Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 9, September : 2023

DELAY ANALYSIS OF APPROXIMATE PARALLEL PREFIX ADDERS

N.Chandu Research scholar, Lendi Institute of Engineering & Technology, navuduchandu@gmail.com.

B.Ramamohan Associate Professor, Lendi Institute of Engineering & Technology, ramamohan.b@lendi.org.

ABSTRACT

Verily, addition units doth find wide usage in the realm of computational kernels, serving faithfully in the pursuit of error-tolerant applications such as the noble arts of machine learning and the processing of signals, images, and videos. Besides their standalone usage, additions are crucial building blocks for subtraction, comparison, multiplication, squaring, and division. Parallel prefix adders (PPAs) are the fastest adders. It shows a parallel prefix graph with carry operator and high prefix operator nodes. PPAs improve the parallelization of carry generation (G) and propagation (P), making them among the fastest adders.

We introduce AxPPAs, approximate PPAs, by using PO approximations in this noble quest. To evaluate our concept for approximate POs (AxPOs), we create four AxPPAs: approximate Brent–Kung (AxPPA-BK), approximate Kogge–Stone (AxPPAKS), Ladner-Fischer (AxPPA-LF), and Sklansky. Four AxPPA designs with latency approximation adders (AxAs) are compared. these designs art designed using Verilog HDL language, and sooth, they art simulated and synthesised on Xilinx ise design.

Keywords: APX BKA, APXKSA, APXLFA, APXSSA and VLSI Architecture.

1.INTRODUCTION

Verily, the concept of APPROXIMATE computing, known as AxC, hath emerged as a novel design paradigm. Its purpose is to enhance efficiency throughout the computing stack by harnessing the innate resilience to faults found in many applications. Verily, AxC doth introduce accuracy, a novel dimension of import, to the realm of trade-offs in design optimisations. It might drastically minimise VLSI circuit area and computational energy. Many error-resilient and computationally demanding applications, such as the processing of digital signals, the manipulation of images and videos, the creation of consumer electronics,

The parallel datapath architecture of hardware accelerators requires many adder units for computers to observe and people to learn from machines. The adder unit is a fundamental component of many other important arithmetic hardware operators because of its widespread use. This means that from now on, estimated adder units will be known as AxA, may be employed in harmonious collaboration within numerous extant units of approximate arithmetic, of greater intricacy, for the purpose of cross-layer approximations. These include modules for squaring, multipliers, square roots, and dividers.

Verily, numerous tomes have put forth grand designs for the structures of AxAs. The AxAs are a low-power approximation of an LSB-MSB logic. The PPAs are, without a doubt, the most rapid and effective adders currently available. Forsooth, they employ the art of logarithmic reduction, where the carry propagation channels are tailored to reduce latency on the most important routes. Indeed, digital hardware design presents a significant difficulty in mastering the art of optimising circuit synthesis of PPAs. The essay in question introduces a concept known as AxPPAs, or approximation PPAs. AxPPAs use logical approximations from least to most significant bits and fast carry propagation. Thus, they

doth offer a hybrid solution, catering to both speed and energy efficiency in the realm of adders.

Verily, our notion of uniting swiftness and efficiency in the realm of PPA approximations doth reveal a novel AxA Pareto front, wherein circuit area and energy diminution are intertwined, all whilst maintaining an equal measure of approximation (that is to say, the frequency of errors). The core of our argument is the great notion of estimating carry propagation (P) and generation (G) of prefix operators (POs). We will design and test four AxPPAs based on the Brent Kung (BK), Kogge Stone (KS), Ladner-Fischer (LF), and Sklansky (SK) architectures to prove approximation POs (AxPOs) are feasible.

To demonstrate their utility, we doth engage in the usage of these proposed AxPPAs in various applications, encompassing a multitude of adders, verily, the ensuing hardware accelerators, as comprehensive instances for study: 1) In video processing, block matching uses either 1) a FIR filter or 2) an SSD pixel comparison. Verily, the case studies of the FIR filter and SSD applications doth bear great relevance, forsooth. For in these circuits, a multitude of adders doth residein a way that has real world consequences for distance, time, and energy loss. To calculate the inner product of a vector multiplication, the FIR filter uses adders. The hardware design of SSDs is based on a summation tree that adds up partial-value computations.

2.LITERATURE SURVEY

Xu, M. Todd, and S. K. Nam hath composed a scholarly work entitled "Approximate computing: A survey." Verily, in the realm of error-tolerant applications, one doth employ approximate multipliers. Such devices doth willingly forsake the precision of outcomes, all in the noble pursuit of reducing power consumption or delay.

In this parchment, we dost embark upon a quest to explore the realm of approximate multipliersand using the high-minded method of static segmentation to do it. A tiny mm internal multiplier is fed the two segments as input, and the result is then shifted ingeniously The approximation error may now be reduced with less strain on hardware resources, thanks to the simple and effective correction strategies we provide. To help us decide whether or not to pursue a hardware implementation in the high-minded world of 28 nm technology, we compare our suggested approximation multiplier to others that have been offered in the past.

When using the suggested correction approach, static segmented multipliers are found to be on the Pareto-optimal frontier, as shown by the comparison. whether it be in close proximity or directly upon it. This is the case for both the normalised mean error distance against power trade-off plot and the mean relative error distance versus power trade-off plot. In light of this, these multipliers are strong candidates for use in situations where their error performance is tolerated. Indeed, this is supported by findings from worthy endeavours like image processing and picture categorization.

3.PERFORMANCE ANALYSIS OF PPAPX ADDERS

The pursuit of perfecting the design of PPAs hath been thoroughly examined, with great diligence, in order to attain the optimisation of data paths. PPA designs include three main blocks: preprocessing, prefix computation, and postprocessing as depicted in the illustrious Figure 1(a). Indeed, the first order of business is preprocessing, from which signals are crafted incrementally for subsequent efforts: 1twofold: 1) to initiate a carry (g), and 2) to spread a carry (p). According to the following Boolean equations, the A and B input bits of the operands are encoded as (g) (generate) and (p) (propagate) during preprocessing.

$$pi = Ai \oplus Bi \qquad (1)$$

$$gi = Ai \cdot Bi \, . \qquad (2)$$

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 9, September : 2023

Verily, A noble truth says the carry-out will be true regardless of the carry-in. A dependable indicator for ith-order carry-in to carry-out transmission. A bitwise AND logic gate and an XOR logic gate evaluate the g and p functions in parallel for all i-order bits with one gate delay. Verily, the circuit of preprocessing doth surely grow in size, in direct proportion to the width of the input bits of the adder.The literature has extensively studied the accurate prefix computing step to optimise energy, circuit area, and latency.Prefix computing may be done in many ways while coordinating circuit area, logic depth, logic gates, fan-out per gate, and connections between carry propagate and produce cells.

The prefix computation gathers the carries and arranges them according to the adders' configuration. POs are the fundamental blocks of the prefix computing phase, and boolean equations (3) and (4) characterise them. The preprocessing stage provides the words gi, pi+1, and gi+1. The PO blocks must include the associative operator, which generates and propagates carry bits. This indicates that the PO blocks in boolean equations (3) and (4) are intertwined in the prefix calculation, which determines the PPA graph topology.

$$P = pi \cdot pi+1 \quad (3)$$

$$G = (gi \cdot pi+1) + gi+1. \quad (4)$$

For sure, The graph organises carry generation and propagation nodes and calculates prefixes. The PO, or delta operator in mathematics, is a basic carry operator. The prefix computing step produces "carry," often known as "C" (Carry out). According to (5), the final total is calculated by recombining the computational prefix C with the preprocessing p. As can be seen in (6), the last processing step includes simultaneously applying an XOR gate to signals C and p for all bits in the ith order.

$$Ci = Cin \cdot (Pi + Gi) \quad (5)$$

$$Si+1 = Pi+1 \oplus Ci\ . \quad (6)$$

Verily, behold Fig. 2, which doth present unto thee four splendid exemplars of PPAs, as if plucked from the annals of antiquity: 1) Brent-Kung, a noble algorithm of great renown, 2) Kogge-Stone, a method of computation known, 3) Sklansky, a technique of numerical might, 4) Ladner-Fischer, a strategy of wondrous insight. Verily, the strategy of Brent-Kung doth gather the computation of prefixes for groups of two bits, and doth employ it to discover prefixes for groups of four bits, and further still, doth employ it to discover prefixes for groups of eight bits.

Indeed, the Kogge-Stone does bring about a miraculous synthesis of effectiveness and reduced fan-out. The noble tree really does have cells that reproduce and spread in parallel. Indeed, the circuit size does rise when the adder's layout is arranged in a regular grid, forsooth, due to the vast number of possible routes. The delay caused by computing intermediate prefixes is reduced thanks to the Sklansky adder, which lives up to its name as a divide-and-conquer adder. Indeed, this mission is finished, but at a high cost, forsooth, a spread that doubles at each level. The performance of this adding cell may degrade with such high fan-out values. Indeed, the Sklansky adder's architecture is comparable to Sklansky's and displays reasonable regularity between Sklansky and Brent-Kung's PPAs. This adder computes odd integer prefixes and represents even digits with an extra step.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 9, September : 2023

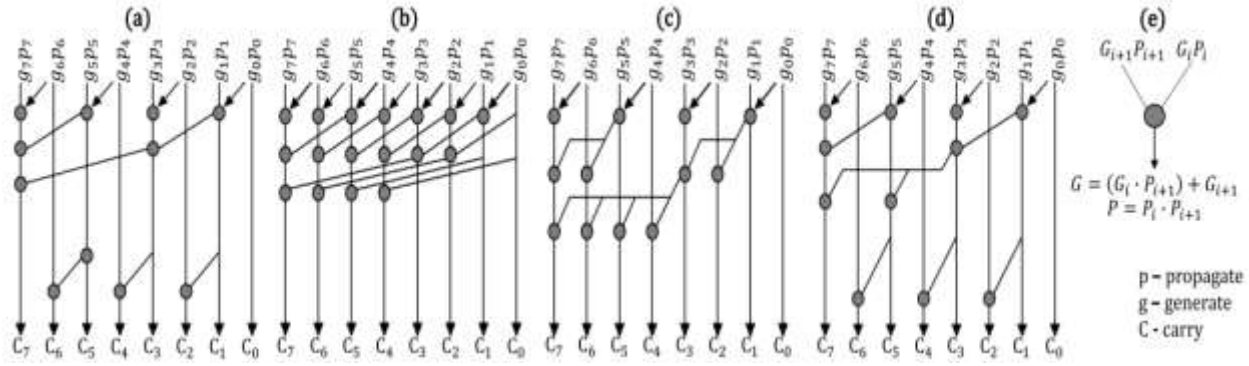

Fig. 1. Examples of the prefix computing architectures: (a) Brent–Kung, (b) Kogge–Stone, (c) Sklansky, and (d) Ladner-Fischer. (e) POs.

Prefix computations in a PPA really are sets of POs. Indeed, approximations are used in the reasoning behind several of AxPPA's POs [see Fig. 3(c)]. During the planning stage, the number of estimated POs may be adjusted to get the required accuracy of the AxPPA. The calculation of POs, as seen in Fig. 1(f), is described in (3) and (4), whereas the computation of AxPOs is described in (7) and (8).

$$P \approx pi+1 \qquad (7)$$

$$G \approx gi+1. \qquad (8)$$

The prefix calculation stage of AxPPA is really free of logic gates thanks to AxPPA. It is true that our modest proposal eliminates the need for PPA prefix calculation by eliminating the PO from the process. The order of the POs in the PPA prefix calculation, shown in Fig. 1(a)-(d), is dependent on the kind of PPA. AxPPA's 'W' really stands for the weight of the words typed in. The number of approximated bits is indicated by the parameter K in AxPPA. The other portions indeed coincide with W K. In truth, K stands in for the approximation LSBs, while the PPA structure is precisely applied to the remaining MSBs.

For example, the AxPOs in Fig. 2(c) may be used to approximatively represent the four least important bits in an adder with sixteen input bits and K = 4. In fact, it makes use of any exact adder architecture and the remaining 12 Most Significant Bits. AxPPA-BK, AxPPA-KS, AxPPA-LF, and AxPPA-SK are the four designs we've built while working on AxPO proposals for four different PPAs. In any case, in 8-bit form, Verily, behold Fig. 2, which doth present a generic example of binary approximate addition, wherein the noble K hath a value of 8.

Let us consider the decimal values of 3322210 and 11625410, and embark upon this mathematical journey. Verily, the worth of K, which is set at 8, doth find itself divided within a 16-bit AxPPA. This division doth occur in a precise manner, as depicted in Figure 2(a)–(c), wherein the 8-bit portion is separated. Furthermore, an approximate portion, also encompassing 8 bits, is present, as illustrated in Figure 2(b)–(d). Verily, this exemplar doth reveal that the sum of 3322210 and 11625410 doth approximate to a value equal to 21714210.

Figure 2(d) depicts a somewhat equivalent total to the example in Figure 2(b), where K is represented by 8 bits. See how Fig. 2(c)-(d) is broken down into its component parts: (a) preprocessing, (b) approximate prefix calculation, and (d) postprocessing. As can be seen in Fig. 2(b) and (c), there is just one XOR logic gate in the critical path of the preprocessing. Be warned that, as shown in Fig. 2(h), only wires are used in approximation prefix computation to connect the preprocessing stage's

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 9, September : 2023

create and propagate carry values with the processing stage's output.

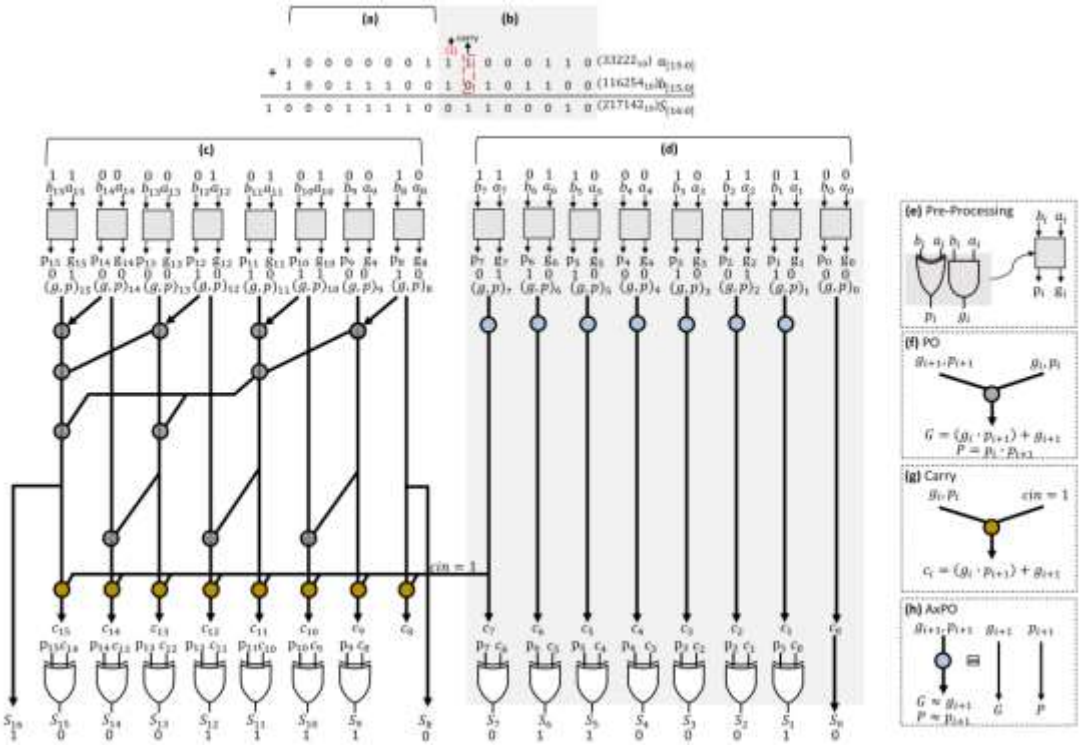

Fig. 2. Steps for the AxPPA-LF example: (a) MSB operation with exact adder, (b) LSB operation with AxPPA adder for K = 8 bits, (c) exact part with LF, (d) structure of AxPPA for K = 8 bits, (e) preprocessing step, (f) POs, (g) Carry, and (h) AxPOs.

There is indeed just one piece of carrying from the approximate section to the accurate part in Fig. 2(d). The POs are shown in all their dark yellow glory in Fig. 2(c), where they elaborate on how the AxPPA's generous carrying in PPA is calculated. The carry operator in Figure 2(g) really does have a critical path that's the same length as two logic gates.

One of these gates be an XOR, and the other, an AND. Verily, behold Fig. 2, which doth present an example, elucidating the steps for the AxPPA. Pray, direct thine gaze to Fig. 2(a) for further enlightenment. Pray, take heed of the light grey rectangles, for they doth signify the preprocessing steps, wherein the calculations of propagate and generate terms are made for each input-bit of the operands, a and b. As shown in Figure 2(b), the internal construction has a modest AND gate for generating and an XOR gate for propagation. As shown in Fig. 2(c), the AxPOs configure (7) and (8), eliminating one AND gate for propagate (P) and one OR and one AND logic gates in generate (G). The final sum, S, is calculated using XOR. This happens between the propagate term and the parallel prefix calculating carries. This is accurate, especially for parts S1–S7. Figure 2(d) demonstrates that S0 and S8 carry comparable amounts that p0 and g7 carry.

4.RESULTS

RTL SCHEMATIC: The register transfer level (RTL) diagram is the architectural plan, so to speak. It is used to check the architecture that has been

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 9, September : 2023

created versus the ideal architecture that is needed for development. Indeed, coding languages like verilog and vhdl are used in conjunction with the hdl language to transform the architecture's description or overview into a functional overview.

The RTL schematic really includes details about the internal connection blocks, allowing for more precise debugging. Verily, the image depicted below doth present the RTL schematic diagram of the crafted architecture.

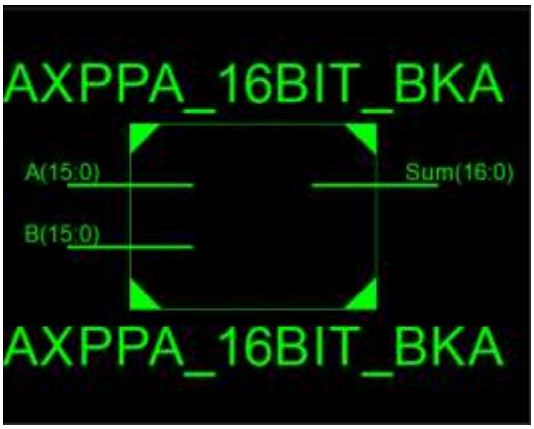

Fig .3. RTL Schematic of Approximate Brent-Kung Adder

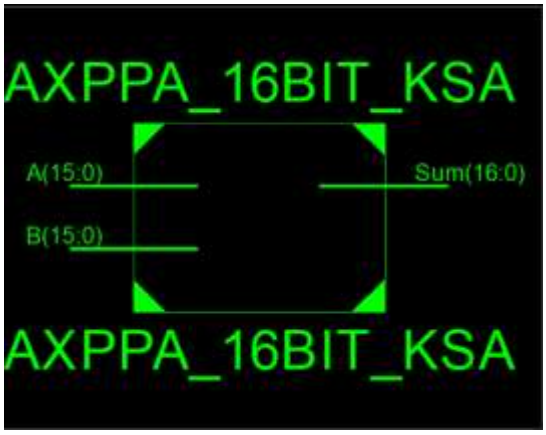

Fig .4. RTL Schematic of Approximate Kogge-Stone Adder

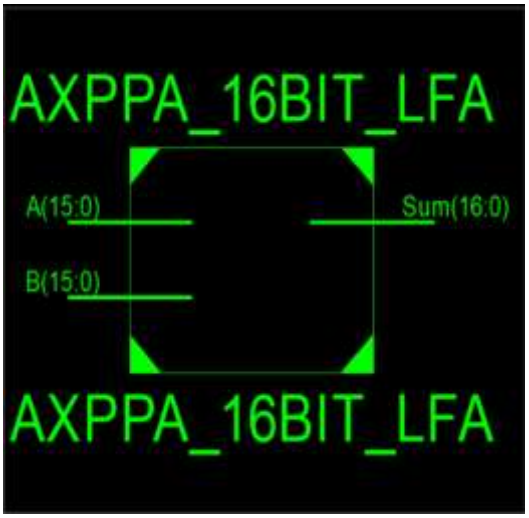

Fig .5. RTL Schematic of Approximate Ladner-Fischer Adder

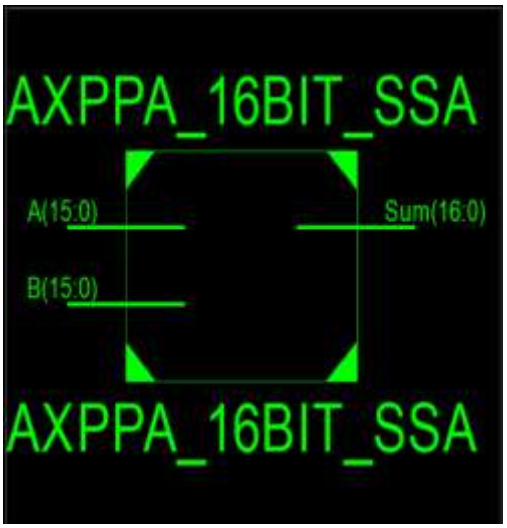

Fig .6. RTL Schematic of Approximate Sklansky Adder

TECHNOLOGY SCHEMATIC: Because the LUT format is used to describe the architecture in the technology schematic, it is considered the area parameter used in VLSI design estimation.Specifically, the LUTs in an FPGA represent the code's memory allocation, hence they are conceptualised as square units.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 9, September : 2023

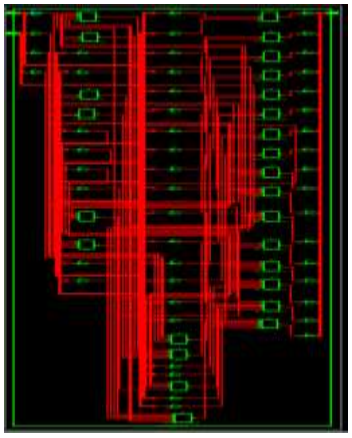

Fig .7. Technology Schematic of Approximate Brent-Kung Adder

Fig .8. Technology Schematic of Approximate Kogge-Stone Adder

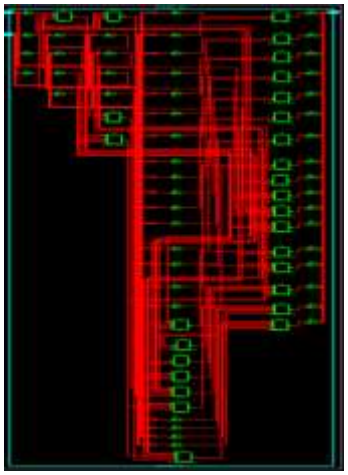

Fig .9. Technology Schematic of Approximate Ladner-Fischer Adder

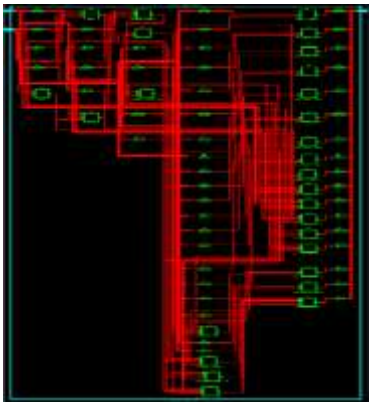

Fig .10. Technology Schematic of Approximate Sklansky Adder

PARAMETERS: Pray, in the realm of VLSI, the parameters at hand are area, delay, and power. By these measures, one may discern the superiority of one architecture over another. Hark! The consideration of the parameters doth be obtained by the usage of the tool XILINX 14.7, and the HDL language doth be the verilog language. Hark! Behold, a table of comparison for parameters doth I present unto thee.

PARAMETER	DELAY (ns)
APPRX_BKA	2.574
APPRX_KSA	3.9
APPRX_LFA	2.788
APPRX_SSA	3.414

Table .1. Comparison table of APX Adders

SIMULATION: Indeed, the procedure known as simulation is considered the last test of its functioning. Therefore, the diagram should be used to double-check all of the blocks and connections. The tool's main screen has changed from the implantation process to the simulation window. And behold, the wave formations are contained inside the simulation window. It really is quick enough to provide us a wide variety of radix number systems.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 9, September : 2023

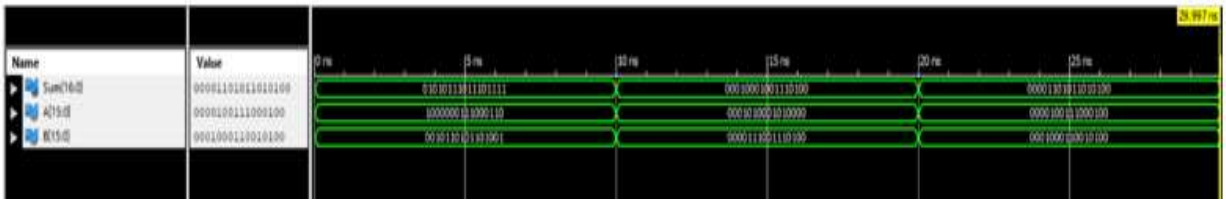

Fig .11. Simulated Waveform of Approximate Brent-Kung Adder

Fig .12. Simulated Waveform of Approximate Kogge-Stone Adder

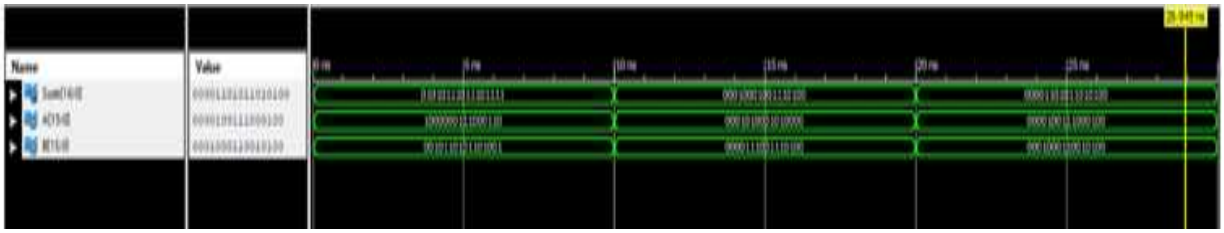

Fig .13. Simulated Waveform of Approximate Ladner-Fischer Adder

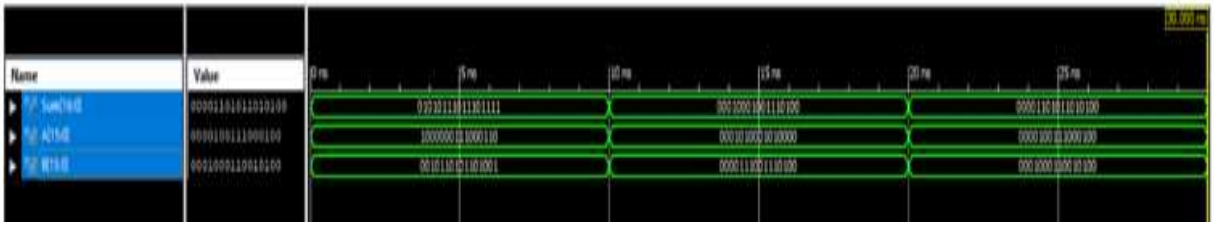

Fig .14. Simulated Waveform of Approximate Sklansky Adder

CONCLUSION

Verily, this article doth propose an AxPPA—a novel AxPPA architecture. Verily, the technique of approximation, once developed, doth calculate the POs of the step of computation that doth precede the prefix. Verily, Forsooth, we examined our AxPPA claims in the context of specific case studies and did it in a way that was free of application bias. Taking into account three highly regarded metrics (MAE, MRED), our modest idea triumphs over the energy-efficient AxAs of the literature in the arena of application-independent case studies.

Through the use of targeted use cases, we were able to demonstrate the better efficiency of our proposal with respect to circuit space and the fine balance of power and quality. Our invention accomplished this by combining a solid-state video accelerator and a FIR-filter signal processing

accelerator into a single package. Our AxPPA concept, in fact, has the greatest cost-cutting potential in syntheses. surpassing the consolidated AxAs found within the annals of literature, in both case studies. Verily, AxPPA doth attaineth standards of excellence in quality, and doth lendeth support to levels of approximation that are higher in nature.

REFERENCES

[1] M. Pashaeifar, M. Kamal, A. Kusha, and M. Pedram, "A theoretical framework for quality estimation and optimization of DSP applications using low-power approximate adders," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 66, no. 1, pp. 327–340, Jan. 2019.

[2] P. Stanley-Marbell et al., "Exploiting errors for efficiency: A survey from circuits to applications," ACM Comput. Surv., vol. 53, no. 3, pp. 1–39, Jun. 2020.

[3] H. B. Seidel, M. M. A. da Rosa, G. Paim, E. A. C. da Costa, S. J. M. Almeida, and S. Bampi, "Approximate pruned and truncated Haar discrete wavelet transform VLSI hardware for energy-efficient ECG signal processing," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 68, no. 5, pp. 1814–1826, May 2021.

[4] P. Pereira et al., "Energy-quality scalable design space exploration of approximate FFT hardware architectures," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 69, no. 11, pp. 4524–4534, Nov. 2022.

[5] G. Paim, L. M. G. Rocha, G. M. Santana, L. B. Soares, E. A. C. da Costa, and S. Bampi, "Power-, area-, and compressionefficient eight-point approximate 2-D discrete tchebichef transform hardware design combining truncation pruning and efficient transposition buffers," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 66, no. 2, pp. 680–693, Feb. 2019.

[6] L. Soares, J. Oliveira, E. Costa, and S. Bampi, "An energy-efficient and approximate accelerator design for real-time Canny edge detection," Circuits Syst. Signal Process., vol. 39, no. 12, pp. 6098–6120, 2020.

[7] G. Paim, H. Amrouch, E. A. C. D. Costa, S. Bampi, and J. Henkel, "Bridging the gap between voltage over-scaling and joint hardware accelerator-algorithm closed-loop," IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 1, pp. 398–410, Jan. 2022.

[8] Z. G. Tasoulas, G. Zervakis, I. Anagnostopoulos, H. Amrouch, and J. Henkel, "Weight-oriented approximation for energyefficient neural network inference accelerators," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 67, no. 12, pp. 4670–4683, Dec. 2020.

[9] B. Silveira et al., "Power-efficient sum of absolute differences hardware architecture using adder compressors for integer motion estimation design," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 64, no. 12, pp. 3126–3137, Aug. 2017.

[10] K. M. Reddy, M. H. Vasantha, Y. B. N. Kumar, and D. Dwivedi, "Design of approximate booth squarer for error-tolerant computing," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 28, no. 5, pp. 1230–1241, May 2020.

[11] M. M. A. da Rosa, G. Paim, R. I. S. J. Castro-Godínez, E. A. C. Costa, and S. and Bampi, "AxRSU: Approximate radix-4 squarer uni," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), Austin, TX, USA, Jun. 2022, pp. 1–4.