HYBRID APPROXIMATE ADDER BASED ON PARALLEL PREFIX AND CARRY SPECULATIVE DESIGNS

Priyanka. K*
Department of Electronics and Communication Engineering
KPR Institute of Engineering and Technology
Coimbatore, India

Dr. Jaikumar. R
Department of Electronics and Communication Engineering
KPR Institute of Engineering and Technology
Coimbatore, India

Abstract— **The application requires balance between accuracy and speed with high-performance approximate adders is presented in this work. Least significant part and most significant part is combined by this adder with exact computation which uses carry speculative technique. The delay and carry chain length is reduced by this proposed method. To improve accuracy the error detection and recovery mechanism. To four different parallel prefix adder architecture carry speculation is applied to determine the proposed system effectiveness. Based on the area, power, and delay the proposed approximate carry speculative parallel prefix adders (AxPPAs) are implemented in comparison with the existing approximate adders. The proposed system decreases the power, area and delay. The ModelSim is used as a simulation tool and Xilinx is used to find the effectiveness of proposed system with delay, area and power calculation.**

Keywords: accuracy, speed, carry speculative adders, delay, power, area, ModelSim, Xilinx, error detection, parallel prefix adders.

I INTRODUCTION

Approximate computing AxC is introduced in this paper it has inherent fault tolerance in many applications with high efficiency. In the design optimization AxC introduces the accuracy and the quality of the output. The area and energy consumption are reduced due to this approximate computing in VLSI circuits[7]. Adder units are very important for error resilient applications like image and video processing, machine learning and digital signal processing [6][11]. Arithmetic operations basic foundation is given by this units. The proposed Approximate parallel prefix adders AxPPAs provides the groundbreaking approach to the adder designs. The proposed design combines both the exact computation and approximate techniques for the higher performance and combines the four adders [12] [13] [14] [15]. To reduce the delay and make the system to perform faster carry speculation which predicts the carry behavior.

The paper is structured in detail as follows. Brief surveys of various existing methods are analyzed in Section II. Section III describes the proposed methodology. The evaluation results and its analysis are explained in section IV. Section V discusses about conclusion of the work.

II RELATED WORKS

P. Pereira et al (2022) presented a paper which uses approximate arithmetic to improve the energy efficiency of hardware for Fast Fourier Transforms (FFTs). For tasks like spectrogram generation different combinations of approximate multipliers and adders to find configurations that minimize the power consumption while maintaining acceptable output quality is explored in this paper [1].

M. M. A. da Rosa et al. (2022) presented a paper that introduces the AxRSU which is the unit for approximating squares in hardware accelerators. The applications like perfect precision are not need like video processing applications are targeted. The energy and are saving is achieved by AxRSU calculations by simplifying the encoding process. The balance between accuracy and efficiency are explored in this work In specific tasks power and size reduction is demonstrated by AxRSU [2].

G. Paim et al.(2022) presented a paper in which the hardware accelerators performance and energy efficiency is enhanced in this new strategy. Hardware accelerators with algorithm in a closed loop system combined with the

conventional voltage over scaling. Optimizing hardware accelerators are replaced by the innovative approach which doesn't rely on the voltage adjustments [3].

R. Roy et al. (2021) presented a paper which delves the designing of parallel prefix circuits a critical component in chips. To craft this circuit this approach uses the reinforcement learning a form of AI. Compared to conventional method this AI learns through trial and error resulting in circuits which is smaller and faster. By optimizing both performance and efficiency in future devices this innovation has the potential to revolutionize the chip [4].

K.L. Tsai et al. (2021) presented the paper that explained about the cost of the conventional method is high because of circuit size, power and processing speed to get accuracy. To balance the tradeoffs between efficiency and accuracy.Power gating is used by the accuracy customizable radix-4 adder.to adjust the internal logic by enabling or disabling the parts of the circuits selectively. For critical calculations ACRA switch is used to switch between high precision mode and low power approximation mode for the smaller number of tasks. It modifies the partial sum in approximate mode.
The balance between power consumption, accuracy and processing speed ACRA is superior to all other configurable adders. The image quality is maintained in high quality even though it operates in the half approximation mode. The SNR ratio is minimal drop [5].

J. Lee et al. (2021) presented the paper that discusses about the approach which is suitable for the minor errors which are tolerable and gives efficiency over exactness. The proposed design contains the carry prediction technique which reduces error compared to the conventional method. The errors in final result are minimized by the error reduction mechanism. By maintaining excellent hardware efficiency, it also achieves the higher accuracy. The tradeoffs between accuracy and efficiency are maintained sustainably by this approach. It chooses efficiency over accurate precision as the error has minimal effect on the practical applications [6].

K. **M. Reddy et al. (2020)** presented a paper which has the squarer units it is the crucial components in multiplication, for applications that can tolerate slight errors. Over absolute precision booth squarer prioritizes efficiency. Compared to conventional method there is a significant reduction in power consumption and processing times. The situation where the perfect results are not essential the trade-off accuracy makes it ideal such as image or audio processing. In error tolerant domains the squarer units make it faster and energy efficient computing [7].

Z. G. Tasoulas et al. (2020) presented a paper which gives the technique for Neural network with low energy consumption accelerators. Based on the weight importance the method of weight-oriented approximation which prioritizes specific calculations. Without retraining the entire network leading to significant energy savings by reduced precision in some calculations [8].

Masoud Pashaeifar et al. (2019) presented the paper that discusses about the quality of output signals in DSP is estimated by approximate adders. Additional noise source is introduced as an approximation noise. The mathematical model is developed to describethe approximation noise's power. Across the bandwidth of interest, the The spectral density error is computed.The predicted signal is compared with SNR i.e. noise ratio with respect to results from simulation. The average discrepancy is less than 2.5 dB to determine the model efficiency. Based on Lagrange multipliers mathematical optimization is done. Within each adder used in design parameters of the DSP blocks determine the approximate data width [9].

Omid Akbari et al. (2016) presented a paper which explains about the approximate look ahead adders that are used to switch between two ways of operation: approximation mode and accurate mode. The wider range of application uses this approach because of its flexibility, for the application requires high accuracy with lower power consumption and an error tolerance for faster speed. By modifying conventional Carry Look-Ahead Adder (CLA) RAP-CLA achieves this efficiency.Compared with conventional method RAP-CLA provides lower area and power consumption. RAP-CLA is evaluated using 15nm FinFET technology which achieves the delay and power reduction in 32-bit RAP-CLA in comparison with the exact CLA. RAP-CLA is used in real worlds application like image processing like sharpening and smoothing of images [10].

N. Zhu et al. (2010) presented a paper which explains about the transistors size which is miniaturized in the VLSI technology which leverages the concept of Error tolerance with the Error Tolerant Adders to improve the performance in digital signal processing.
By decreasing the small degree of accuracy Error Tolerant Adders prioritize speed and power consumption. The significant reduction in both power and execution time is done with ignoring the stringent accuracy in DSP systems. Compared to conventional methods of adders ETA gives the improvement of 65% of Power Delay Product (PDP) [11].

P. Brent and H.T. Kung (1982) presented a paper that carry out the approximate 16-bit binary numbers are added.
By sacrificing some accuracy, it achieves faster speed and lower power consumption. Like machine learning or signal processing it is used for the ideal applications where exact addition isn't critical. Compared to regular brent-kung function it uses fewer logic gates and simpler functions. Truncation, rounding or custom circuits which are the approximation techniques. It is suitable for energy efficient applications with smaller footprint and lower power [12].

R. Ladner and M. Fischer (1980) presented a paper, in 16-bit additions it trades accuracy for speed and area. For parallel processing Ladner-Fischer architecture. Higher order bits or ignored or approximating. It is crucial for the applications like machine learning which needs speed. Ignoring top 4 it uses 4-bit adders for lower 12 bits. Especially in most significant bits it introduces error in final results [13].

P.M. Kogge and H.S Stone (1973) presented a paper which trades accuracy for the speed and area accuracy. For applications like machine learning it prioritizes speed and performs approximate addition on 16-bit numbers. In the calculation it achieves this by ignoring or approximating lower order bits. The accuracy is reduced due to it but it makes the circuit smaller and faster [14].

J. Sklansky (1960) presented a paper that adds one bit at a time of binary numbers. Due to simple logic, it is easy to understand. It is faster for basic needs in single clock cycle. It is not ideal for the high-speed task it is limited by serial processing. It is useful for low-power applications and learning [15].

III PROPOSED METHODOLOGY

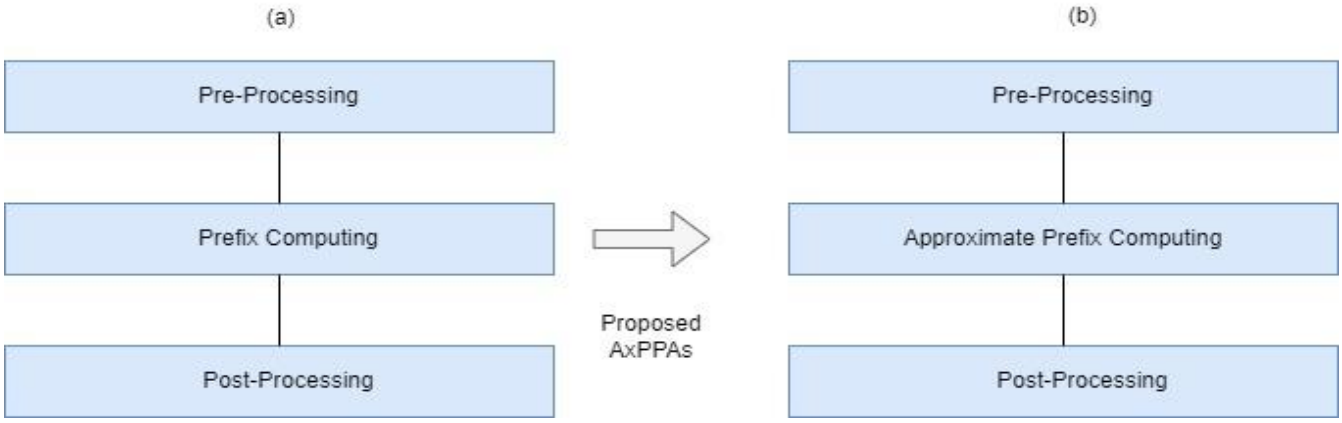

Fig 3.1 (a) Conventional PPA (b) Proposed AxPPA

Fig 3.1 shows conventional PPA and proposed AxPPA. PPA are separated into three blocks, namely pre-processing,prefix computing, and post processing. whereas the proposed system consists of pre-processing, approximate prefix computing and post-computing.

A. Pre-Processing

This stage contains bitwise with Create a carry g, then spread a carry p. Inputs A and B are converted into g and p.

B. Approximate prefix computing

Which contains the pre-processing, intermediate block approximated in it which contains set of Pos. It does the logic part in approximation modePos .

C. Post-Processing

Final output is generated by the Post processing by the computation of preprocessing p.

TABLE I GATE AREA AND UNIT DELAY FOR DIFFERENT TYPES OF ADDERS

ADDERS TYPES	UNIT DELAY	GATE AREA
Brent-Kung	$2\log_2(W)-2$	$2W-2-\log_2(W)$
Kogge-Stone	$\log_2(W)$	$W\log_2(W)-(W-1)$
Ladner-Fischer	$\log_2(W)$	$W/2\log_2(W)$
Sklansky	$2\log_2(W)-1$	$2W-2-\log_2(W)$

Where W is bit width. And table I shows the different address types and the there unit delay and gate area.

TABLE II TRUTH TABLE

INPUT				Generate		Propagate	
g[i+1]	g[i]	p[i+1]	p[i]	PO	AxPO	PO	AxPO
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1
0	0	1	1	0	0	1	1
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	1
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	1	1	1	1
1	1	0	0	1	1	0	0
1	1	0	1	1	1	0	0
1	1	1	0	1	1	0	1
1	1	1	1	1	1	1	1

The error of the AxPO proposal is highlighted with red color

$-G=(g[i]\ .\ p[i+1])+g[i+1]$, $-G\approx g[i+1]$, $-P=p[i+1]\ .\ p[i]$, $-P\approx p[i+1]$

Table II shows the truth table which compares the parallel prefix adders of PO and AxPO. Prefix sums of input arrays are computed with parallel prefix adder circuit. For various input values the table shows the output of both the PO and AxPO. For each input the right column shows the propagate P and generate G signals. To compute the prefix sums these signals are used. The highlighted contents show the error in AxPO Incorrect propagate and generate signals are proposed by this table.

It consists of 1-bit sub adders and carry predictor unit and select unit. 1-bit sub-adder performs addition operation depending on the inputs. Carry predictor unit it anticipates the carry signals where the current block is generated based on the input bits. Select Unit the critical decisions are made carry predicted by the carry predictor unit and actual carry which is generated by sub-adder is chooses by this unit.

Enhanced Carry Prediction of Speculative Approximate Adder

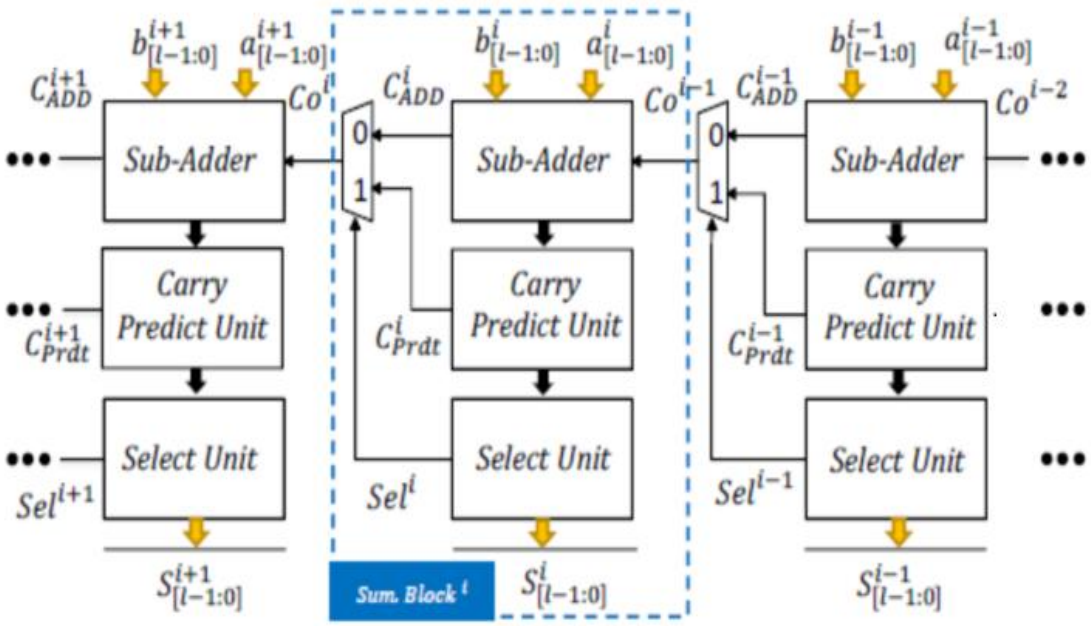

Fig 3.2. Enhanced Carry Prediction of Speculative Approximate Adder

Fig 3.2. shows the enhanced carry prediction of speculative approximate adder. For improved performance carry speculative adder architecture utilizes a carry predictor unit. The adder operates by dividing the n-bit operation into parallel l-bit summation blocks (where l is the block size). Integration of a subadder, select unit, and carry predictor unit is done in each block.

The carry input for each input in sub-adder is chooses is the key innovation here. Conventionally the previous block's carry is considered. However, this design proposes a speculative approach where the carry input for the Based on, the i-th block is established.The combination of input signals from both the current block (i-th) and the subsequent block (i+1st).

This speculative approach, as will be demonstrated later, offers significant accuracy improvements compared to existing approximate adders. While conventional designs like BCSA rely on previous block information for carry prediction, this new method leverages additional data, leading to more accurate results.

$$CO^i = \overline{Sel}^l . C^i_{ADD} + . C^i_{Prdt} \text{ ---------- (1)}$$

$$Sel^i = K^{i+1}_0 + G^i_{l-1} \text{ ----------(2)}$$

$$C^i_{Prdt} = G^i_{l-1} \text{ ---------- (3)}$$

$$C^i_{ADD} = P^i_{l-1}G^i_{l-2} + P^i_{l-1}P^i_{l-2}G^i_{l-3} + \cdots\cdots + \prod_{k-1}^{l-1} P^i_k . G^i_0 \text{------(4)}$$

Where K is first bit position of the kill bit. Equation 1 combines the previous stage propagated carry potential The current input bits are used to generate carry.Carry generation based on kill bits and the previous generate signals are identified by the equation 2. Previous generated carry is assigned to the C^{i}_{Prdt} is reprensted by equation 3. Based on the previous stage the carry is propogated and the based on the generate signals the carry is generated to those stage is represnted in the euation 4.
Fig 3.3 shows the proposed adder with Error Recovery Unit (ERU) which is used to minimize the errors.

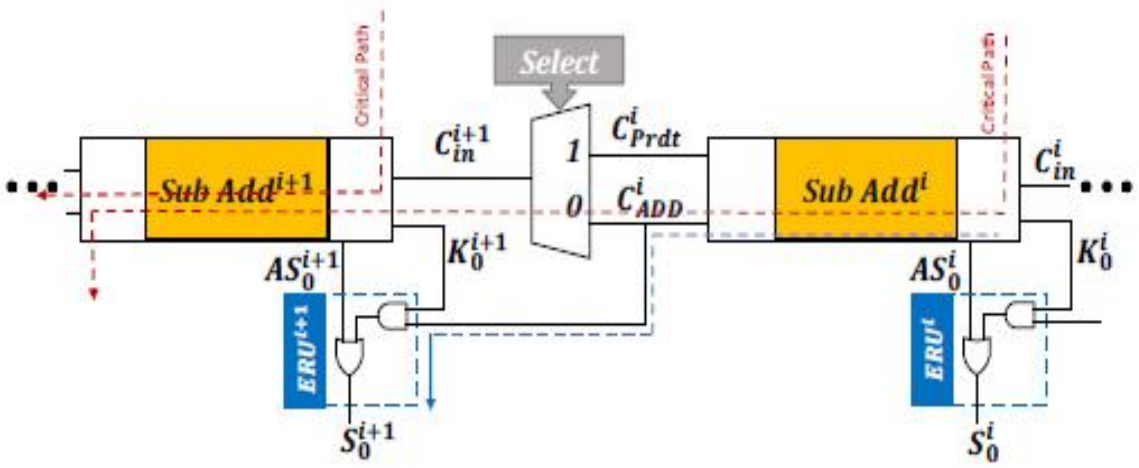

Fig 3.3 Proposed Adder with Error Recovery Unit (ERU)

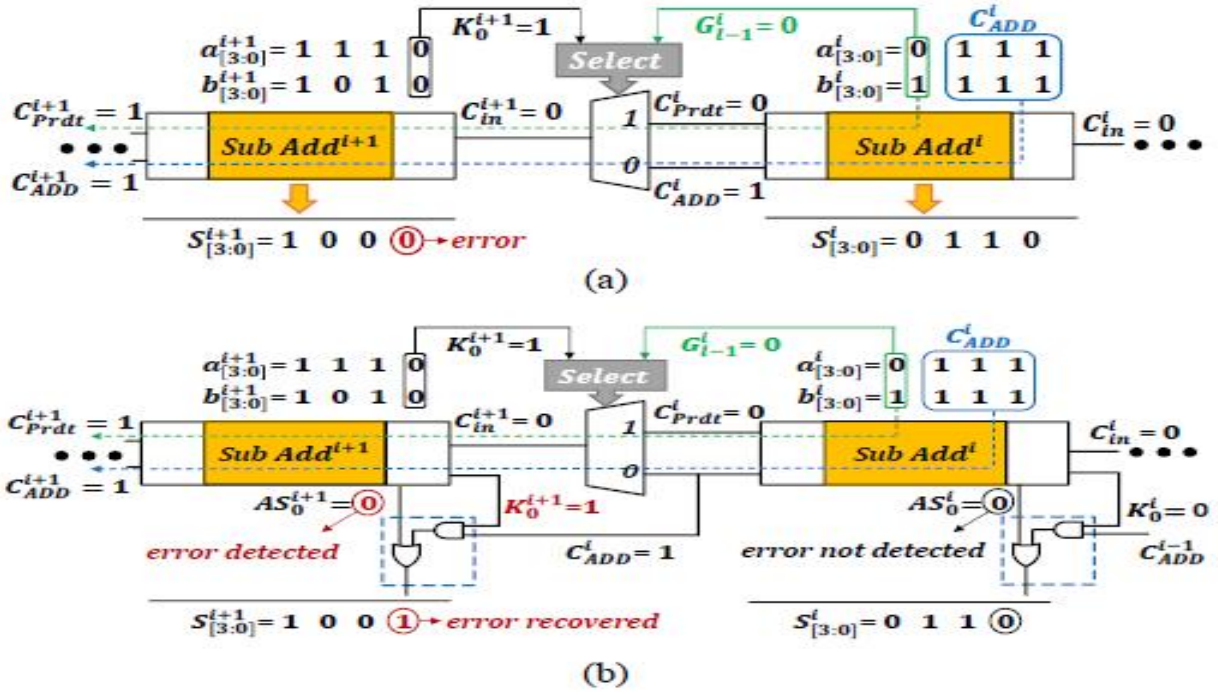

Fig 3.4. Existing Adder vs. Proposed Design Example Functionality a) without ERU design b) with ERU design

Fig 3.4. shows the Existing Adder vs. Proposed Design Example Functionality
a) without ERU design b) with ERU design which illustrates the functionality of proposed adder with Error Recovery Unit (ERU) error recover mechanism is lacked in existing system of without ERU which is overcome by the with ERU which identifies and potentially correcting errors in the calculated sum, leading to

improved accuracy compared to the design without error recovery.

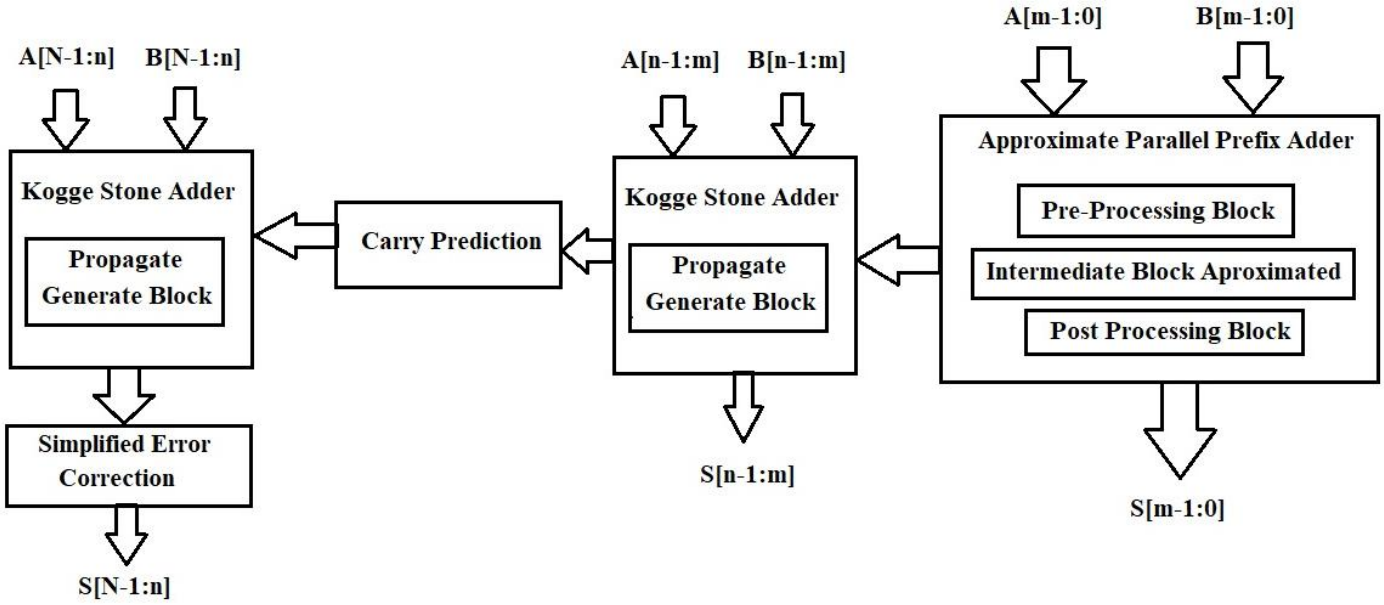

Fig. 3.5. Proposed Block Diagram of Hybrid Kogge-Stone based Carry Speculative Adder

Fig. 3.5. depicts the proposed block diagram of hybrid Kogge-Stone based carry speculative adder. This adder is designed for high-speed accurate arithmetic operation. It combines both carry propagation methods and approximation techniques. The adders are divided into two sections. They are Less Significant part and Most Significant part. Less Significant Part employs the approximation technique at the initial stage of processing. Two operands A and B are transformed into two signals propagate P and generate G. It determines whether a bit creates a carry or spreads a carry from the previous bit.

Most Significant Part employs high accuracy by the higher order bits. To generate an approximate sum the G and P signals are directly fed into the sum block from the LSP. Speculative carry propagation is used by MSP for high accuracy.

Speculative carry propagation estimates each block's carry output, depending on its internal operands and subsequent carry propagation i+1 which reduces the overall delay and this calculated carry is used as an input for the next block by this carry chain is efficiently shortened.

To minimize the errors introduced during approximation stage in LSP error correction module is used this design integrates simplified error correction. The discrepancies arise during final output is detected and rectified using this error correction module. The proposed system provides the faster speed due to initial processing, high accuracy because the carry propagation maintains accurate calculations and reduced complexity as the design of conventional method is simplified. The error tolerant adders which increases power and energy consumption.

From the block diagram shown above:

- A[N-1:0] and B[N-1:0] are added two binary numbers N is the total number in this case.
- Alm-1:0] and Blm-1:0] the least important m components of A and B.
- The addition performed in MSP is done by Kogge stone adder.
- Kogge stone adder uses propagate block and generate block to calculate carry bits.
- The inputs given to the Kogge stone adder is prepared using pre-processing block and post-processing block and these are used to correct the errors in the output.
- S[n-1:m] and Sim-1:0]: Sum of two numbers i.e. S[n-1:m] is sum of least significant bits m and Sim-1:0] is the sum of most significant bits.

W-Bit PPA Pseudo Code -Algorithm 1

```
1.  Input: A, B, Cin, W
2.  Output: S
3.  for i=1 to W do
4.  p(1,i) ← A(i) XOR B(i)
5.  g(1,i) ← A(i) AND B(i)
6.  end for
7.  y ← 1
8.  for i=2 to (Delay calculation in Table II) do
9.  for j=2 + y to W +-1 1 do
10. P (i,j) ← p(i-1, j) AND p(i-1,j-y)
11. G(i,j) ← p(i-1,j) AND (g(i-1,j-y) OR g(i-1,j))
12. end for
13. y ← 2^(j-1)
14. end for
15. C(0) ← Cin
16. for i=1 to W+1 do
17. C(i) ← Cin AND P(W+1,i) OR G(W+1,i)
18. S(i) ← C(i-1) XOR p(1,i)
19. end for
```

W-Bit Input and K-bit AXPPA Pseudo Code-Algorithm 2

```
1.  Input: A, B, Cin, W, K
2.  Output: S
3.  if K > 0 then {Approximate part}
4.  for i=0 to K-1 do
5.  p(1,i) ← A(i) XOR B(i)
6.  g(1,i) ← A(i) AND B(i)
7.  end for
8.  S(0) ← p(0)
9.  for i=1 to K-2 do
10. S(1,i) ← p(1,i) XOR g(1,i-1)
11. end for
12. carry ← g(K-1)
13. end for
14. if (W-K) > 0 then {exact part}
15. for i=k to W-1 do
16. p(1,i) ← A(i) XOR B(i)
17. g(1,i) ← A(i) AND B(i)
18. end for
19. y ← 1
20. for i= (K+1) to (Delay calculation in table II)
21. for j=(K+y+1) to W+1 do
22. P(i,j) ← p(i-1,j) AND p(i-1,j-y)
23. g(i,j) ← p(i-1,j) AND (g(i-1,j-y) OR g(i-1,j))
24. end for
25. y ← 2^(j-1)
26. end for
27. Cin ← carry
28. for i=(K+1) to W+1 do
29. c(i) ← Cin AND P(W+1,i) OR G(W+1,I)
30. S(i) ← c(i-1) XOR P(W+1,i)
31. end for
32. end if
```

IV RESULTS AND ANALYSIS

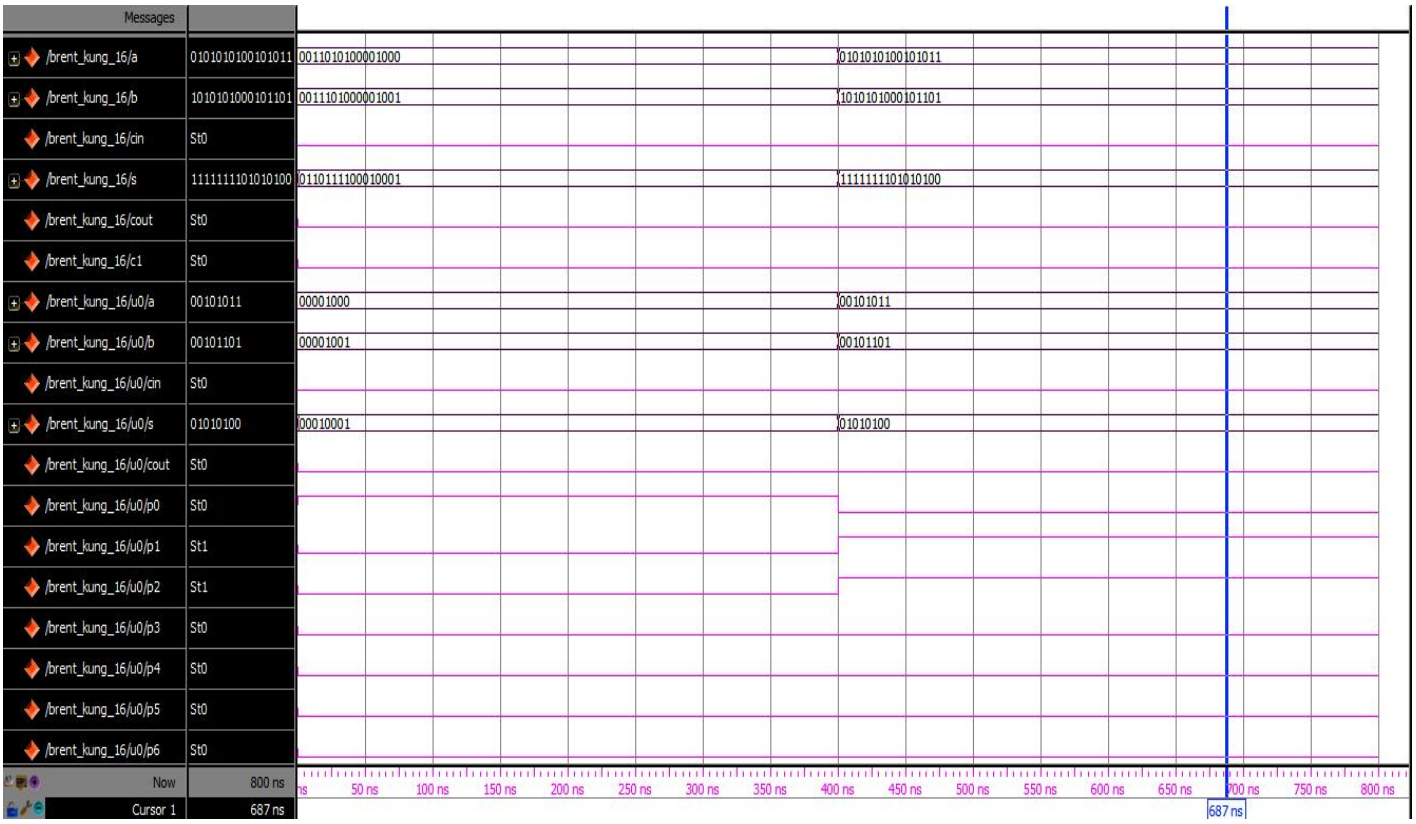

Fig 4.1.The 16-bit approximation simulation result Brent-Kung adder

Fig 4.1. displays the 16-bit approximation simulation result Brent-Kung adder. Inputs A and B of approximate Brent-Kung adder with carry input CIN and CR as output carry. LSB addition is performed through exact parallel prefix Brent-Kung operation. This approach reduces design cost. . The value of input A and B are A=0101010100101011 and B=1010101000101101 and Sum S=1111111101010100

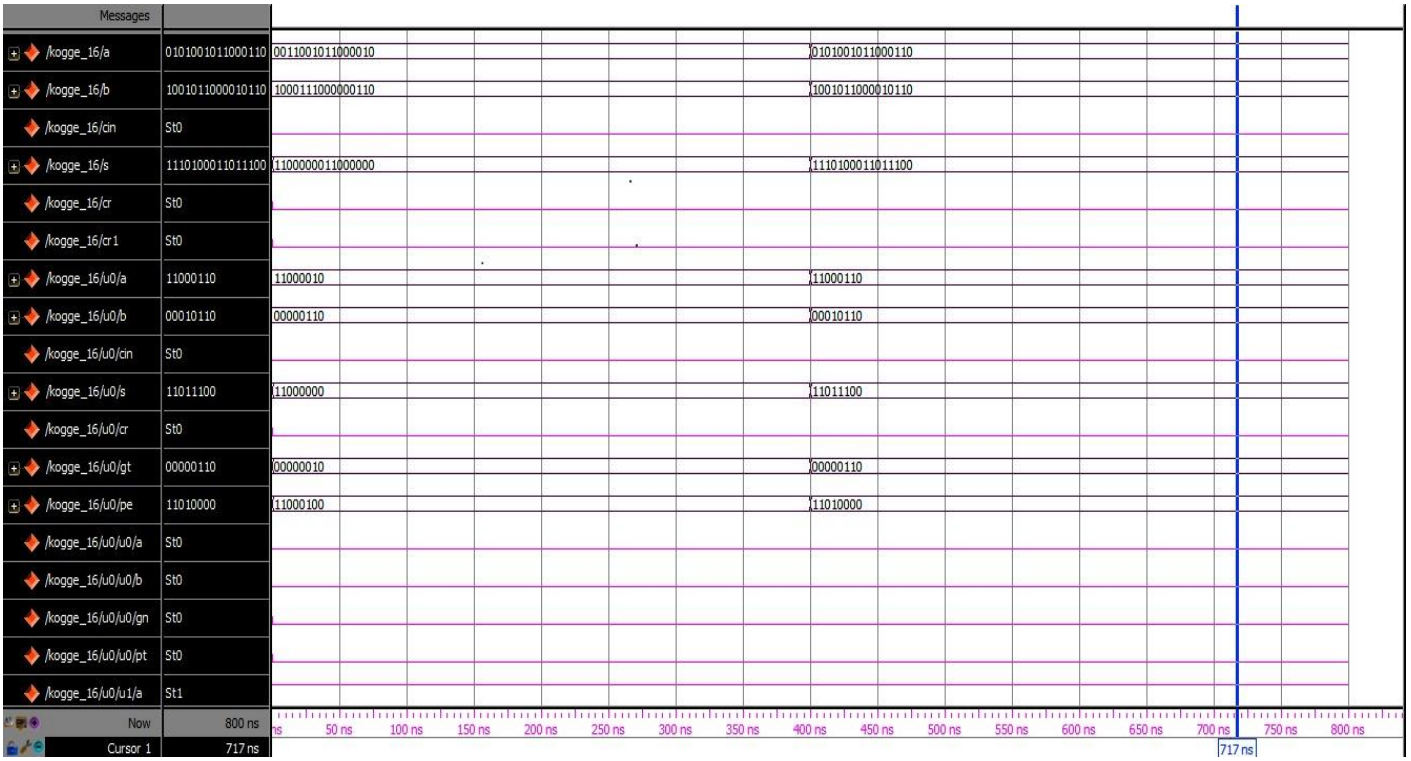

Fig 4.2The 16-bit approximation simulation result Kogge-stone adder

Fig 4.2. displays the 16-bit approximation simulation result Kogge-stone adder. Inputs A and B of approximate Brent-Kung adder with carry input CIN, Sum output is S and CR as output carry . LSB addition is performed through approximation and MSB addition is performed through exact parallel prefix Kogge-stone operation. It improves efficiency but the area is increased due to routing. The value of input A and B are A=0101001011000110 and B=1001011000010110 and Sum S=1110100011011100.

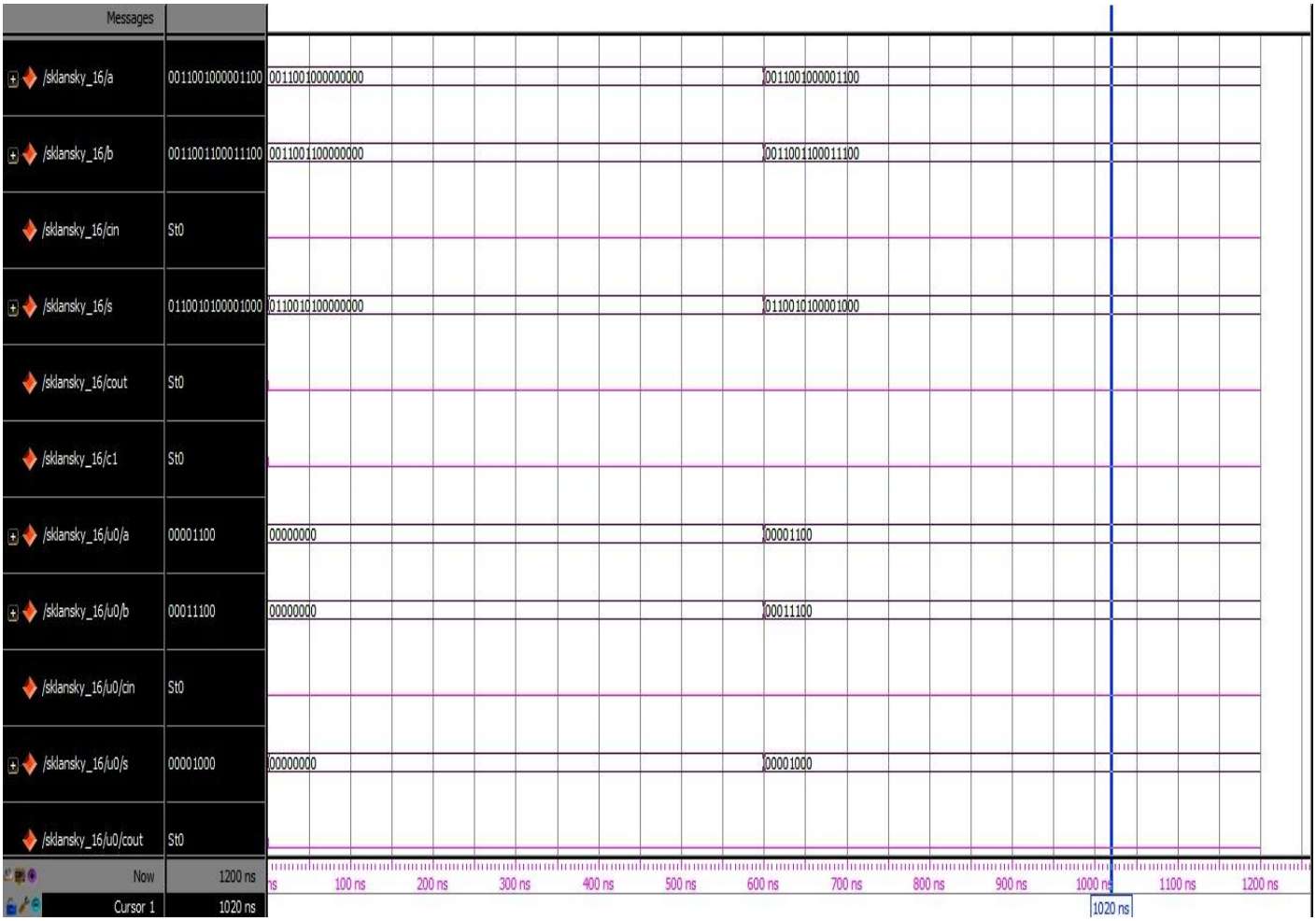

Fig 4.3.The 16-bit approximation simulation result Sklansky adder

Fig 4.3. displays the 16-bit approximation simulation result Sklansky adder. A and B is the input operands of approximate Sklansky adder with CIN as the carry input. S is the sum output with COUT as the carry output. LSB addition is performed through approximation and MSB addition is performed through exact parallel prefix Sklansky operation. This approach reduces delay but the cost is doubled and the performance is degraded due to it. The value of input A and B are A=0011001000001100 and B=0011001100011100 and Sum S=0110010100001000.

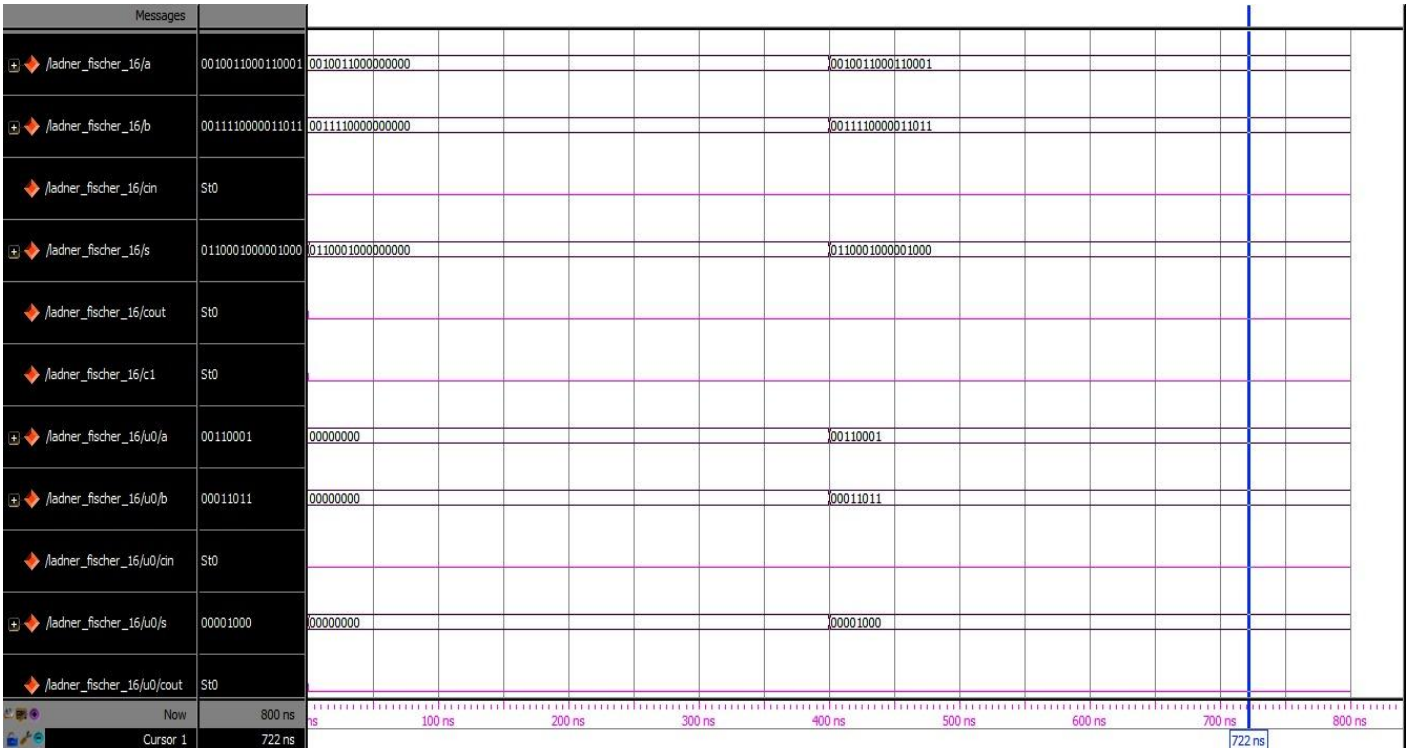

Fig 4.4. The 16-bit approximation simulation result Ladner-Fischer adder

Fig 4.4. displays the 16-bit approximation simulation result Ladner-Fischer adder. Inputs A and B of approximate Brent-Kung adder with carry input CIN, S sum output and COUT as output carry. LSB addition is performed through approximation and MSB addition is performed through exact parallel prefix Ladner-Fischer operation. The value of input A and B are A=0010011000110001 and B=0011110000011011 and Sum S=0110001000001000.

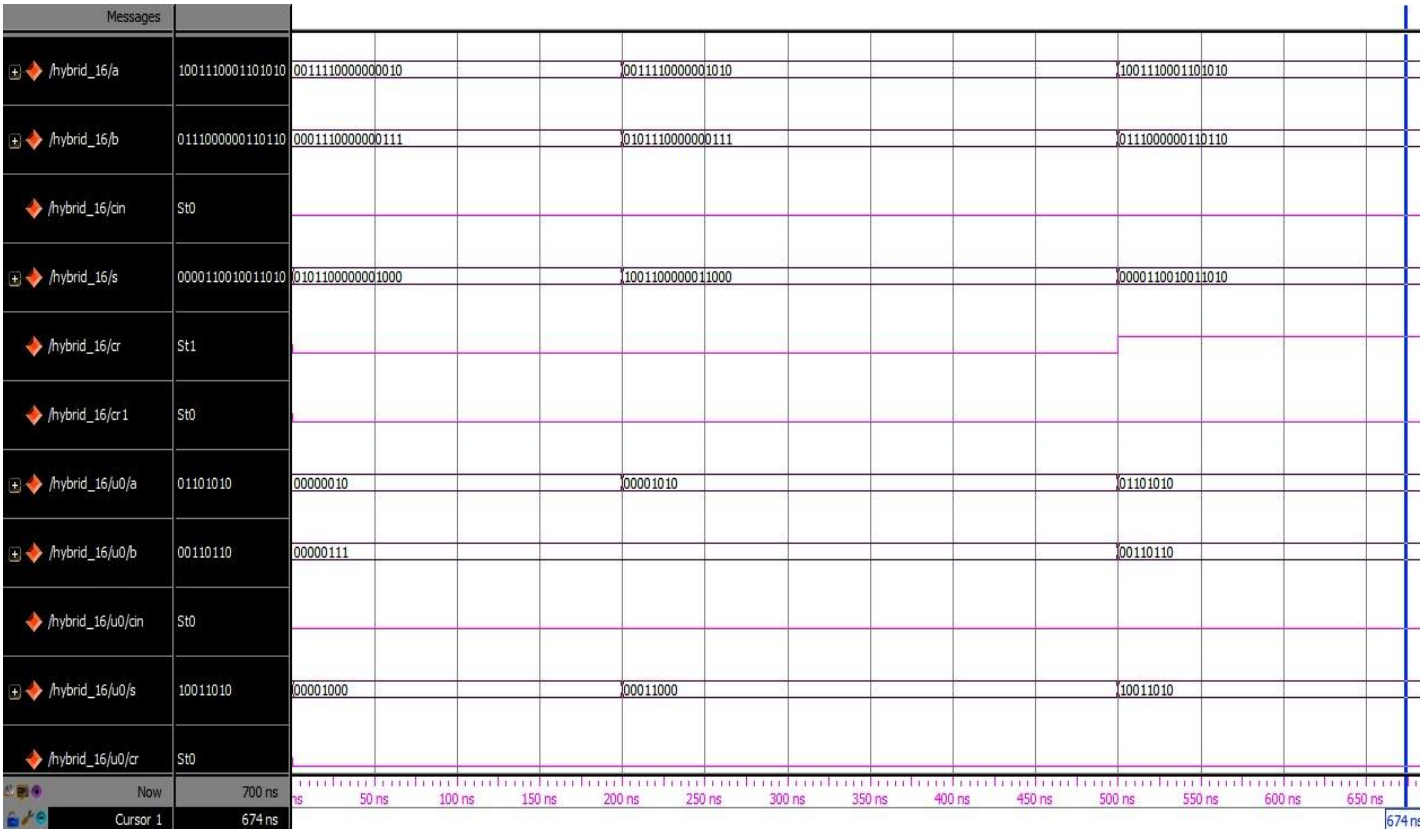

Fig 4.5. The 16-bit approximation simulation result Hybrid Kogge-stone Carry speculative adder

Fig 4.5.displays the 16-bit approximation simulation result Hybrid Kogge-stone Carry speculative adder. The value of input A and B are A=1001110001101010 and B=0111000000110110 and Sum S=0000110010011010 Inputs A and B of approximate Hybrid Kogge-stone Carry speculative adder with carry input CIN, S sum output and COUT as output carry. LSB addition is performed through approximation and MSB addition is performed through combination of carry speculative logic and parallel prefix Kogge-stone operation. Efficiency and fan-out reduction are achieved by the Kogge-Stone. There are generation and propagation cells in parallel in the tree. However, because there are more route options when the adder arrangement is arranged in a regular grid, the circuit area grows.The 8-bit MSB is divided into two blocks. The carry from first block is speculated and given to the second block. The Kogge Stone adder is employed for the sub-adders. In the suggested approach, we concentrate on minimizing the error by choosing an appropriate carry input on the subsequent block. Consequently, in each of these cases, the carry output is selected with the highest accuracy. The carry output of the i th block is determined under four cases where, for each of them, either or is selected.

TABLE III COMPARISON TABLE

Types	Area (Gate count)	LUTs	Slices	Delay (ns)	Power (mW)
Brent-kung	216	36	22	16.059	231.56
Kogge-stone	138	23	14	16.347	183.86
Ladner-Fischer	216	36	22	16.059	231.56
Sklansky	168	28	17	17.665	203.76
Hybrid-KS-BCSA	114	19	12	16.266	177.39

Table III shows different adder types and their area gate count which is 114 in proposed method which is less than all other existing adder type likewise the LUTs is 19 which is less for proposed hybrid -KS-BCSA adders than all other adders. The slices are just 12 for the Hybrid-KS-BCSA adders. The delay and power is 16.266ns and 177.39 milli watt than any other adders like Brent-kung, Kogge-stone, Ladner-Fischer and Sklansky.

V CONCLUSIONS

A new approximate hybrid adder is proposed and compared with various parallel prefix adders.The POs of the prefix computation phase are calculated using the established approximation technique. The AxPPA proposals are made for certain case studies and are not dependant on any particular application.Block based carry speculative logic is applied to MSP to minimize area and power consumption. Considering various metrices the proposed method provides the higher efficiency. In contrast to the usual method the the capability consumption and energy consumption is more in the proposed hybrid architecture. The proposed adder results in 17.4% resource optimization and 3.5% power reduction compared with Kogge stone adder, which is in an average better of other types of approximate parallel prefix adders.

REFERENCES

[1] 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.
[2] 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.
[3] 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.

[4] R. Roy et al., "PrefixRL: Optimization of parallel prefix circuits using deep reinforcement learning," in Proc. 58th ACM/IEEE Design Automat. Conf. (DAC), Dec. 2021, pp. 853–858.

[5] K.-L. Tsai, Y.-J. Chang, C.-H. Wang, and C.-T. Chiang, "Accuracyconfigurable Radix-4 adder with a dynamic output modification scheme," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 68, no. 8, pp. 3328–3336, Aug. 2021.

[6] J. Lee, H. Seo, H. Seok, and Y. Kim, "A novel approximate adder design using error reduced carry prediction and constant truncation," IEEE Access, vol. 9, pp. 119939–119953, 2021.

[7] 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.

[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] 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.

[10] Omid Akibari et al. "Improved Efficiency of Reconfigurable Approximate Look-Ahead Adder" IEEE Transactions on Circuits and Systems II: Express Briefs PP(99):1-1 DOI:10.1109/TCSII.2016.2633307. Nov. 2016.

[11] N. Zhu, W. L. Goh, W. Zhang, K. S. Yeo, and Z. H. Kong, "Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 8, pp. 1225–1229, Aug. 2010.

[12] R. P. Brent and H. T. Kung, "A regular layout for parallel adders," IEEE Trans. Comput., vol. C-31, no. 3, pp. 260–264, Mar. 1982.

[13] R. Ladner and M. Fischer, "Parallel prefix computation," J. ACM, vol. 27, pp. 831–838, Oct. 1980.

[14] P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," IEEE Trans. Comput., vol. C-22, no. 8, pp. 786–793, Aug. 1973.

[15] J. Sklansky, "Conditional-sum addition logic," IEEE Trans. Electron. Comput., vol. EC-9, no. 2, pp. 226–231, Jun. 1960.