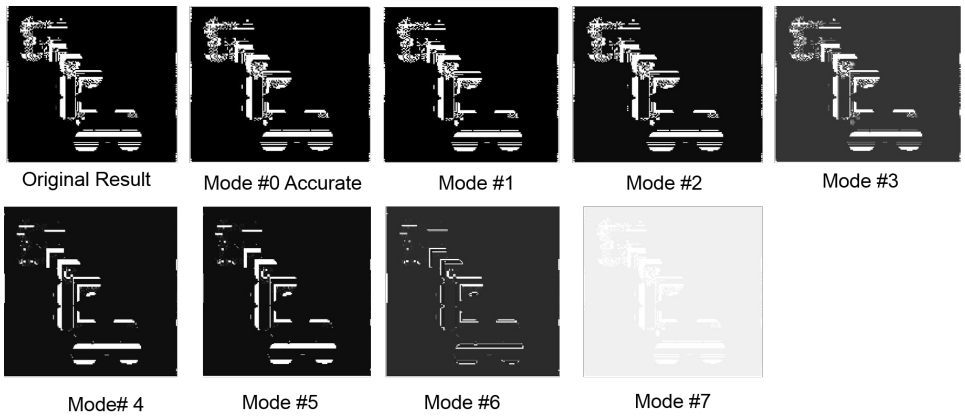


Fig. 10: Pixel by Pixel Multiplication using LPQ-SAM

LPQ-SAM to be reconfigured at runtime. The design of LPQ-SAM is based on the modified Booth multiplier, which leads to a reduced number of partial products and thus power savings. In the future, we plan to increase the levels of approximation. The ability to automatically reconfigure LPQ-SAM based on the applications is also an interesting future direction that we plan to explore.

References

1. Q. Zhang, F. Yuan, R.Ye, Q. Xu, ApproxIt: An Approximate Computing Framework for Iterative Methods, Proc. DAC (2014), pp. 97
2. Chippa, Vinay K. and Chakradhar, Srimat T. and Roy, Kaushik and Raghunathan, Anand, Analysis and Characterization of Inherent Application Resilience for Approximate Computing, Proceedings of the 50th Annual Design Automation Conference (DAC) 2013, pp. 1-113.
3. M. Shafique, W. Ahmad, R. Hafiz, T. Henkel, A Low Latency Generic Accuracy Configurable Adder, Proceedings of the 52Nd Annual Design Automation Conference (DAC) 2015, pp. 1-86.
4. 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.
5. V. Kumar, A. Singh, S. Upadhyay, B. Kumar, Power-Delay-Error Efficient Approximate Adder for Error-Resilient Applications, Journal of Circuits, Systems, and Computers (JCSC), 2018.
6. R. Nair, Big Data Needs Approximate Computing: Technical Perspective, Commun. ACM **58**,(2014), pp.104.
7. J. Miao, K. He, A. Gerstlauer, and M. Orshansky. Modeling and synthesis of quality-energy optimal approximate adders,In 2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)2012, pp.728-735.
8. M. B. Sullivan and E. E. Swartzlander. Truncated error correction for flexible approximate multiplication, In 2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers(ASILOMAR),2012, pp. 355-359.

9. L.N.B.Chakrapani, K. Muntimadugu, A. Lingamneni, J. George, K. Palem., Highly Energy and Performance Efficient Embedded Computing Through Approximately Correct Arithmetic: A Mathematical Foundation and Preliminary Experimental Validation, Proc. CASES '08, pp. 187-196.
10. M. Monajati, S.M. Fakhraie, E. Kabir, Approximate Arithmetic for Low-Power Image Median Filtering, Circuits, Systems, and Signal Processing Vol.**34**, 2015, 3191.
11. K. Bondalapati, V.K. Prasanna, Reconfigurable computing: Architectures, models and algorithms, Current Science **78**, 2000, pp. 828.
12. R. Ye, T. Wang, F. Yuan, R. Kumar, Q. Xu,On reconfigurationoriented approximate adder design and its application, in *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* (2013), pp. 48-54.
13. O. Akbari, M. Kamal, A. Afzali-Kusha, M. Pedram, RAP-CLA: A Reconfigurable Approximate Carry Look-Ahead Adder, IEEE Transactions on Circuits and Systems II: Express Briefs **pp**(99),2017.
14. R. Ye, T. Wang, F. Yuan, R. Kumar, Q. Xu, On recon
gurationoriented approximate adder design and its application in *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2013, pp. 48-54.
15. P.E. Madrid, B. Millar, E.E. Swartzlander, Modified Booth algorithm for high radix fixed-point multiplication, IEEE Transactions on Very Large Scale Integration (VLSI) Systems **1**, 1993, pp. 164.
16. K. Bhardwaj, P.S. Mane, J. Henkel, Power- and area-efficient Approximate Wallace Tree Multiplier for error-resilient systems in *15th International Symposium on Quality Electronic Design*, 2014, pp. 263-269.
17. S.S. Sakthi, N. Kayalvizhi, Power aware and high speed reconfigurable modified booth multiplier in *Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE*, 2011, pp. 352–356.
18. C.R. Baugh, B.A. Wooley, A Two's Complement Parallel Array Multiplication Algorithm, IEEE Transactions on Computers Vol. **C-22**(12), 1045(1973).
19. A. Sudhakar, D. Gokila, in *Proceedings of the 12th International Conference on Networking, VLSI and Signal Processing* (Stevens Point, Wisconsin, USA, 2010), pp. 79–82.
20. A.D. Booth, A signed Binary Multiplication Technique, Quart. J. Mech. Appl. Math.v, 1951),pp. 236-240
21. E.E. Swartzlander, Truncated multiplication with approximate rounding in *Signals, Systems, and Computers, 1999. Conference Record of the Thirty-Third Asilomar Conference on*, **2**, 1999, pp. 1480-1483
22. N. Zhu, W.L. Goh, W. Zhang, K.S. Yeo, Z.H. Kong, Design of Low-Power High-Speed Truncation-Error-Tolerant Adder and Its Application in Digital Signal Processing, IEEE Transactions on Very Large Scale Integration (VLSI) Systems Vol.**18**(8), 2010, pp. 1225 .
23. N. Zhu, W.L. Goh, K.S. Yeo, An enhanced low-power high-speed Adder For Error-Tolerant application in *Proceedings of the 2009 12th International Symposium on Integrated Circuits*, 2009, pp. 69-72
24. A.K. Verma, P. Brisk, P. Ienne, Variable Latency Speculative Addition: A New Paradigm for Arithmetic Circuit Design in *2008 Design, Automation and Test in Europe*, 2008, pp. 1250-1255.
25. A.B. Kahng, S. Kang, Accuracy-configurable adder for approximate arithmetic designs in *Design Automation Conference (DAC)*, 201), pp. 820-825.
26. P. Kulkarni, P. Gupta, M. Ercegovac, Trading Accuracy for Power with an Underdesigned Multiplier Architecture in *2011 24th Internatioal Conference on VLSI Design*

(2011), pp. 346-351.
27. A. Momeni, J. Han, P. Montuschi, F. Lombardi, Design and Analysis of Approximate Compressors for Multiplication IEEE Transactions on Computers **64**(4), (2015), pp. 984.
28. C. Liu, J. Han, F. Lombardi, A low-power, high-performance approximate multiplier with configurable partial error recovery in *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2014, pp. 1–4.
29. M. de la Guia Solaz, W. Han, R. Conway, A Flexible Low Power DSP With a Programmable Truncated Multiplier IEEE Transactions on Circuits and Systems I: Regular Papers **59**(11), 2012, pp. 2555 .
30. K.Y. Kyaw, W. Goh, K.S. Yeo, Low-power high-speed multiplier for error-tolerant application in *Electron Devices and Solid-State Circuits (EDSSC), 2010 IEEE International Conference of* (2010), pp. 1-4.
31. J. Liang, J.Han, F. Lombardi, New Metrics for the Reliability of Approximate and Probabilistic Adders IEEE Transactions on Computers **62**, 2013, pp. 1760.
32. F. Farshchi, M. S. Abrishami, S. M. Fakhraie, New approximate multiplier for low power digital signal processing, The 17th CSI International Symposium on Computer Architecture Digital Systems (CADS), 2013.
33. S. Sri Sakthi , N. Kayalvizhi, Power Aware Reconfigurable Multiplier for DSP Applications, International Journal of Computer Science and Engineering Technology (IJCSET), 2011.
34. M. V. Praveen Kumar and S. Sivanantham and S. Balamurugan and P. S. Mallick, International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011. pp. 532-536.
35. V. Gupta and D. Mohapatra and S. P. Park and A. Raghunathan and K. Roy, IMPACT: IMPrecise adders for low-power approximate computing, International Symposium on Low Power Electronics and Design (ISLPED), 2011, pp 409-414.
36. V. Gupta and D. Mohapatra and A. Raghunathan and K. Roy, Low-Power Digital Signal Processing Using Approximate Adders IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), 2013 **32**, pp. 124-137.
37. J. Miao and K. He and A. Gerstlauer and M. Orshansky, Modeling and synthesis of quality-energy optimal approximate adders, IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2012, pp. 728-735.
38. K. Cao, J. Zhou, G. Xu, T. Wei and S. Hu, "Exploring Renewable-Adaptive Computation Offloading for Hierarchical QoS Optimization in Fog Computing," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Dec. 2019
39. K. Cao, G. Xu, J. Zhou, T. Wei, M. Chen and S. Hu, "QoS-Adaptive Approximate Real-Time Computation for Mobility-Aware IoT Lifetime Optimization," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 38, no. 10, pp. 1799-1810, Oct. 2019