Efficient Approximate Adders with Minimal Error for FPGA Implementation

Kommana Raghavendra Rao[1], Adapaka Shanmukha Poorna Chand[2], Kallepalli Keerthi[3],
Budida Avinash[4], Mohammed Ahmed Ashraf[5]
[1]Assistant Professor, [2, 3, 4, 5,] Undergraduate. Department Of ECE,
PSCMR College Of Engineering and Technology, Vijayawada, Andhra Pradesh, India
raghukommana@gmail.com[1], shanmukhapoornachand14316@gmail.com[2],
keerthikallepalli6@gmail.com[3], avinashbudida59@gmail.com[4], ahmedashraf130701@gmail.com[5]

Abstract - **In this project, A method has been suggested for creating approximate adders that are both efficient and have low errors. The suggested approach maximizes FPGA resources to minimize error in approximate adders. We suggest two approximate adders for FPGAs with our approach: low error and space-saving approximate adder (LEADx), and efficient in both area and power approximate adder (APEX). Both approximate adders consist of a precise section and an approximate section. The components of these adders are structured systematically to reduce the mean square error (MSE) as much as possible. LEADx exhibits a smaller MSE compared to the approximate adders discussed in prior literature. APEX adders have a smaller area and consume less power compared to other approximate adders and traditional adders. In a video encoding application, approximate adders are utilized as a case study. LEADx offered superior quality compared to the other similar adders for video encoding application. Hence, our suggested approximate adders are suitable for effective FPGA realizations of error-tolerant tasks. The proposed method's effectiveness is demonstrated through synthesis and simulation using Xilinx ISE 14.7.**

Keywords - ***Approximate computing, approximate adder, FPGA, low error, low power, LUT***

I. INTRODUCTION

Approximate computing is a new design technique that trades off accuracy for performance, area and/or power consumption for error-tolerant applications such as video coding. The video compression is error-tolerant in nature since the only requirement is to produce output that has sufficient quality to provide good user experience. Therefore, approximate computing has a huge potential to improve the performance, area and/or power consumption of hardware implementations. FPGAs serve as an excellent platform for a wide range of applications from small-scale embedded devices to high-performance computing systems due to their short time-to-market, enhanced flexibility and run-time re-configurability. However, despite supporting specialized hardware accelerators and co-processors, FPGA-based systems typically consume more power and/or energy, compared to their ASIC counterparts. Therefore, besides employing traditional energy optimization techniques, there is a need for exploring new avenues in energy-efficient computing solely for FPGA-based systems. One such attractive trend is the Approximate Computing paradigm, which is re-emerging due to the breakdown of Moore’s law and Dennard scaling, and the ever-increasing demand for high-performance and energy efficiency. Approximate computing sacrifices the accuracy and precision of intermediate or final calculations in order to achieve noteworthy improvements in critical path delay, area, power and/or energy usage. This balance is advantageous for applications that have built-in resilience, meaning they can still generate acceptable results even if some calculations are incorrect due to approximations. A wide range of applications like image and video processing, data mining, machine learning, etc., in the recognition, mining and synthesis domains exhibit this property. Current methods for approximate computing can be used in various parts of the computing system, including hardware design and software development. There is an extensive amount of research related to approximations at both hardware and software layers.

Voltage over-scaling and functional approximation are the two primary approximate computing techniques utilized in hardware design. Approximate adders can be broadly classified into the following categories: segmented adders, which divide n-bit adder into several r-bit adders operating in parallel; speculative adders, which predict the carry using only the few previous bits; and approximate full-adder based adders, which approximate the accurate full-adder at transistor or gate level. Typically, segmented and speculative adders exhibit greater speed and require larger chip areas compared to precise adders. Approximate n-bit adders use a combination of m-bit approximate adder in the least significant part (LSP) and (n − m)-bit accurate adder in the most significant part (MSP), as shown in Figure 1.

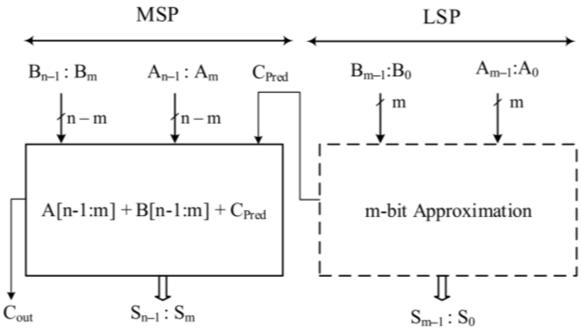


Fig 1: Generalized Approximate Adder

The basic element of an ASIC implementation is a logic gate, whereas FPGAs use lookup tables (LUTs) to implement logic functions. Therefore, ASIC based optimization techniques cannot be directly mapped to FPGAs. FPGAs are commonly utilized for executing error-tolerant tasks through addition and multiplication functions. Approximate computing can enhance the efficiency of FPGA-based implementations for these applications. There have been only a small number of approximate adders designed specifically for FPGAs in previous research. These approximate adders focus on improving either the efficiency or accuracy. Therefore, the design of low error efficient approximate adders for FPGAs is an important research topic. We additionally suggest a power-efficient and space-efficient approximate adder (APEX) designed for FPGAs. Although its MSE is higher than that of LEADx, it is lower than that of the approximate adders in the literature. It has the same area, lower MSE and less power consumption than the smallest and lowest power consuming approximate adder in the literature. It possesses a smaller size and consumes less power compared to the other similar adders discussed in the literature.

II. LITERATURE SURVEY

W. Ahmad and I. Hamzaoglu. In this article, we introduce a high-performance SAD hardware design with minimal maximum and average errors for FPGAs. The suggested rough SAD hardware makes use of the unused LUT inputs to cut down on area and power usage while still delivering a near precise outcome. The suggested SAD hardware demonstrates lower maximum and average error when compared to the SAD hardware discussed in existing literature. It utilizes fewer LUTs, up to 20% less, compared to the smallest approximate SAD hardware described in existing literature. It uses up to 20% less LUTs than the smallest approximate SAD hardware in the literature. It consumes up to 38% less power than the lowest power consuming approximate SAD hardware in the literature [1].

P. Balasubramanian, R. Nayar, D. L. Maskell, and N. E. Mastorakis. This paper proposes an innovative approximate adder with optimized error metrics and design optimization. Design optimization refers to maximizing design characteristics including power, delay, and area/resources. Error metrics optimization indicates whether the suggested approximate adder is realistic for usage in real-world scenarios. For short, we call the proposed approximate adder with a near-normal error distribution and hardware optimization HOAANED. We take into consideration the ASIC and FPGA design contexts for the implementation of HOAANED and the other approximation adders. For the FPGA implementation, we used a Xilinx Artix-7 device; for the ASIC based implementation, we used a 32/28-nm CMOS standard digital cell library. In order to illustrate the usefulness of HOAANED in practice, we looked at digital image processing as a real implementation and carried out numerous imaging experiments. Using the peak signal-to-noise ratio (PSNR) as a qualitative figure-of-merit, the quality of the images reconstructed using the approximation adders was assessed. It is discovered that, when compared to the other approximate adders for image processing, HOAANED produces an increased PSNR. This is made possible by HOAANED's simultaneous optimization of design metrics and error characteristics [2].

C. Niemann, M. Rethfeldt, and D. Timmermann. For applications that can accept a certain level of imprecision, approximate or inexact arithmetic is a viable path toward reduced power usage. This is true for image and audio processing as well, since human perception is not very precise. Furthermore, since these applications are naturally tolerant of a certain degree of error, other uses like neural networks or artificial intelligence processing can also profit from this type of arithmetic. Multipliers are among the most important parts of arithmetic circuits in terms of space, power, and latency. Numerous advanced methods for estimating multipliers have already been released for ASICs. Nevertheless, these ASIC methods perform poorly when combined with the particular Lookup-Table (LUT)-based FPGA design. There is a severe lack of approximation design as FPGAs become more and more important for applications like signal processing approach pertaining to FPGAs. We put forth a method for approximating signal processing that is especially designed for the FPGAs' LUT-based hardware. It reduces energy requirements and enables notable performance gains. In comparison to the Xilinx Vivado multiplier IP core, we obtain a 45.9% reduction in LUT area and a 30.6% decrease in delay, all while introducing a negligible average relative error of only 0.14%. Our suggested layout can be seen at https://github.com/niemann-c/approx-mult-for-fpga. It is free to use [3].

E. Kalali and I. Hamzaoglu. In this paper, an surmised Tall Productivity Video Coding (HEVC) intra precise forecast procedure is proposed for diminishing region of HEVC intra forecast equipment. The proposed guess procedure employments closer neighboring pixels rather than removed neighboring pixels in intra precise forecast conditions. It causes 0.0569% bit rate increment and 0.0028 dB PSNR misfortune on normal. In this paper, an surmised HEVC intra precise forecast equipment executing the proposed estimation method is additionally proposed. FPGA and ASIC executions of the proposed surmised equipment can handle 24 and 40 quad full HD (3840 × 2160) video outlines per moment, separately. The proposed surmised HEVC intra precise expectation equipment is the littlest and the moment speediest HEVC intra forecast equipment within the writing. It is ten times smaller and 20% slower than the speediest HEVC intra expectation equipment within the writing [4].

N. Van Toan and J.-G. Lee. In this research, approximation multipliers with varying degrees of precision are presented, which are effectively implemented on Field Programmable Gate Arrays (FPGAs) through the use of recently suggested approximate logic compressors. Compared to the state-of-the-art works, our approximate multiplier designs provide larger power-delay-area products (PDAP) increases at similar accuracies. Furthermore, our solutions outperform the Lookup Table based multiplier intellectual properties available on an FPGA in terms of delay, occupied area, and dynamic power dissipation. Specifically, PDAP improvements up to 7.1 x, 8.3 x, and 5.0 x can be achieved by our suggested 8-, 16-, and 32-bit multipliers. Applications in image processing, such sharpening and picture multiplication, further show how useful and applicable our designs are. Experiments indicate that for the picture In terms of sharpness, our 8 x 8 multipliers can produce a structural similarity index metric (SSIM) of 0.9989, a strong peak signal-to-noise ratio (PSNR) of 46.81 dB, and a dynamic power reduction of up to 36.7% depending on the specific multiplier. Approximate 16 x 16 multipliers can provide a high dynamic power saving of up to 58.15%, an SSIM of 1.0, and a high PSNR of 80.25 dB for the image multiplication. The suggested multipliers should work well in high-performance, low-power, error-resistant applications based on these experiments [5].

F. Ebrahimi-Azandaryani, O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram. This brief presents a block-based carry-lookahead approximate adder with low energy consumption. The design is formed by dividing the adder into exclusive summation blocks, which can be chosen from both carry propagate and parallel-prefix adders. Here, the carry output of each block is predicted using the input operands of the block and the next block. In this adder, the carry chain length is decreased to two blocks at most, with typically only one block used to compute the carry output, resulting in a shorter average delay. Furthermore, to enhance precision and lower the output error rate, a mechanism for error detection and recovery is suggested. The efficiency of the suggested approximate adder is evaluated against advanced approximate adders using a cost function that considers energy, delay, area, and output quality. The findings show a 50% decrease in cost function when compared to alternative adders [6].

W. Ahmad, B. Ayrancioglu, and I. Hamzaoglu. Calculating motion is the most demanding part of video encoder hardware in terms of computing power and energy consumption. Roughly built hardware can outperform precise hardware in terms of performance, size, and power usage for error-tolerant applications with satisfactory quality. This article presents a proposed approximate adder. A thorough analysis is provided comparing various approximate circuits, including the new approximate adder and the conventional bit truncation method, for motion estimation in H.264 and HEVC. The suggested approximate adder reduced power consumption by 10% in motion estimation hardware, surpassing the quality of other approximate circuits [7].

C. K. Jha, K. Prasad, A. S. Tomar, and J. Mekie. In recent years, there has been a significant increase in the popularity of approximate circuits for ASICs because of the advantages they offer in energy efficiency and performance, while maintaining high output quality. Approximating on FPGAs is still difficult because of the increased level of detail in logic implementation on these devices. The look up tables (LUTs) are the smallest configurable units in FPGAs that are utilized for logic implementation. In this article, we utilize the built-in structures found in FPGAs for carrying out the implementation of SEDAAF. SEDAFF is an approximate adder that can be configured at runtime to either perform a one-bit precise addition or a two-bit approximate addition using the same hardware. SEDAF also has a highest limited inaccuracy, meaning that for an n-bit adder, if m-bits are estimated, the maximum error is equal to 2 m − 1. Compared to current designs, SEDAAF uses 25% less power and has a power delay product that is 17% lower. SEDAF surpasses current top designs in output quality for Sobel edge detection and is suitable for approximate processors for exact and approximate additions [8].

III. EXISTING METHOD

In the existing method, an approximate adder LOA (lower-part OR adder) is designed. The lower-part OR adder (LOA) is an approximate adder design implemented using an approximate FA for the least significant bits (LSBs) of a multi-bit adder. The LOA consists of two parts: an accurate part and an inaccurate part. The former part uses a traditional precise adder, such as the ripple carry adder (RCA) and carry-look-ahead adder (CLA), to calculate the most significant bits (MSBs) with no computation error. Whereas, the latter part only uses an OR operation to approximately obtain LSB summations. Moreover, the result of a logical AND calculation for the most significant bit (MSB) input pair in the imprecise section is used as a carry input for the precise section.

Disadvantages

a. Area occupied on the hardware is high.

b. Energy consumption is high and accuracy is very low

c. Delay increases as the computation time of output increases.

IV. PROPOSED METHOD

In the proposed system, a methodology to reduce the error of approximate adders by efficiently utilizing FPGA resources, such as unused LUT inputs has been proposed. We propose a low error and area efficient approximate adder (LEADx) for FPGAs.

Approximate computing is a new design technique that trades off accuracy for performance, area and/or power consumption for error-tolerant applications such as video coding. The video compression is error-tolerant in nature since the only requirement is to produce output that has sufficient quality to provide good user experience. Therefore, approximate computing has a huge potential to improve the performance, area and/or power consumption of hardware implementations.

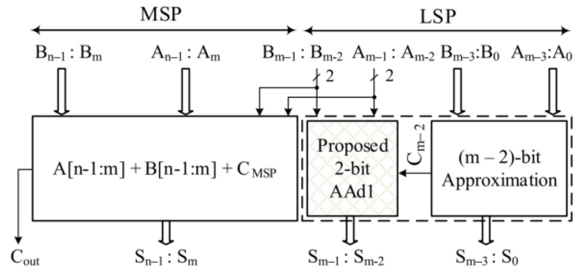


Fig 2: Proposed Approximate Adder

Figure 2 represents the proposed model of Approximate Adder. It achieves better quality than the other approximate adders for video encoding application. We additionally suggest an efficient approximate adder (APEX) for FPGAs that saves on both area and power. While its mean squared error surpasses LEADx, it falls below the approximate adders in traditional designs. It has the same area, lower MSE and less power consumption than the smallest and lowest power consuming approximate adder in the literature. It has a reduced area and decreased power usage compared to other approximate adders found in traditional adders.

The proposed LEADx approximate adder can be seen in the diagram below. A n-bit LEADx utilizes d(m − 2)/2e instances of AAd2 adder in the lower m − 2 bits of the approximate adder design illustrated in Figure 3.

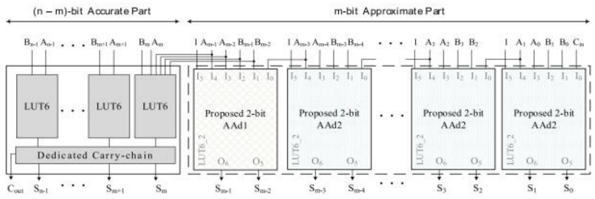

(n – m)-bit Accurate Part
m-bit Approximate Part
B_{n-1} A_{n-1} . . . B_{m+1} A_{m+1} B_m A_m
1 A_{m-1} A_{m-2} B_{m-1} B_{m-2}
1 A_{m-3} A_{m-4} B_{m-3} B_{m-4}
1 A_3 A_2 B_3 B_2
1 A_1 A_0 B_1 B_0 C_{in}
LUT6
LUT6
LUT6
Dedicated Carry-chain
I_5 I_4 I_3 I_2 I_1 I_0
Proposed 2-bit AAd1
Proposed 2-bit AAd2
Proposed 2-bit AAd2
Proposed 2-bit AAd2
LUT6_2
O_6 O_5
C_{out} S_{n-1} . . . S_{m+1} S_m
S_{m-1} S_{m-2} S_{m-3} S_{m-4} . . . S_3 S_2 S_1 S_0

Fig 3: Proposed LEADx

proposed approximate adder. In LEADx, Cm−2 = Am−3. AAd2 implements a 5-to-2 logic function that is mapped to a single LUT. Similarly, AAd1 is also mapped to a single LUT. Therefore, dm/2e LUTs are used for the LSP. These LUTs work in parallel. Hence, the latency of LSP is the same as the latency of a lone LUT (tLUT). The crucial pathway of LEADx extends from the input Am−2 to the output Sn−1.

In the proposed APEX, the S0 to Sm−3 outputs are fixed to 1 and the Cm−2 is 0. This provides significant area and power consumption reduction at the expense of slight quality loss. It is important to note that this is different from bit truncation technique which fixes both the sum and carry outputs to 0. The ME of truncate adder is 2m+1 − 2 which is much higher than ME of APEX (2m−2 − 1). The proposed APEX approximate adder is shown in Figure 3. Same as LEADx, the critical path of APEX is from the input Am−2 to the output Sn−1.

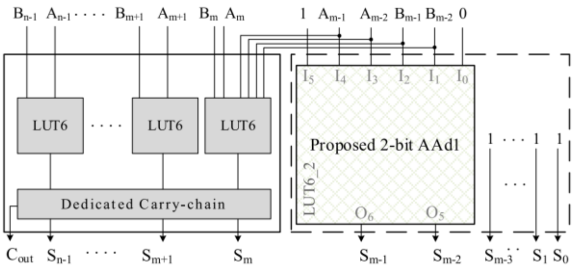

B_{n-1} A_{n-1} · · · · B_{m+1} A_{m+1} B_m A_m
1 A_{m-1} A_{m-2} B_{m-1} B_{m-2} 0
I_5 I_4 I_3 I_2 I_1 I_0
LUT6
LUT6
LUT6
Proposed 2-bit AAd1
LUT6_2
O_6 O_5
1 · · · 1 1
Dedicated Carry-chain
C_{out} S_{n-1} · · · · S_{m+1} S_m S_{m-1} S_{m-2} S_{m-3} S_1 S_0

Fig 4: Proposed APEX

V. RESULTS AND DISCUSSIONS

The below table presents the implementation results of 16-bit adders using an 8-bit approximation. Input and output registers are used in the implementation of every adder. Because of carry propagation in their respective LSPs, SEDA and LBA operate more slowly than the accurate adder. The delay of the accurate adder is the same for the other 16-bit approximate adders. It is significant to remember that the highest frequency of the Virtex 7 FPGA limits their delay. This does not imply that these adders' critical paths are identical.

Table 1: FPGA implementation results of 16-bit adders with 8-bit approximation.

Adder	LUTs		Delay (ns)	Power (mW)
	MSP	LSP		
Accurate	8	8	1.35	6.16
LEADx	8	4	1.35	6.09
APEx	8	1	1.35	4.32
AFA [17]	8	5	1.35	6.17
DeMAS–2 [20]	8	5	1.35	6.12
HOAANED [16]	8	1	1.35	4.52
LBA [19]	8	11	1.56	6.10
LOA [14]	8	4	1.35	5.66
SEDA [22]	8	6	1.35	6.16

All the approximate 16-bit adders, except LBA, use fewer LUTs than the accurate adder. Since an accurate adder is used in the MSP of all these adders, the reduction in LUTs occurs only in the LSP. Since LEADx performs 2-bit addition in a single LUT, its LSP uses 50% fewer LUTs than the accurate adder. APEX and HOAANED utilize the fewest LUTs. The two adders experience a notable decrease in LUTs due to the incorporation of constant functions in their LSPs.

The decrease in the number of LUTs in different approximate adders is due to the utilization of approximation techniques, enabling the synthesis tool to combine two sum outputs into one LUT. LEADx consumes slightly less power than the accurate adder. APEX consumes the lowest power among all the approximate adders. For the 16-bit adder with 8-bit approximation, the power consumption of APEX is 29% less than that of the accurate adder and 4.5% less than that of the second lowest power consuming adder, HOAANED.

These findings demonstrate that compared to the FPGA-specific adders in the literature, our suggested LEADx has a smaller size, lower power, and superior quality. The findings demonstrate that, among FPGA-specific approximate adders in the literature, DeMAS is the most effective. The RTL Schematic of the LEADx will be presented at Figure 5.

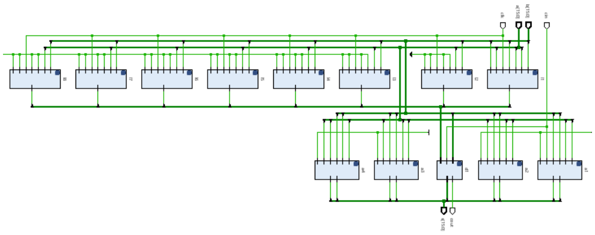
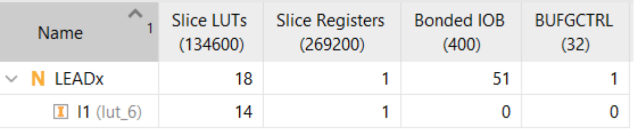


Fig 5: RTL Schematic of LEADx

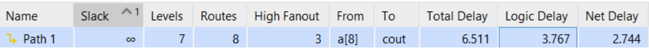
Compared to DeMAS, LEADx has an 86% lower MSE and a 7% smaller area using an 8-bit approximation. Area consumed by the LEADx will be me ntioned at Figure 6. According to the research, LOA is among the most effective ASIC-based approximation adders. When implemented on an FPGA, LEADx outperforms LOA in terms of quality for the same cost.



Name	Slice LUTs (134600)	Slice Registers (269200)	Bonded IOB (400)	BUFGCTRL (32)
N LEADx	18	1	51	1
I1 (lut_6)	14	1	0	0

Fig 6: Area of LEADx

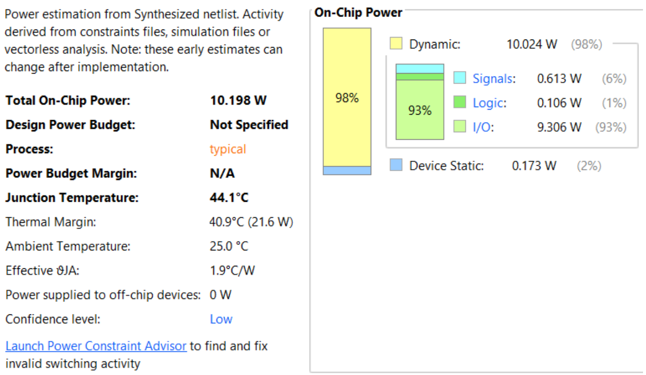
By talking about the three factors that used to be decreased of improve the performance of an approximate adder i.e., area, power and delay. The proposed values of Delay and power will be taken as the given in Figure 7.



Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay
Path 1	∞	7	8	3	a[8]	cout	6.511	3.767	2.744

Fig 7: Delay of LEADx

LEADx has an 87% lower MSE than LOA at the same cost using an 8-bit approximation. HOAANED is appropriate for implementation on FPGA. The power consumption of LEADx will be given at Figure 8.



Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**10.198 W**
Design Power Budget:	**Not Specified**
Process:	typical
Power Budget Margin:	**N/A**
Junction Temperature:	**44.1°C**
Thermal Margin:	40.9°C (21.6 W)
Ambient Temperature:	25.0 °C
Effective ϑJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

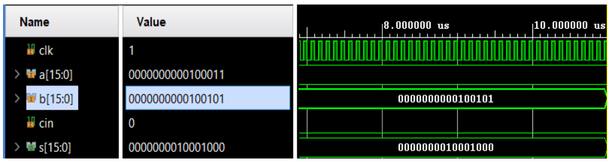
Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

Dynamic:	10.024 W	(98%)	
	Signals:	0.613 W	(6%)
	Logic:	0.106 W	(1%)
	I/O:	9.306 W	(93%)
Device Static:	0.173 W	(2%)	

Fig 8: Power of LEADx

When implemented on an FPGA, APEX, however, uses less power and produces better results than HOAANED at the same cost. APEX possesses over 60% lower MSE than HOAANED at the same cost. The Behavioural simulation of the LEADx will be illustrated at Figure 9.



Name	Value
clk	1
a[15:0]	0000000000100011
b[15:0]	0000000000100101
cin	0
s[15:0]	0000000010001000

Fig 9: Simulation results of LEADx

APEX and HOAANED exhibit the most efficient utilization of LUTs, with their usage being the lowest among all adders. This reduction in LUTs is primarily attributed to the integration of constant functions in their LSPs. Conversely, other approximate adders achieve LUT reduction through approximation techniques, enabling synthesis tools to consolidate two sum outputs into a single LUT. The RTL Schematic of APEX will be mentioned at Figure 10.

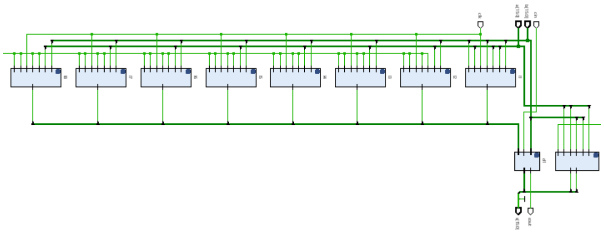
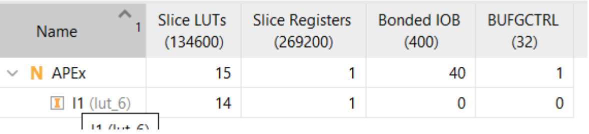


Fig 10: RTL Schematic of APEX

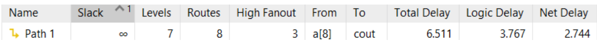
LEADx demonstrates a marginally lower power consumption than the accurate adder, while APEX stands out as the most power-efficient among all approximate adders. Specifically, in a 16-bit adder with 8-bit approximation, APEX consumes 29% less power than the accurate adder and 4.5% less than the second-lowest power-consuming adder, HOAANED. The Area of APEX will be shown at Figure 11.



Name	Slice LUTs (134600)	Slice Registers (269200)	Bonded IOB (400)	BUFGCTRL (32)
N APEx	15	1	40	1
I1 (lut_6)	14	1	0	0

Fig 11: Area of APEX

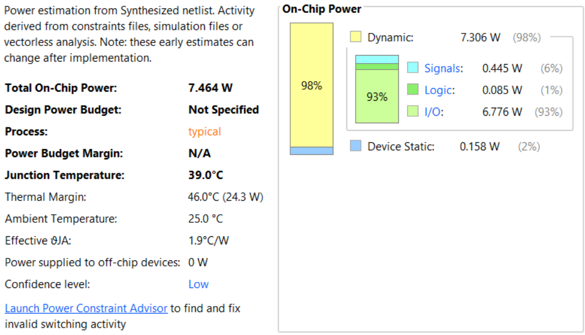
While most approximate adders exhibit a linear decrease in LUTs with increasing approximation levels, the corresponding power reductions do not consistently follow this trend. However, APEX notably achieves significant power reduction compared to the accurate adder, albeit with a slight compromise in accuracy. The Delay of APEX will be mentioned at Figure 12



Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay
Path 1	∞	7	8	3	a[8]	cout	6.511	3.767	2.744

Fig 12: Delay of LEADx

By covering approximation levels from 4-bit to 20-bit within a 32-bit adder. Notably, the accurate 32-bit adder utilizes 32 LUTs and consumes 10.75 mW of power. While most approximate adders exhibit a linear decrease in LUTs with increasing approximation levels, the corresponding power reductions do not consistently follow this trend. The power consumption of LEADx will be given at Figure 13.



Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**7.464 W**
Design Power Budget:	**Not Specified**
Process:	typical
Power Budget Margin:	**N/A**
Junction Temperature:	**39.0°C**
Thermal Margin:	46.0°C (24.3 W)
Ambient Temperature:	25.0 °C
Effective ϑJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

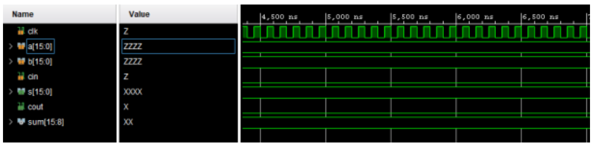
Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

Dynamic:	7.306 W	(98%)	
	Signals:	0.445 W	(6%)
	Logic:	0.085 W	(1%)
	I/O:	6.776 W	(93%)
Device Static:	0.158 W	(2%)	

Fig 13: Power of APEX

However, APEX notably achieves significant power reduction compared to the accurate adder, albeit with a slight compromise in accuracy. The reductions in LUTs, power consumption, and delay achieved by 64-bit approximate adders with 16-bit approximation compared to the accurate 64-bit adder. LEADx reduces LUTs by 12.5% compared to the accurate adder, whereas APEX surpasses with a 23.4% reduction in LUTs and a 21% decrease in power consumption compared to the accurate adder. The simulation results of APEX will be given at Figure 14.



Name	Value
clk	Z
a[15:0]	ZZZZ
b[15:0]	ZZZZ
cin	Z
s[15:0]	XXXX
cout	X
sum[15:8]	XX

Fig 14: Simulation results of APEX

Table 2: Evaluation table that shows the values of the three factors for the both LEADx and APEX

Name	Area (LUT's)	POWER	Delay (ns)
LEADx	18	10.198	6.511
APEX	15	7.464	6.511

The Above table shows the parameters Area, Power and Delay of the Approximate Adders LEADx and APEX.

VI. CONCLUSION

In this paper, two low error efficient approximate adders for FPGAs are proposed, where approximation is done only in the LSP and the MSP is kept accurate. The first approximate adder, LEADx, achieves better area and delay compared to exact adder implementations. The second approximate adder, APEX, has reduced area, and less power consumption than the existing adders.

It has smaller area and lower power consumption than the other approximate adders in the literature. Hence, the suggested approximate adders are suitable for implementing error-tolerant applications on FPGAs.

VII. REFERENCES

[1] G. A. Gillani, M. A. Hanif, B. Verstoep, S. H. Gerez, M. Shafique, and A. B. J. Kokkeler, ''MACISH: Designing approximate MAC accelerators with internal-self-healing,'' IEEE Access, vol. 7, pp. 77142–77160, 2019.

[2] E. Kalali and I. Hamzaoglu, ''An approximate HEVC intra angular prediction hardware,'' IEEE Access, vol. 8, pp. 2599–2607, 2020.

[3] T. Ayhan and M. Altun, ''Circuit aware approximate system design with case studies in image processing and neural networks,'' IEEE Access, vol. 7, pp. 4726–4734, 2019.

[4] W. Ahmad and I. Hamzaoglu, ''An efficient approximate sum of absolute differences hardware for FPGAs,'' in Proc. IEEE Int. Conf. Consum. Electron. (ICCE), Las Vegas, NV, USA, Jan. 2021, pp. 1–5.

[5] H. Jiang, C. Liu, L. Liu, F. Lombardi, and J. Han, ''A review, classification, and comparative evaluation of approximate arithmetic circuits,'' ACM J. Emerg. Technol. Comput. Syst., vol. 13, no. 4, pp. 1–34, Aug. 2017.

[6] A. C. Mert, H. Azgin, E. Kalali, and I. Hamzaoglu, ''Novel approximate absolute difference hardware,'' in Proc. 22nd Euromicro Conf. Digit. Syst. Design (DSD), Kallithea, Greece, Aug. 2019, pp. 190–193.

[7] N. Van Toan and J.-G. Lee, ''FPGA-based multi-level approximate multipliers for high-performance error-resilient applications,'' IEEE Access, vol. 8, pp. 25481–25497, 2020.

[8] L. Chen, J. Han, W. Liu, P. Montuschi, and F. Lombardi, ''Design, evaluation and application of approximate high-radix dividers,'' IEEE Trans. Multi-Scale Comput. Syst., vol. 4, no. 3, pp. 299–312, Jul. 2018.

[9] A. K. Verma, P. Brisk, and P. Ienne, ''Variable latency speculative addition: A new paradigm for arithmetic circuit design,'' in Proc. Design, Automat. Test Eur. (DATE), Munich, Germany, Mar. 2008, pp. 1250–1255.

[10] N. Zhu, W. L. Goh, G. Wang, and K. S. Yeo, ''Enhanced low-power highspeed adder for error-tolerant application,'' in Proc. Int. SoC Design Conf., Incheon, South Korea, Nov. 2010, pp. 323–327.

[11] A. B. Kahng and S. Kang, ''Accuracy-configurable adder for approximate arithmetic designs,'' in Proc. 49th Annu. Design Automat. Conf. (DAC), New York, NY, USA, 2012, pp. 820–825.

[12] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, ''Low-power digital signal processing using approximate adders,'' IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.

[13] W. Ahmad, B. Ayrancioglu, and I. Hamzaoglu, ''Comparison of approximate circuits for H.264 and HEVC motion estimation,'' in Proc. 23rd Euromicro Conf. Digit. Syst. Design (DSD), Kranj, Slovenia, Aug. 2020, pp. 167–173.

[14] 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 Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[15] A. Dalloo, A. Najafi, and A. Garcia-Ortiz, ''Systematic design of an approximate adder: The optimized lower part constant-OR adder,'' IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 26, no. 8, pp. 1595–1599, Aug. 2018.

[16] P. Balasubramanian, R. Nayar, D. L. Maskell, and N. E. Mastorakis, ''An approximate adder with a near-normal error distribution: Design, error analysis and practical application,'' IEEE Access, vol. 9, pp. 4518–4530, 2021.

[17] S. Dutt, S. Nandi, and G. Trivedi, ''Analysis and design of adders for approximate computing,'' ACM Trans. Embedded Comput. Syst., vol. 17, p. 40, Dec. 2017.

[18] D. Celia, V. Vasudevan, and N. Chandrachoodan, ''Optimizing poweraccuracy trade-off in approximate adders,'' in Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE), Dresden, Germany, Mar. 2018, pp. 1488–1491.

[19] A. Becher, J. Echavarria, D. Ziener, S. Wildermann, and J. Teich, ''A LUT based approximate adder,'' in Proc. IEEE 24th Annu. Int. Symp. Field Program. Custom Comput. Mach. (FCCM), Washington, DC, USA, May 2016, p. 27.

[20] B. S. Prabakaran, S. Rehman, M. A. Hanif, S. Ullah, G. Mazaheri, A. Kumar, and M. Shafique, ''DeMAS: An efficient design methodology for building approximate adders for FPGA-based systems,'' in Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE), Dresden, Germany, Mar. 2018, pp. 917–920.

[21] S. Boroumand, H. P. Afshar, and P. Brisk, ''Approximate quaternary addition with the fast carry chains of FPGAs,'' in Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE), Dresden, Germany, Mar. 2018, pp. 577–580.