DESIGN AND ANALYSIS OF ARITHMETIC UNITS

A Thesis submitted in fulfillment of the requirement for the Award of the Degree of

MASTER OF TECHNOLOGY

In VLSI Design

Submitted By

YASHODA BISHT

601762021

Under Supervision of

Dr. Bharat Garg

Assistant Professor

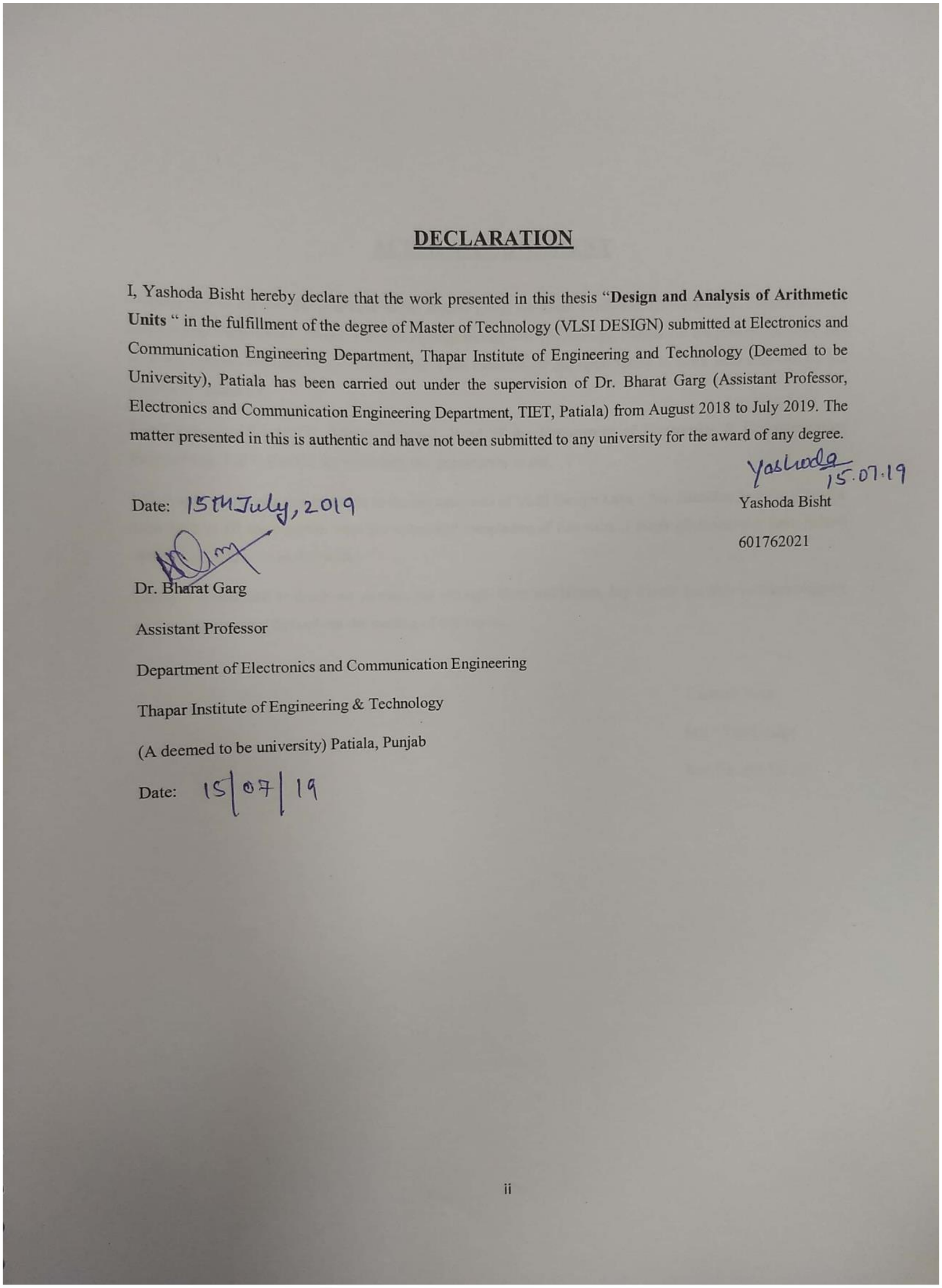
THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY

(DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB

JULY 2019



DECLARATION

I, Yashoda Bisht hereby declare that the work presented in this thesis "**Design and Analysis of Arithmetic Units** " in the fulfillment of the degree of Master of Technology (VLSI DESIGN) submitted at Electronics and Communication Engineering Department, Thapar Institute of Engineering and Technology (Deemed to be University), Patiala has been carried out under the supervision of Dr. Bharat Garg (Assistant Professor, Electronics and Communication Engineering Department, TIET, Patiala) from August 2018 to July 2019. The matter presented in this is authentic and have not been submitted to any university for the award of any degree.

Yashoda 15.07.19

Yashoda Bisht

601762021

Date: 15th July, 2019

Dr. Bharat Garg
Assistant Professor
Department of Electronics and Communication Engineering
Thapar Institute of Engineering & Technology
(A deemed to be university) Patiala, Punjab

Date: 15/07/19

ACKNOWLEDGEMENT

I would like to express my gratitude to Dr. Bharat Garg, Assistant Professor, Electronics & Communication Engineering Department, Thapar Institute of Engineering and Technology, Patiala for his patient guidance and constant support throughout this thesis work. I am thankful for the opportunity to work with him and for his invaluable guidance.

I am also thankful to Dr. Alpana Agarwal, Head of the Department of Electronics & Communication Engineering, TIET, Patiala, for providing this opportunity to me.

I also want to express my gratitude to the lab assistants of VLSI Design Labs – Mr. Manohar and Mr. Ajay for their help in all the possible ways for successful completion of this work. I thank all those who have helped directly or indirectly in this work.

Lastly, I would like to thank my parents, my siblings- Hina and Hitesh. My friends for their constant support and encouragement throughout the making of this report.

Yashoda Bisht

ME-VLSI Design

Roll No: 601762021

ABSTRACT

With the advancement in technology, several applications like the mobile computing, social media, scientific research involving complex computations, data transferring etc., are widely used. This requires a large amount of resources for storage and transmission that calls for a number of modern devices and data centers. The modern devices exhibit a tradeoff of power and performance. The novel architectures/designs improve one parameter at the cost of other and fail to improve both the parameters simultaneously. Certain applications such as multimedia, data mining, recognition etc. exhibit error tolerance/resiliency due to non-existence of unique output, redundancy in input data and probability computations. In these applications small amount of error is tolerable and outputs are completely acceptable. For these applications, approximate circuits/design can be developed.

Significant research has been done to achieve approximate algorithm/architecture and circuits. Various approximate architectures are presented to reduce power dissipation, area and improved delay. Along with the development of approximate architecture for DSP cores (filters), several approximate arithmetic units are presented. Among the arithmetic unit, adder being the prime component of any digital circuit therefore, several approximate adder architectures are presented. Approximate adders are widely used and under experimentations nowadays. These approximate adders have been classified into the approximate and accuracy-reconfigurable adders. This report covers the design and analysis of the approximate adder designs.

In this report, a simplified reverse carry-propagate adder (SRP) where the carry propagates from most significant bit (MSB) to least significant bit (LSB) is proposed. Further different structures have been proposed depending on the propagating carry. The efficacy of the proposed SRP adders has been calculated on comparison with the state-of-the-art reverse carry adder (RCPA). The proposed SRP has been implemented in Verilog and synthesized with the Synopsys design compiler using 65nm library. The results indicate that for eight-bit numbers, at a constant delay of 0.17 n sec, **SRP-FA-II** is providing an improved power-area-delay product of **46%** over RCPFA-III, about **85.6%** improvement over RCA and **96.6%** improvement over ACA. For 16-bit numbers, SRCP-FA-II is providing an improved power-area-delay product of **53.73%** over RCPFA-III, **70.6%** improvement over RCA and about **91.7%** improvement over ACA. Finally, for comparative analysis of quality metrics, the Lena image (256x256) has been smoothed by Gaussian smoothing filters (GSF) embedded with the proposed SRP and existing 16-bit approximate adders. The simulation results demonstrate that proposed adder outperforms over the existing designs.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

The VLSI industry has experienced a number of improvements in terms of technology with time. According to Moore's law, the number of transistors on a microchip would double in two years but the cost would reduce to half. In order to increase the number of transistors and simultaneously reducing the chip area, the problem of large power dissipation would arise. Different tradeoffs of power consumption and performance provide various design solution. Both power dissipation and delay are the prime concern for the modern devices. The novel architectures/designs improve one parameter at the cost of other and fail to improve both the parameters simultaneously. There are several applications such as- multimedia [1], data mining [2], recognition etc. that exhibit error tolerance/resiliency due to non-existence of unique output, redundancy in input data and probabilistic computations [3], [4], [5], small amount of error is tolerated and outputs are completely acceptable in these applications [6]. For these applications, power and performance can be simultaneously improved by trading accuracy i.e. accuracy is considered as new tradeoff parameter to improve design metrics. Approximate computing [7][8][9] is a paradigm to achieve this trade-off and the applications that support it are called the error tolerant applications. This chapter first explores the error tolerant applications, followed by the approximate cores and the arithmetic units like multipliers and adders.

1.1 ERROR TOLERANT APPLICATIONS

Applications where the accuracy can be traded off with improved power dissipation are called the *error tolerant applications*. Applications like video decoder, video processing etc., where the user can compromise on the accuracy. These interact with human senses and are tolerant to slight inaccuracies, for example the phone lines do not carry the sound perfectly but errors are tolerable enough for human use. Machine learning, digital signal processing applications, audio/video learning, are few other applications where small errors are acceptable.

These applications can tolerate some errors and the output is still useful and thus the focus can be given to other improving other factors like the power consumption, improving the delay and reducing the area.

1.2 APPROXIMATE CORES

There are different DSP cores where approximate computing can be employed. Examples are the DCT cores [10][11][12][13] (widely used for image processing), FIR filters, Gaussian-cores that can be employed to improve the performance by the trade-off of accuracy with power-dissipation, area or delay.

1.3 ARITHMETIC UNIT

Different components like the adders and the multipliers are discussed under the arithmetic unit. Adders being the heart of any circuit can also replace multipliers in several circuits.

1.3.1 Multiplier Architectures

Multiplier is an essential component in any digital signal processing system. It consumes most of the power in any digital circuit [14]. Thus, if the speed can be improved and the power consumption could be reduced, a good design can be achieved. There are different types of multipliers namely, Array multiplier, Wallace multiplier, Bypassing multiplier, Vedic Multiplier. An Array multiplier performs the multiplication operation of two operands using the shift and add operation, being regular but its powers consumption was very high. The Wallace multiplier does the fast-parallel scheming but the sequential adding stages have been reduced in order to reduce the partial product accumulation. In Bypassing Multiplier, ay row/column is made disabled if the corresponding bit of the multiplier /multiplicand is zero, this has greatly reduced the power consumption. A new algorithm to multiply two signed/unsigned numbers was proposed by the Booth multiplier. Another multiplier namely the Vedic multiplier was proposed that was based on the Urdhva Tiryakbhyam Sutra and is found to be more efficient in terms of power and speed in comparison with state-of-the-art multipliers. Multipliers [15][16] can also be used for approximate computing where the accuracy can be traded-off with improves power dissipation [17][18].

1.3.2 Adders Architectures

Adders are said to be the heart of any circuit because a number of adders are found in the processors. So any reduction in the power consumption, reduction in area or delay, such improvements in the design can effectively improve the performance of the adders.

A lot of research has been carried out in the field of adders but now-a-days, approximate adders have been widely used. Approximate adders [19][20] are also the adders but here the improvements are made whether in the power consumption [21], in reducing the area or in delay at the cost of accuracy. These can be used in such applications like the audio/video [22] signal transmissions, DSP applications where the accuracy can be compromised. The adders can be further classified a segmented and the unsegmented adders.

The literation has been covered in the next chapter.

CHAPTER-2

LITERATURE REVIEW

As discussed in the previous chapter, approximate computing [23] is a paradigm to improve the power dissipation at the cost of accuracy. Several applications like multimedia processing, data mining, recognition etc. that can tolerate some errors and improve other factors like power dissipation, delay etc. are the error tolerant applications. Adders being the primary components of any digital circuit are to be given more importance over the multipliers. Different state-of-the-art adders have been discussed in this chapter and their performances have been covered in literature.

Adders are said to be the core of any digital circuit owing to low cost and area. These are under a lot of research now a days and thus adders are most extensively explored in the approximate computing. The adders are classified as: approximate and accuracy-configurable adders. i. Approximate adders are those where the outputs are not exact i.e. they are approximated by using certain logics. ii. Accuracy-configurable adders are those where accuracy can be configured at run-time. Section I, this starts with discussing the approximate adders, for example like the Error Tolerant Adder (ETAI) [24] and its modified versions like ETAIIM. ETAI divides the operands into accurate- the most significant bits (MSB's) and inaccurate parts-the least significant bits (LSB's). For the inaccurate/approximate part, all the successive bits in the sum are set to '0' if both the preceding bits (operands) are '1'. Normal addition operation takes place for the accurate/exact part. Another modified adder ETAII [25] was presented where the carry propagation path was reduced by taking two components-one sum and another carry generator with the carry flowing concurrently. But in order to further improve the accuracy ETAIIM [26] [27] was developed in which the first three carry generators were cascaded, this gave more area overhead than ETAI.

Sometimes we need 100% results and these have to be reconfigured at runtime so accuracy reconfigurable adders came into existence. In next paragraphs a number of accuracy reconfigurable adders namely a few like the accuracy-configurable adder (ACA) [28], gracefully degrading adder (GDA) [29], Generic accuracy configurable adder (GeAr) [30], Correlation Aware Predictor (CAP) [31], Accurus [32], optimized lower part constant-OR adder [33], Variable Latency Speculative Adder [34] ,signed-unsigned adder [35], Reconfigurable Approximate Carry Look-Ahead Adder (RAP-CLA) [36] etc. are explored.

2.1 APPROXIMATE ADDERS

Approximate adders are those where the outputs are not exact i.e. they are approximated by using certain logics. For example, if adding two 16-bit numbers, the addends would be segmented into two blocks. The two blocks would be of 8-bit each and then the operation would take place, this improves the critical path. A number of approximate adders are mentioned below:

2.1.1 ETAI

In ETAI, carry is not generated or take any carry signal to eliminate the carry-propagation path, this may induce some errors and to minimize those errors, Zhu *et al*. have used a strategy, suppose taking two 16-bit numbers and dividing them into Accurate and Inaccurate parts.: i. The Accurate part consists of the MSB, solved from right to Left and normal operation takes place. ii. The Inaccurate part consists of the LSB where the operation takes place from Left to Right and normal bit operation is performed when both I/p bits are'0' or different and when both I/p bits are '1', the checking process gets stopped and from this bit onwards, all the sum bits to the right are set to '1' where we take the two 16-bit operands,

A = "1011001110011010" and B = "0110100100010011" and the final result is "10001110010011111".

In this way overall delay time is reduced and power consumption also gets reduced but accuracy is degraded. This is useful in DSP applications such as image processing and speech processing where accuracy is not a very much important concern unlike in control systems. ETA I has a problem of calculating small number inputs as it yields a result that is far from correct results so a modified version, ETA II was proposed.

2.1.2 ETAII

In ETAII, the entire carry propagation path is split into a number of short paths and the carry propagates concurrently. For an N-bit adder, if the number of blocks are M then the number of bits =N/M. There are two circuitries-carry and sum generators. The carry generator creates the carry out signal but does not take the previous carry while the Sum Generator takes the previous carry that is the carry goes from previous carry generator to the next sum generator, thus carry propagates only between two neighboring blocks. The block diagram of ETAII is shown in Figure 2.1 So for N=4, which is taken as an ideal condition where P (probability of getting correct carry signal) = 0.96875 and the delay would also be less as compared to N=12. The power dissipation has been reduced but still the accuracy is degraded for large operands so a modified version has been introduced, modified ETAII (ETAIIM).

Table 2.1 Probability of getting correct carry with different number of bit positions

No. of bit considered	**1bit**	**2bits**	**4bits**	**8bits**	**12bits**
Probability of getting correct carry signal	0.75	0.875	0.96875	0.99805	0.99988

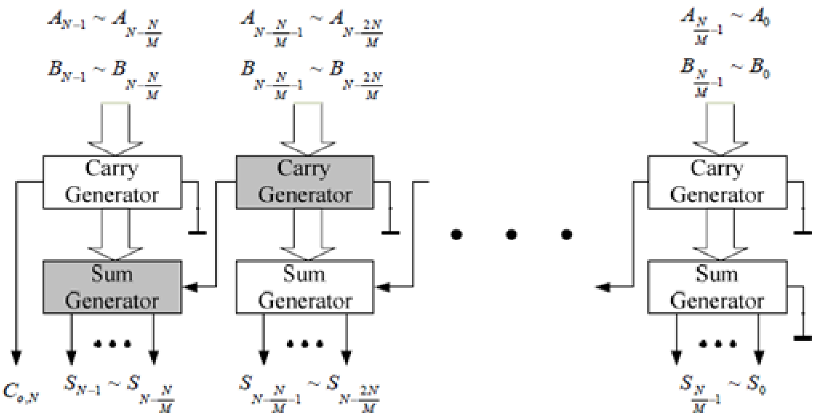


Figure 2.1 Block diagram of Type II ETA(ETAII).

2.1.3 Modified error tolerant adder (ETAIIM)

In ETAIIM, the first three carry generators are cascaded and generate the carry signals for the two highest blocks. The structure of ETAIIM is shown in Figure 2.2. It has been observed that PDP has gradually reduced (same as ETA) but its area has increased. ETAII/ETAIIM have outperformed other adders like RCA, CSK [39], CLA [] in terms of power consumption and speed performance. But still ETAIIM applications are strictly confined to DSP applications where accuracy can be compromised.

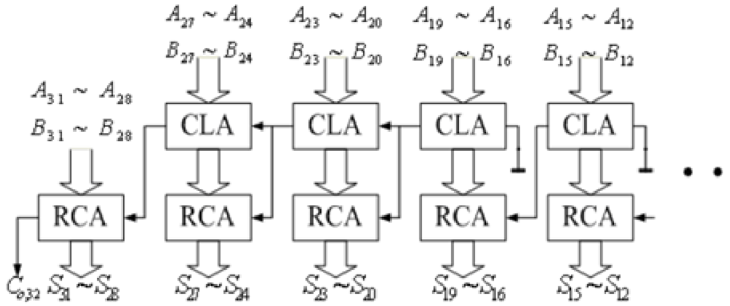


Figure 2.2 Structure of Modified ETAII (ETAIIM).

However, various applications require varying accuracies and sometimes 100% accuracy is also required like in image processing example in security cameras, clear images are required to track the motion. So a model is required where results could be reconfigured at runtime according to our accuracy requirement in both accurate and inaccurate mode. So accuracy reconfigurable adders were proposed and different kinds of such adders have been discussed below.

2.2 ACCURACY RECONFIGURABLE ADDER

Sometimes we need 100% accuracy and accuracy requirements may vary according to applications so we need to have such an adder in which we can configure the result at the run time. Previous adders have faced difficulty in detecting and correcting error so several reconfigurable adders were proposed. So another technique named “An accuracy configurable approximate adder (ACA)” has been proposed by A.B. Kahng and S. Kang [28] to provide configuration during runtime. This adder can achieve significant throughput improvement. This ACA can provide with different accuracies in different modes, like in accuracy configurable adder (ACA), there is an error detecting and error correcting circuit that has been introduced. Sometimes the quality of the output may be affected during runtime, which can be somehow reduced using gracefully degrading adder (GDA).

Also the number of sub adders were fixed in an ACA, so another adder generic accuracy reconfigurable adder (GeAr), was introduced provides a high number of reconfigurable and when considering the number of corrections proposed with these adders, still the corrections may not be up to the mark as the processing starts from the least significant bits (LSB) so another adder, Accuracy reconfigurable approximate adder (Accurus) was proposed, where the convergence takes from most significant bit (MSB) to LSB. Now detailed explanation has been provided below.

2.2.1 Accuracy configurable adder

The power benefits of accuracy configurable adder have been shown in Figure 2.3 which explains that in different modes: accurate and approximate modes, there is different normalized power at different accuracies while at 100% accuracy the power consumption is highest.

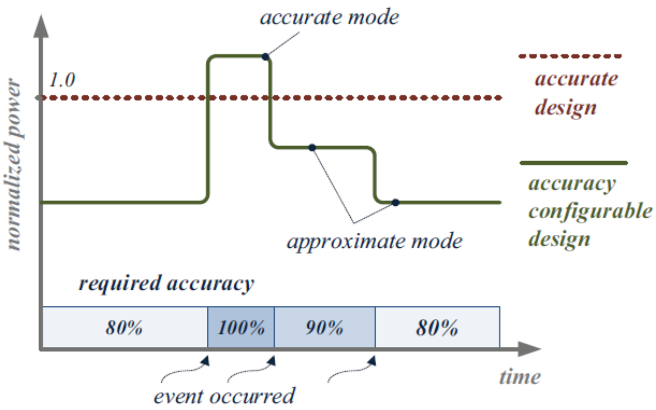


Figure 2.3 Power benefits from accuracy configurable design [28].

The central contribution of ACA is to provide with features like error correction and accuracy. Figure 2.4 shows the proposed adder for a 16-bit add, here taken are two operands: A[15:0] and B[15:0] and this 16-bit adder is sub-divided into three-sub-adders: SUM_H having A_H+B_H, SUM_M having A_M+B_M ,SUM_L having A_L+B_L, of length 2k, where k is the bit width, here k=4 and length of sub adder (2k) is 8bits, these adders would give the partial summations.

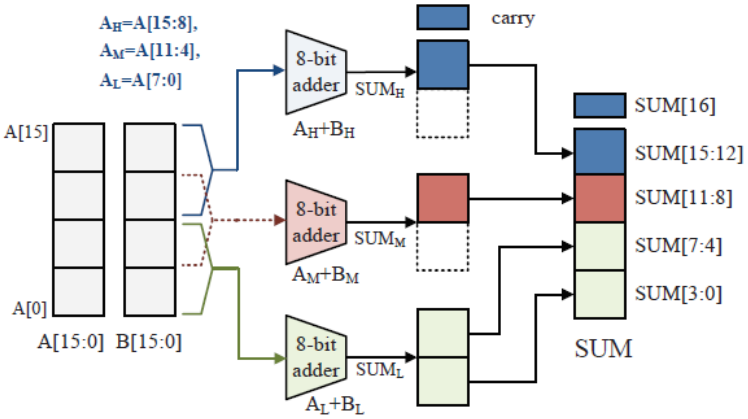


Figure 2.4 Approximate adder 16-bit adder [28].

In the result, each sub-divided module produces a k bit result except for the last sub adder that gives a 2k bit result. This is done in order to divide the carry chains so that the delay could be reduced. This reduced carry chains would reduce the critical path delay and the performance is improved with high operating frequency and low-power consumption as the applied operating voltage is now low. Figure 2.5 shows the general implementation of the ACA adder which consists of (N/k)-1 sub adders

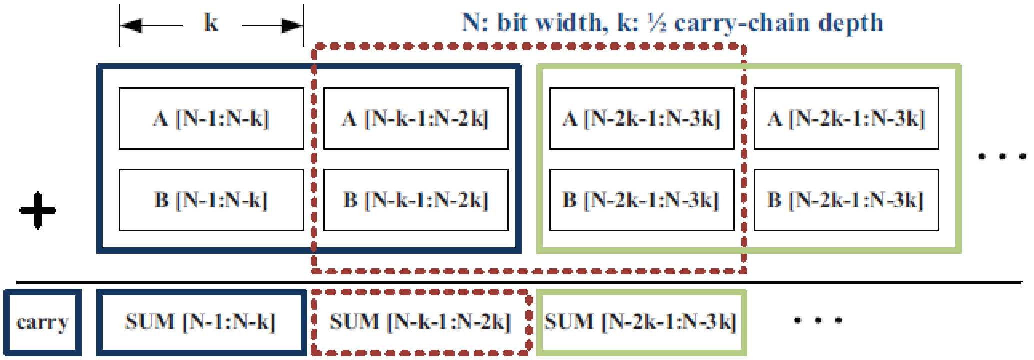


Figure 2.5 General implementation for the proposed adder [28].

In ACA adder, the output can be incorrect except for the last sub adder when a carry is being propagated because it has been considered that no carry is being propagated from LSB to MSB. When the carry [4] bit from A_L+B_L is ‘1’ and SUM_M [1111], the output has an error in SUM [11:8]. The output will be correct when there is no error in three sub adders. So in order to remove this error: An Error Detection and Correction (EDC) circuit is introduced. The probability of having a correct result is:

$$P (N, k) = (1-((1/2) (1/2)^{k})) \tag{2.1}$$

Figure 2.6 shows an error detection and correction circuit, here the detection is done by the AND gates and for the correction, an ‘incrementor’ circuit has been introduced. Here when SUM [1111] then the Detection circuit will generate a signal called error signal which is then corrected by the ‘incrementor’ by adding a ‘1’ to the output (to remove the carry).

The generated error signal will then go to the MUX which will choose the ‘$SUM_{correct}$’. When compared to the conventional CLA adder, this ACA more accuracy that is pass rate is 0.942 at k=4.

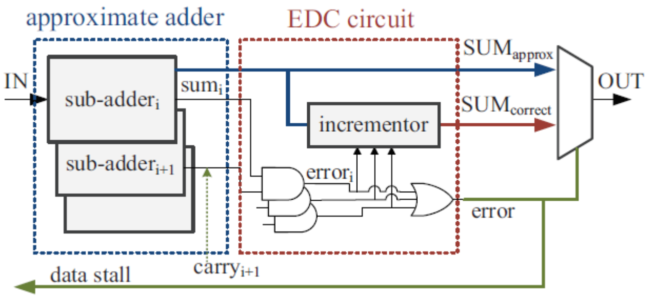


Figure 2.6 Error detection and correction with the approximate adder [28].

The power consumption by ACA, by using power gated circuit, power consumption has been reduced, explained in Figure 2.7.

Here by using power gated circuit, the later stages (whether to use the EDC circuit) can be turn off or not that is in several cases where EDC is not required, unnecessary power consumption by the EDC circuit can be avoided.

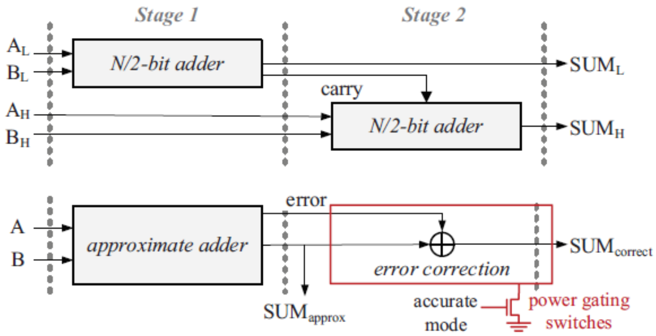


Figure 2.7 Pipelined adder implementation –conventional adder (above) and approximate adder (below).

In approximate operation, the error correction stage is power-gated. Although ACA circuit provides better accuracy and performance but consumes 11.5% more power than the conventional adder in accurate mode (mode-1) due to the EDC circuit but showing a considerable power reduction 12.4%, 31.0% and 51.6% on the modes: 2, 3 and 4, respectively.

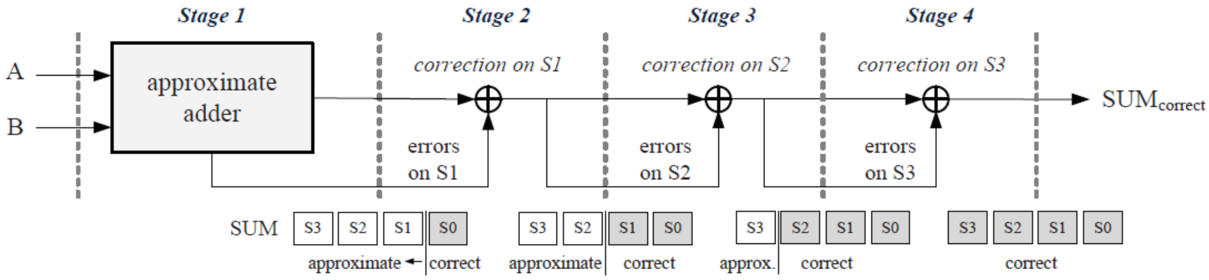


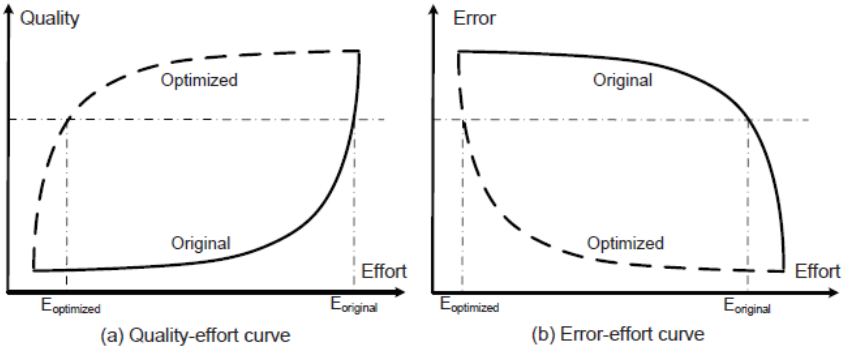
Figure 2.8 Accuracy-configurable implementation for pipelined adder.

ACA has fixed approximation quality that can be seen in Figure 2.8. For low accuracy, only Stage1 would be on and for 100%accuracy, all stages: Stage1, Stage2, Stage3, Stage4 should be on which requires much more effort.

2.2.2 Gracefully degrading adder (GDA)

This problem of ACA can be resolved by designing a quality-configurable-system (QCS), which R. Ye *et al*. [29] have presented in their paper “On Reconfiguration-Oriented Approximate Adder Design and Its Application”. It is preferable to design a QCS that would maintain a tradeoff of the computational quality and computational efforts according to the applications.

Here a "quality-effort" curve has been considered that would tell how much effort have to be put forward for the required quality. So considering a 'Quality-effort' curve in GDA as shown in Figure 2.9, if a dotted curve (optimized) is the output instead of the original curve, then it is said to be a good QCS. As shown in Figure 2.9 a, it would be best if same quality at minimum effort is received, also in Figure 2.9 b, at an optimized effort, the error is reducing gracefully.



(a) Quality-effort curve

(b) Error-effort curve

Figure 2.9 The proposed gracefully-degrading accuracy- configurable adder [29]

In Accuracy gracefully degrading Adder, two mechanisms: a. Reconfigurable Sub-Adder Bit-Length b. Reconfigurable Carry-in Prediction. In a. Reconfigurable Sub-Adder Bit-Length, they have taken two N-bit addends, A and B, taking A [3:0] and B [3:0] and these are joined by MUX, also the C_{in} can be chosen from the 'Carry-in Prediction' or by using the LSB unit.

So if the carry-in Prediction as the C_{in} (carry in) is taken for the adders, less delay would be there rather than taking C_{in} as the LSB unit, at the tradeoff of accuracy.

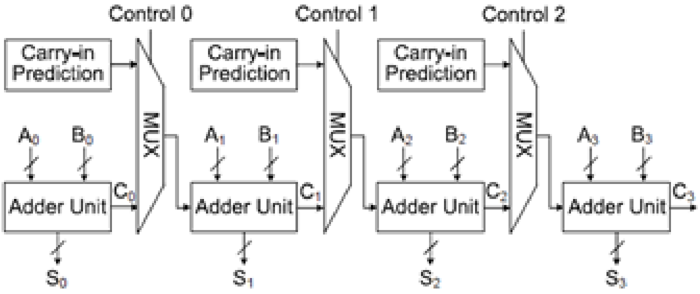


Figure 2.10 The proposed gracefully-degrading accuracy-configurable adder.

In this way by using the control signals, the adders can be combined in the same way as the sub adders have been made in ACA. This has been explained in Figure 2.10.

2.2.3 Generic accuracy configurable adder (GeAr)

In reconfigurable carry-in prediction, here the idea is to generate carry-ins from least significant bits of addends. Here the two N-bit addends are A = (A_{N-1}, ... A_i, A_0) and B = (B_{N-1}, ..., $B_{i,}$ B_0) where A_i and B_i are the i_{th} bits if A and B. The 'P' propagate, 'G' generate and 'C_{in}' are given by:

$$P_{i=}A_i\ B_{i,} \tag{2.1}$$

$$G_{i=}\ A_i\ B_i \tag{2.2}$$

$$C_{i+1}=G_i+P_iC_i \tag{2.3}$$

By adding several prediction logics and control logics, a prediction scheme can be designed that would add/turn off several prediction components trading off the power/delay and prediction accuracy. There is further scope in the improvement of QCS design. In (accuracy configurable adder) ACA, it has been recorded that the configuration of the sub adders were fixed and less flexibility, so in order to support variable approximation modes, M. Shafique *et al.*[30] proposed "A low latency generic accuracy configurable adder" (GeAr) that provides a higher number of configurations compared to state-of-the art and based on the requirement any sub adder can be selected for error correction. GeAr provides error recovery for adders without simulating them.

In GeAr model, N has been taken as the length of the operands, number of sub adders are denoted by 'k', number of bits of sub adders is 'L', R is the number of resultant bits and P is the number of previous bits used for carry prediction, each sub adder produces L-bits sum except the last sub adder that produces a sum of L bits (L=R+P). Number of sub adders (k) can be given by:

$$k=((N-L)/R)+1 \tag{2.4}$$

The generic form of GeAr adder is shown in Figure 2.11.

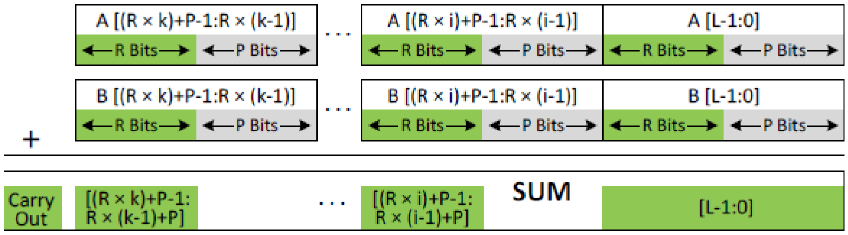


Figure 2.11 GeAr adder generic form: first sub adder, subsequent sub adders (right to left) [30].

Now taking different configuration of GeAr that is different value of R, P and k. It has been explained suing Figure 2.12 and Figure 2.13.

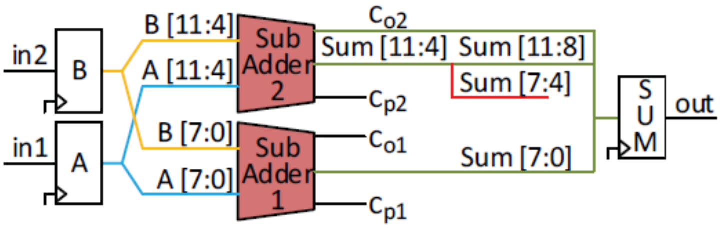


Figure 2.12 GeAr with N=12, R=4, P=4 and k=2.

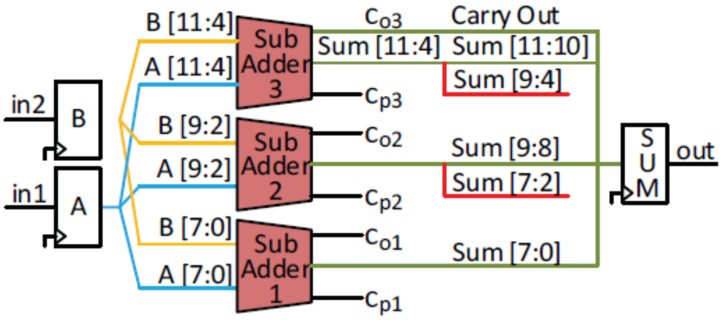


Figure 2.13 GeAr with N=12, R=2, P=6 and k=3.

It can be seen that on comparing the above two configurations, the GeAr in Figure 2.13, shows low delay, low area and accuracy is just average while on the contrast the configuration shown in Figure 2.14

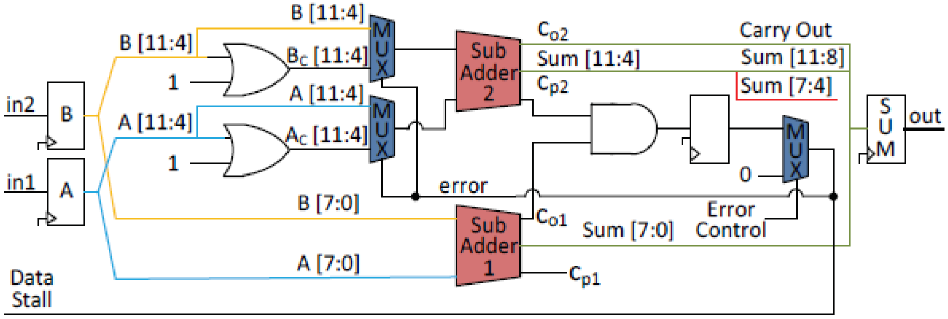


Figure 2.14 Error detection and correction for GeAr adder with N=12, R=4, P=4, k=2.

Figure 2.14 achieves average delay and area but high accuracy. Now coming to the Error and Detection circuit, which was already included in the ACA but here in the error correction step, as shown in Figure 2.15, where both the inputs of the i_{th} adders are passed through the OR gate and their LSB's are set to '1' and these LSB’s will generate the carry. The lower inputs are used in MUX, which will generate accurate results. There is an additional feature proposed in the correction circuitry that is the inclusion of the ‘error control’ select signal, which will select the sub adder that needs correction. So in this way any sub adder can be selected for error-correction, this has been explained in Figure 2.15.

In this way, GeAr provides a large number of selectable configurations compared to that of the state of the art adders. But still in GeAr latency is a big issue and to resolve this latency problem, another adder Correlation aware predictor (CAP) [31] was introduced.

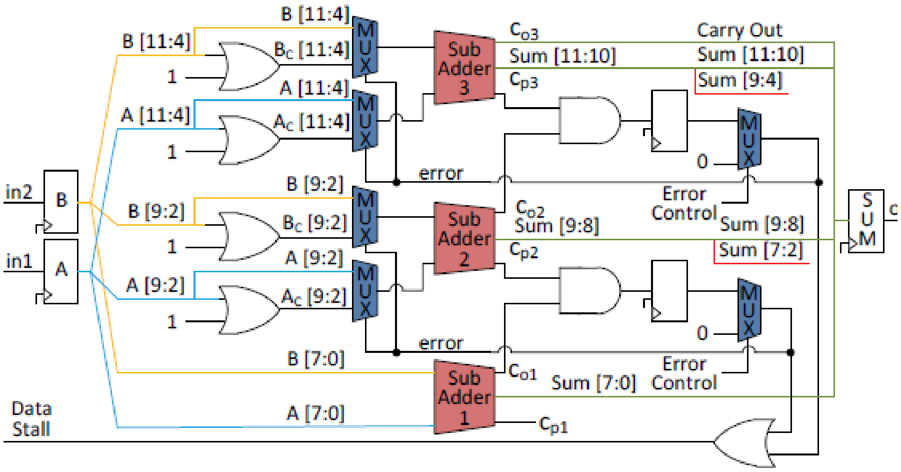


Figure 2.15 Error detection and correction for GeAr adder with N=12, R=2, P=6, k=3.

2.2.4 Correlation aware predictor (CAP)

Though GeAr was highly configurable but its latency was very much high. CAP was introduced that utilized the spatial temporal correlation of the bits and predicted the carry-in for the sub-adders.
By using an audio-signal, spatial and temporal correlations have been shown in the real-input streams and thus CAP utilized the correlation of carry signals.

CAP is a mux based predictor with controlling signals G(generate) and K(kill) giving outputs 1 or 0 respectively and when G=K=0 then for the predict-in carry, a flip flop has been used that would store the previous carry-in bit unlike in other adders where the predict carry is taken as 0.An EDC circuit has been included that consists of a XOR-gate that compares the predicted carry with the carry-out from the sub-adder and an error flag would be generated and extra cycles would be required to correct the output, also the flip flop has to be updated, explained in Figure 2.16.

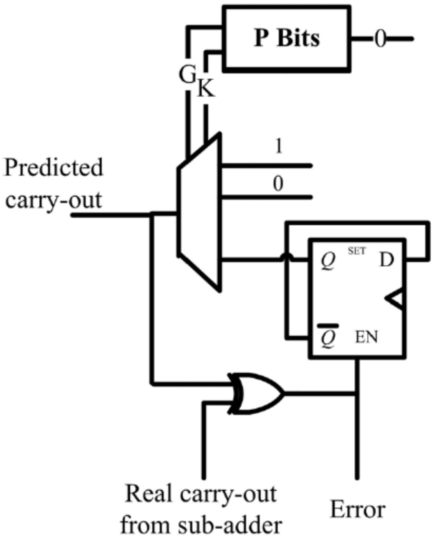


Figure 2.16 EDC circuit of CAP [5]

CAP used less predictive bits thus reducing the adder latency than the GeAr but still the error rate is higher than GeAr. So keeping in mind the accuracy, another adder Accurus has been introduced.

2.2.5 Accuracy configurable approximate adder (Accurus)

While ACA has been proved an efficient adder but still it lacks in the corrections made as they were performed in the least significant bit positions.

V. Benara *et al.* [32] proposed “Accurus: A fast convergence technique for accuracy configurable approximate adder circuits”.

In this adder. they proposed to start with the most significant bit rather than the least significant bit that resulted in fast convergence.

This has been explained in Figure 2.17. By using this proposed method, accurate results can be achieved in the initial stages of computation.

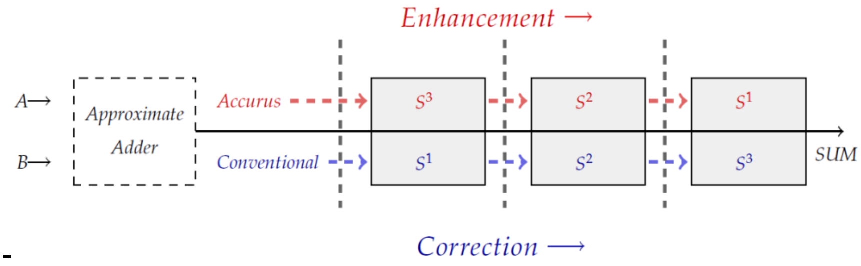


Figure 2.17 Unified Pipelined Correction Flow [32].

2.2.6 Reconfigurable approximate carry look-ahead adder (RAP-CLA)

Sometimes the accuracy requirements may vary and we require fast adders, so a Reconfigurable Approximate Carry Look-Ahead Adder (RAP-CLA) [36] comes into picture, which has the ability to switch between error-resilient and exact applications. The RAP-CLA consists of three circuitries: approximate part, the augmenting part and supplementary part. The combination of approximate and the augmenting part gives the correct C_{i+1} and for the approximate calculations, only the supplementary part is used. Only one mux is used and the selection is done by a 'selection operating mode signal' denoted by 'app' that chooses between the approximate and the accurate sum. A PMOS circuit is provided to remove the power consumption of the supplementary circuit when not in use as shown in Figure 2.18.

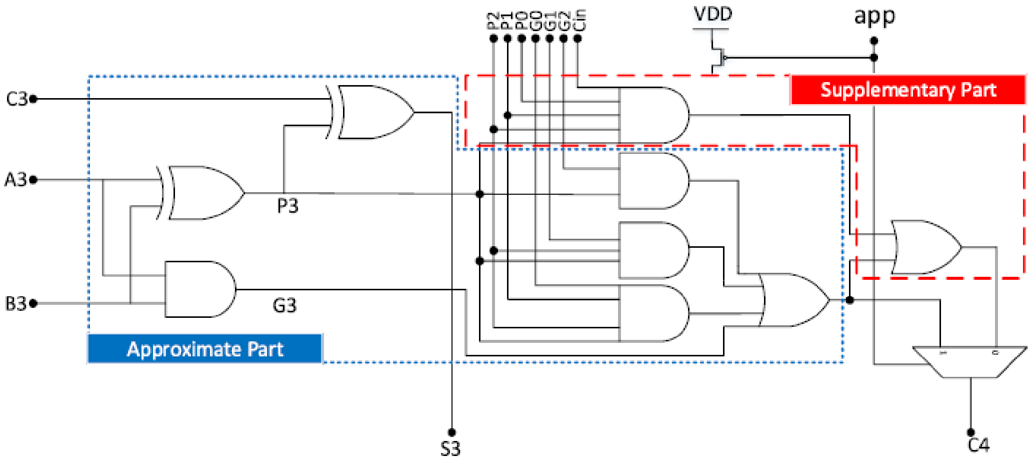


Figure 2.18 Circuit diagram for the RAP-CLA.

Thus CLAs with different accuracies can be designed using RAP-CLA. But the error delay product (EDP) for 8-bit adders has been gradually reduced than an 8-bit CLA. Another adder namely simple yet efficient accuracy- configurable adder design (SARA) [37] has been proposed to reduce the critical path and thus reducing the power consumption. This further improves the area-power-delay trade-off than the RAP-CLA.

2.2.7 Simple yet efficient accuracy- configurable adder design (SARA)

In the state-of-the-art adder SARA, carry-prediction has been used without increasing the overhead that is by not using any error-detection/correction logic circuit. This works on carry ripple adders (CRA) unlike some adders like RAP-CLA, which work on CLA. SARA consists of an N-bit adder comprising of K segments of L-bit sub-adders, which has been explained in Figure 2.19. The carry-bit prediction is used as

$$c_i^{prdt} = g_i \tag{2.5}$$

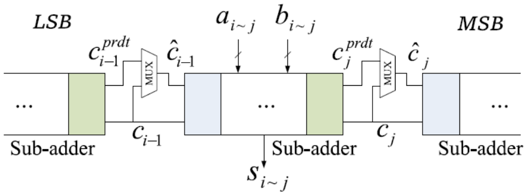


Figure 2.19 Design of SARA.

The selection between the two sub-adders have been done using the MUX with selection as $\hat{c}$, which is used in the following ways:

$$\widehat{c_i} = c_i{}^{prdt}, \text{ if approximation mode} \tag{2.6}$$

$$\widehat{c_i} = c_i , \text{ if accurate mode} \tag{2.7}$$

Such an approach as shown in Figure 2.20.

The inaccurate carry will not be propagated to c_i and the sum will be accurate when the addends are equal ($a_i = b_i$).

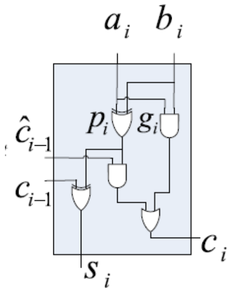


Figure 2.20 Carry-in configurable full adder.

Compared to the CRA, the overhead of the SARA is just the MUXs. This has greatly reduced the critical path and thus the power consumption, shown in Figure 2.21.

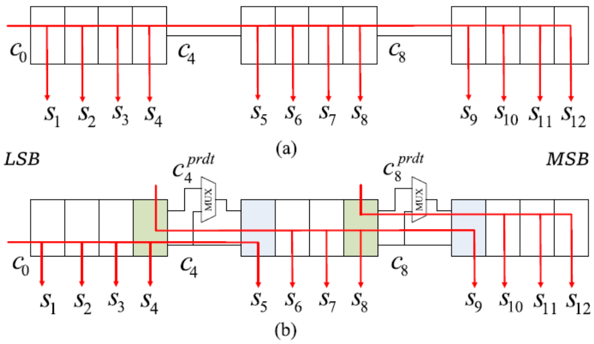


Figure.2.21 Implementation of 12-bit adder using a. CRA and b. SARA.

2.2.8 Approximate adder with low relative error and correct sign calculation

Most of the conventional adders focus on improving performance, reducing the power consumption and improving the delay and reducing the area but these adders rarely focus on controlling the relative error and potential sign error in the calculations. So J.Hu et al. [35] have proposed a new technique in their adder that ensures the correct sign calculations in 2's complement signed addition, which has been shown in Figure 2.22

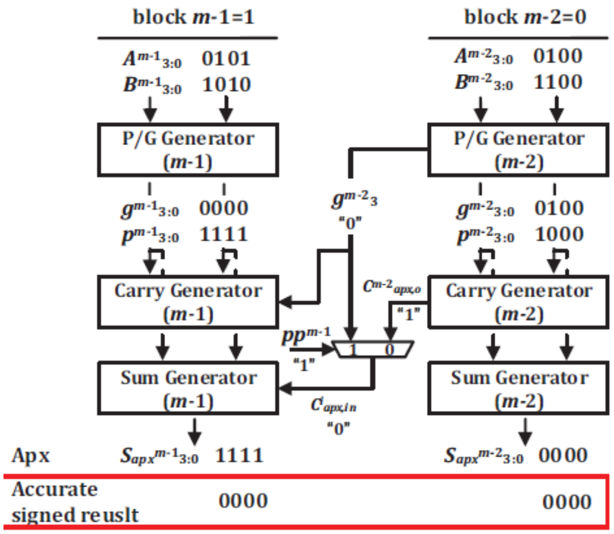


Figure 2.22 Case of approximate addition with potential sign error.

2.2.9 Reverse Carry Adder

In the state-of-the-art adders, it has been seen that the carry flows from LSB to the MSB and since propagating carry is more significant than the output carry so focusing on the propagating carry and that too from most significant bit to the least significant bit would provide with less delay, energy and energy-delay product [42][43]. Therefore O. Akbari *et al*. [36] presented a reverse carry propagate adder

(RCPA). In this RCPA structure, as shown in Figure 2.23, the carry signal flows from MSB to LSB and thus carry input signal is more significant than the output carry signal. Three implementations of the reverse carry propagate full adder (RCPFA) are proposed: RCPFA I, RCPFA II, RCPFA III. In this reverse carry propagate adder, the propagation is performed by introducing a forecast signal which acts like an output signal and also the weight of the carry decreases. This adder is less vulnerable to delay issues as compared to the conventional adders.

A conventional FA has 3 inputs with same weights and has two outputs with different weights.

$$2C_{i+1}+S_i=A_i+B_i+C_i \qquad (2.7)$$

Where A_i, B_i are the inputs of A and B respectively, C_i is the carry input and C_{i+1} is the carry output and S_i is the i_{th} bit of sum. Now we have a modified equation:

$$S_i-C_i=A_i+B_i-2C_{i+1} \qquad (2.8)$$

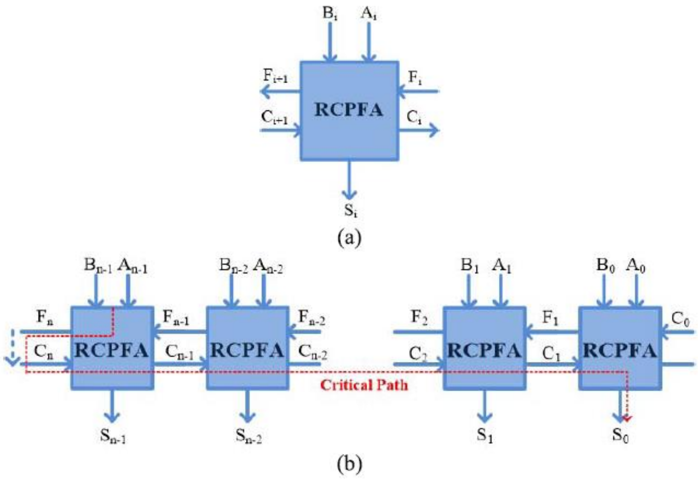


Figure 2.23 a) Block diagram of RCPFA and b) n-bit RCPA.

Although the proposed adder has provided efficient reduction in delay but most of the things are working on assumptions like taking a forecast signal which is acting as an output signal (F_{i+1}). So here C_{i+1} is taking an undecided value which cannot actually explains the flow of RCPA like C_{i+1} is taken as the MSB bit of any of the operands in the first MSB addition. Such assumptions cannot completely explain the working of the RCPFA. The review of the research papers have been summarized in the Table 2.2 given below.

Table 2.2 Comparative review analysis of state-of-the-art adders.

Year	**Publication**	**Adder Architectures**	**Improvements**	**Limitation/Future Scope**
2018	IEEE Transactions on Circuits and Systems-II	RAP-CLA: A Reconfigurable Approximate Carry Look-Ahead Adder	The ability to switch between error-resilient and exact applications	But the error delay product (EDP) for an 8-bit adders is gradually reduced over the CLA
2018	IEEE transactions on VLSI	A simple yet efficient accuracy- configurable adder (SARA)	Further improves the area-power-delay trade-off than the RAP-CLA	The accuracy has been greatly reduced
2018	IEEE Transactions on VLSI Systems	Approximate reverse carry propagate adder for energy-efficient DSP applications (RCPA)	Reduce the errors caused by setup time violation	Adders are still inefficient
2018	IEEE Transactions on Circuits and Systems	RAP-CLA	Can be switched between the approximate and exact operating modes	Error rate has increased
2017	IEEE 35th International Conference on computer design	Correlation Aware Predictor (CAP)	This utilizes spatial-temporal correlation for carry-in bit prediction for cub-adders thus improving the adder latency	Its error rate is still lower then GeAr.
2016	IEEE Computer Society Annual Symposium on VLSI	Accurus	Fast convergence of the result towards the most accurate results	No improvements in power consumption, area and clock period as compared to state- of-the-art adders
2015	Design Automation and Test in Europe DATE	Approximate Adder with Low Relative error and Correct Sign Calculation	Reduces relative error and ensures correct sign calculation in signed calculation	On increasing the block length(k), the delay of the adders is getting increased
2015	Design Automation Conference (DAC)	GeAr	Provides higher number of potential configurations	The output of GeAr has errors when the carry-in values of any sub-adders is mis-predicted.
2013	Proc. IEEE/ACM International Conf. (ICCAD)	GDA	Provides much better quality-effort tradeoff with its QCS	QCS can be further improved with different configuration strategies
2012	Design Automation Conference	ACA	ACA can operate in both accurate and inaccurate modes	Consumes 11.5% more power than conventional adder in accurate mode

	(DAC)			due to recovery circuits
2009	Integrated Circuits, ISIC’09. Proceedings of the 2009 12th International Symposium on IEEE	ETAI	To remove errors	Problem of calculating smaller number inputs
2009	Integrated Circuits, ISIC’09. Proceedings of the 2009 12th International Symposium on IEEE	ETAII	Speed performance improved with no degradation in power consumption	Degraded accuracy for large input operands
2009	Integrated Circuits, ISIC’09. Proceedings of the 2009 12th International Symposium on IEEE	ETAIIM	Accuracy improvements are highest compared to ETAI, ETAII	Transistor counts (area) is larger than ETAI

CHAPTER-3

RESEARCH OBJECTIVE AND METHODOLOGY

This chapter covers the research objectives followed by the methodology we used to achieve our research objectives.

3.1 RESEARCH OBJECTIVE

In the state-of-the-art adder RCPFA, though the adder is utilizing the mechanism of reverse carry propagation still there is further scope of improvements in the design by reducing the implementation complexity.

Following are the research objectives:

a. Designing a high-performance adder

b. Designing a low power adder.

c. Designing a low-complexity adder circuit.

The next section presents proposed methodology to achieve research objectives.

3.2 METHODOLOGY

We have followed a methodology following a series of steps:

It starts with the extensive literature survey on approximate adders that includes the previous research undertaken in the given field of interest. Then we proceed with the identification of research gaps, they are as follows a. Designing a high-performance adder, b Designing a low power adder, c. Designing a complexity adder circuit. The proposed architectures are designed in Verilog and implemented on Xilinx and the results are simulated using Synopsys Design Compiler using 65nm technology.

Thereafter, the implementation and simulation results of the proposed design have been compared with the state-of-the-art adder and analysis is done. If the calculated results get matched with the expected results, then we can proceed with the thesis writing otherwise we need to re-design and then again, the implementation and the simulation will be calculated and the same process will be carried out until the expected results are met.

The flowchart showing the methodology has been shown below.

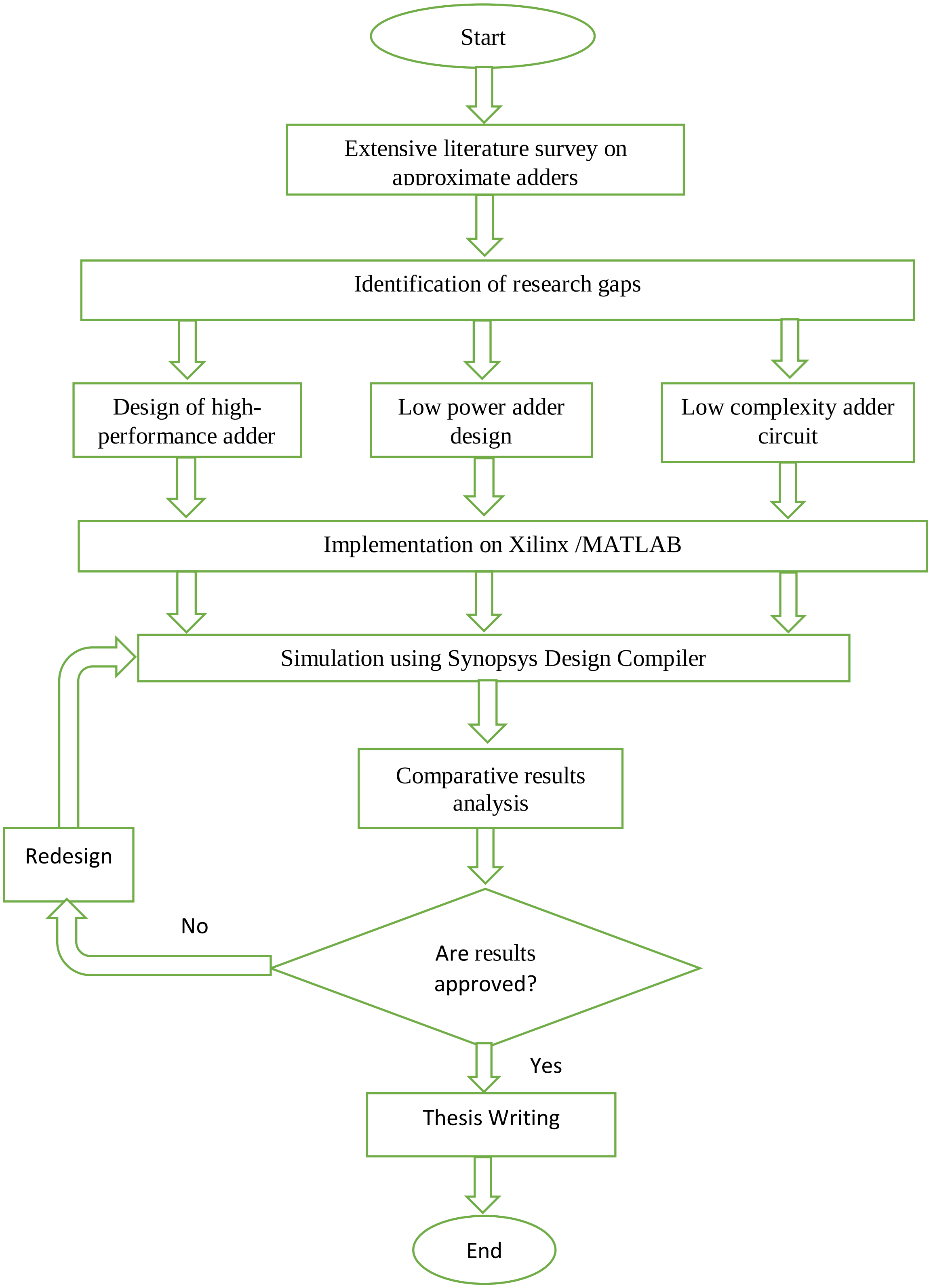


Figure 3.1 Flowchart showing the methodology.

CHAPTER-4

SIMPLIFIED REVERSE CARRY PROPAGATE ADDER

In this chapter, the different simplified reverse carry propagate adder (SRP adder) architectures have been presented depending on: F_i (Input forecast signal) and F_{i+1} (output forecast signal). Further a low-error rate SRP adder (LER-SRP adder) is introduced that improves the performance of the proposed SRP adder. The equations for sum (S) and carry (C) are mentioned below as given by RCPFA,

$$S_i = F_i(\overline{C_{i+1}} + A_iB_i) + \overline{C_{i+1}}(A_i + B_i) \tag{4.1}$$

$$C_i = F_i(C_{i+1} + \overline{A_i + B_i}) + C_{i+1}(\overline{A_iB_i}) \tag{4.2}$$

4.1 PROPOSED SIMPLIFIED REVERSE CARRY-PROPAGATE ADDER

Further improvements are possible in the previous RCPFA structures, like the reduction in the area and the improvement in the performance of the reverse carry adder, therefore assuming different logics for F_i and F_{i+1}.

4.1.1 SRP adder using F_i

In the proposed SRP adder, the forecast signal has been internally generated that is we have assumed different values for F_i. Following five kinds of different architectures have proposed, namely- SRP-I, SRP-II, SRP-III, SRP-IV and SRP-V.

4.1.1.1 *SRP-I*

In the structure SRP-I, by taking $F_i = 0$, different values of S_i and C_i have been calculated as mentioned in the truth Table, Table 4.1.

Table 4.1 Truth table of SRP-I showing the output expressions.

Inputs			**Outputs Expressions**	
A_i	B_i	C_{i+1}	S_i	C_i
0	0	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	0	1	0
0	1	1	0	1
0	1	0	1	0

The derived equations for S_i and C_i are given by:

$$S_i = (A_i + B_i)\overline{C_{i+1}} \tag{4.3}$$

$$C_i = C_{i+1}(\overline{A_i B_i}) \tag{4.4}$$

The structure of SRP-I is shown in Figure 4.1, this structure consists of two AND gates, one NAND, one OR and one NOT gate.

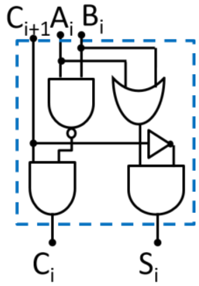


Figure 4.1 Structure of SRP-I

4.1.1.2 *SRP-II*

Using SRP-II, we have made three approximate adders with four-bit, eight-bit and sixteen-bit adders. Taking $F_i = A_i$, the equations for S_i and C_i are given by:

$$S_i = \overline{C_{i+1}}(A_i + B_i) + A_i B_i \tag{4.5}$$

$$C_i = C_{i+1} \tag{4.6}$$

These equations have been derived from the truth Table shown in Table 4.2

Table 4.2 Truth table of SRP-II showing the output expressions.

Inputs			**Outputs Expressions**	
A_i	B_i	C_{i+1}	S_i	C_i
0	0	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	0	1	0
0	1	1	0	1
0	1	0	1	0

The structure of SRP-II that consists of 5 gates has been shown in Figure 4.2.

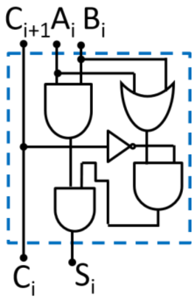


Figure 4.2 Structure of SRP-II

4.1.1.3 *SRP-III*

Using SRP-III, taking $F_i = A_i \oplus B_i$, the mentioned truth Table shown in Table 4.3 has covered different values of S_i and C_i for various inputs A_i and B_i.

Table 4.3 Truth table of SRP-III showing the output expressions.

Inputs			**Outputs Expressions**	
$\boldsymbol{A_i}$	$\boldsymbol{B_i}$	$\boldsymbol{C_{i+1}}$	$\boldsymbol{S_i}$	$\boldsymbol{C_i}$
0	0	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	0	1	0
0	1	1	0	1
0	1	0	1	0

Further equations for S_i and C_i are given as per the truth Table:

$$S_i = \overline{C_{i+1}}(A_i + B_i) \tag{4.7}$$

$$C_i = C_{i+1}(\overline{A_i B_i}) \tag{4.8}$$

The structure of SRP-III is shown in Figure 4.3

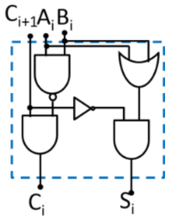


Figure 4.3 Structure of SRP-III

4.1.1.4 *SRP-IV*

In this case, we have taken $F_i = \overline{B_\imath}$, the equations for S_i and C_i are given by:

$$S_i = \overline{C_{\imath+1}} \tag{4.9}$$

$$C_i = C_{i+1}(\overline{A_\imath B_\imath}) + \overline{(A_\imath + B_\imath)} \tag{4.10}$$

These equations have been derived depending on the truth Table shown in Table 4.4.

Table 4.4 Truth table of SRP-IV showing the output expressions.

Inputs			**Outputs Expressions**	
$\boldsymbol{A_i}$	$\boldsymbol{B_i}$	$\boldsymbol{C_{i+1}}$	$\boldsymbol{S_i}$	$\boldsymbol{C_i}$
0	0	0	1	1
0	1	0	1	0
0	1	1	0	1
0	1	0	1	0
0	1	1	0	1
0	1	0	1	0

The structure of SRP-IV is shown in Figure 4.4

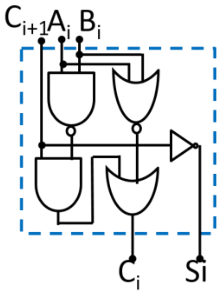


Figure 4.4 Structure of SRP-IV

4.1.1.5 *SRP-V*

For $F_i = 1$, the truth Table shows the different values of S_i and C_i, as shown in Table 4.5, the equations for S_i and C_i are given by:

$$S_i = \overline{C_{i+1}} + (A_i B_i) \tag{4.11}$$

$$C_i = C_{i+1} + (\overline{A_i + B_i}) \tag{4.12}$$

Table 4.5 Truth table of SRP-V showing the output expressions.

Inputs			**Outputs Expressions**	
A_i	B_i	C_{i+1}	S_i	C_i
0	0	0	1	1
0	1	0	1	0
0	1	1	0	1
0	1	0	1	0
0	1	1	0	1
0	1	0	1	0

The structure of SRP-I is shown in Figure 4.5

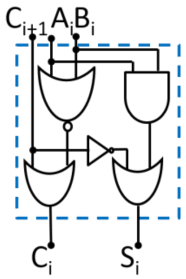


Figure 4.5 Structure of SRP-V

Though the SRP adders proposed using F_i have improved the design but still the forecast signal is not taking in account the previous inputs and that may be a drawback which led us to process for new architectures that are proposed by assuming the output forecast signal F_{i+1}.

4.1.2 SRP adders using $\mathrm{F}_{\mathrm{i+1}}$

In this scenario, the forecast output signal F_{i+1} has been assumed for different values of input A_i and B_i . The different structures have been proposed namely- SRP-A, SRP-B, SRP-C, SRP-D and have been explained below.

4.1.2.1 *SRP-A*

Using SRP-A, F_{i+1} has been assumed as 1. Three approximate adders with four-bit, eight-bit and sixteen-bit adders. Figure 4.6 shows the RTL schematic of SRP-A structure.

Table 4.6 Truth table of SRP-A showing the output expressions.

Inputs				**Outputs Expressions**	
$\boldsymbol{A_i}$	$\boldsymbol{B_i}$	$\boldsymbol{C_{i+1}}$	$\boldsymbol{F_i}$	$\boldsymbol{S_i}$	$\boldsymbol{C_i}$
0	1	1	1	0	1
1	0	1	1	0	1
1	1	1	1	1	1
1	1	1	0	1	1

Taking $F_{i+1} = 1$, the equations for S_i and C_i are given by:

$$S_i = A_i B_i C_{i+1} \tag{4.13}$$

$$C_i = C_{i+1} \tag{4.14}$$

The structure of SRP-A is shown in Figure 4.6

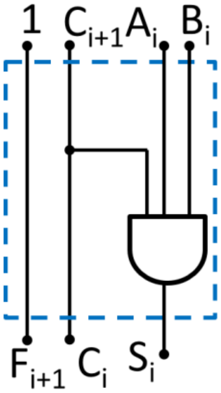


Figure 4.6 Structure of SRP-A

4.1.2.2 *SRP-B*

For SRP-B, $\overline{A_i + B_i}$ has been assumed for F_{i+1}. We have made three approximate adders with four-bit, eight-bit and sixteen-bit adders.

Table 4.7 Truth table of SRP-B showing the output expressions.

Inputs				**Outputs Expressions**	
$\boldsymbol{A_i}$	$\boldsymbol{B_i}$	$\boldsymbol{C_{i+1}}$	$\boldsymbol{F_i}$	$\boldsymbol{S_i}$	$\boldsymbol{C_i}$
1	0	0	0	1	0
0	1	0	0	1	0
0	0	0	0	0	0
0	1	0	1	1	0

Taking $F_{i+1} = \overline{A_i + B_i}$, the equations for S_i and C_i are given by:

$$S_i = \overline{C_{i+1}}(B_i + A_i\overline{F_i}) \tag{4.15}$$

$$C_i = 0 \tag{4.16}$$

The structure of SRP-B is shown in Figure 4.7

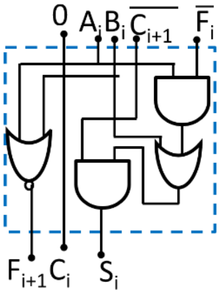


Figure 4.7 Structure of SRP-B

4.1.2.3 *SRP-C*

In this proposed structure 0 has been assumed for F_{i+1} with different values of A_i , B_i and C_i. Thus, three approximate adders with four-bit, eight-bit and sixteen-bit adders have been designed using SRP-C architecture.

Table 4.8 Truth table of SRP-C showing the output expressions.

Inputs				**Outputs Expressions**	
$\boldsymbol{A_i}$	$\boldsymbol{B_i}$	$\boldsymbol{C_{i+1}}$	$\boldsymbol{F_i}$	$\boldsymbol{S_i}$	$\boldsymbol{C_i}$
1	0	0	0	1	0
0	1	0	0	1	0
0	0	0	0	0	0
0	1	0	1	-*	-*

*- represents that this input will not occur corresponding to this output.

Taking $F_{i+1} = 0$, the equations for S_i and C_i are given by:

$$S_i = \overline{C_{i+!}}(A_i + B_i) \tag{4.17}$$

$$C_i = 0 \tag{4.18}$$

The structure of SRP-C is shown in Figure 4.8

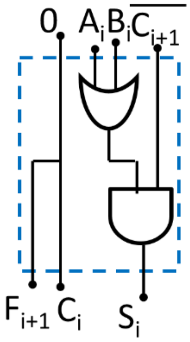


Figure 4.8 Structure of SRP-C

4.1.2.4 *SRP-D*

By assuming $F_{i+1} = (A_i\overline{B_i})$, we have made three approximate adders with four-bit, eight-bit and sixteen-bit adders.

Table 4.9 Truth table of SRP-D showing the output expressions.

Inputs				Outputs Expressions	
A_i	B_i	C_{i+1}	F_i	S_i	C_i
1	0	0	0	1	0
0	1	0	0	1	0
0	0	0	0	0	0
0	1	0	1	1	0

Taking $F_{i+1} = (A_i\overline{B_i})$, the equations for S_i and C_i are given by:

$$S_i = \overline{C_{i+1}}(B_i + A_i\overline{F_i}) \tag{4.19}$$

$$C_i = 0 \tag{4.20}$$

The structure of SRP-D is shown in Figure 4.9

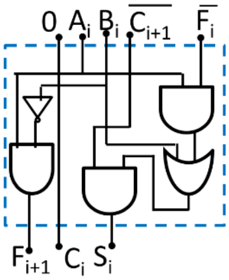


Figure 4.9 Structure of SRP-D

These structures are then applied to the FA as shown in Figure 10.

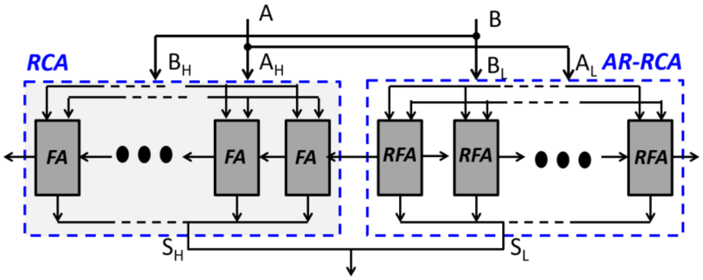


Figure 4.10 Proposed simplified reverse-carry propagate full adder (SRP-FA)

4.2 LOW ERROR RATE SIMPLIFIED REVERSE-CARRY PROPAGATE ADDER

We have proposed a low-error rate reverse carry propagate adder (LER-SRP adder) in which the MSB bits of both operands are denoted by A_H= A [7:4] and B_H= B [7:4] and LSB bits are denoted by A_L= A [3:0] and B_L= B [3:0]. This approximate adder has improved the performance -when the MSB's of the operands are 0 then the LSB values would move to the SRP, where normal addition will take place. So, in this way without adding any extra circuitry we would use the approximate circuit (SRP adder) only when required. Here if the MSB's are 0 that is A_H= '0000' and B_H= '0000' then CTL will be '0' and when this CTL would pass to MUX choosing A_L and B_L to move to the SRP. At the end the S [7:0] would show the total sum [$S_{H\,,}$ S_L] which has been explained in Figure 4.11.

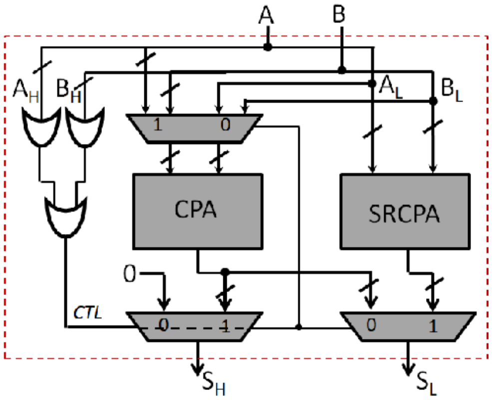


Figure 4.11 Block diagram of the proposed LER-SRP adder

This low-error rate SRP adder focuses on improving the error rate of the proposed approximate adders. By using the SRP adder when the MSB's are zero, we can utilize the SRP adder and get more correct values.

The next chapter shows the simulation results where we can clearly evaluate how the low-error rate SRP adder improves the functionality of the proposed SRP adder.

CHAPTER-5

SIMULATION ENVIRONMENT AND RESULT ANALYSIS

This chapter covers the different simulation environments being used for the designing of the proposed SRP adders. Thereafter the simulation results have been compared with the state-of-the-art adders and an analysis has been drawn.

5.1 SIMULATION ENVIRONMENT

The proposed SRP adder structures have been designed on block-level using HDL like Verilog and implemented on Xilinx. Thereafter the proposed designs have been simulated using 65nm technology on Synopsys Design Compiler. The design metrics have been calculated using MATLAB and at the last the Lena image (256x256) has been smoothed by Gaussian smoothing filters (GSF) [41] embedded with the proposed SRP and existing 16-bit approximate adders showing the image processing applications [44].

5.2 DESIGN AND QUALITY METRICS

The proposed SRP-FA have been implemented on Xilinx [45] and logic utilization data has been drawn that includes the number of LUTs and the delay(ns). Further the area(μm^2), power(μW) and delay(ns) have been calculated using the Synopsys design compiler [46]. Also, various quality metrics like MED, NED, PSNR, SSIM have been calculated using MATLAB [48].

5.3 SIMULATION RESULTS

In this section, the different implementation and simulation results of the proposed SRP-adders have been shown and a comparative analysis has been made.

5.3.1 For SRP-FA considering F_i

The proposed SRP-FA structures have been implemented on Xilinx using Verilog and its RTL schematics have been included in this report. The RTL schematics of SRP-FA-I has been shown xin Figure 5.1.

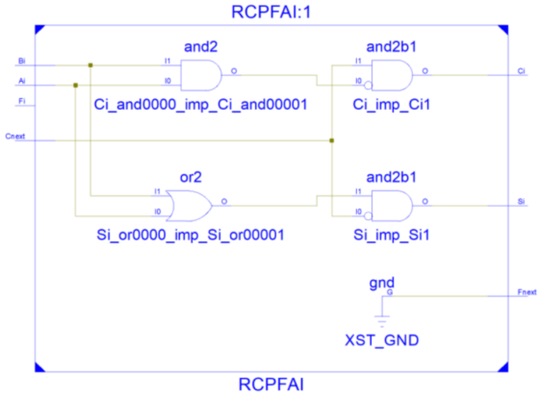
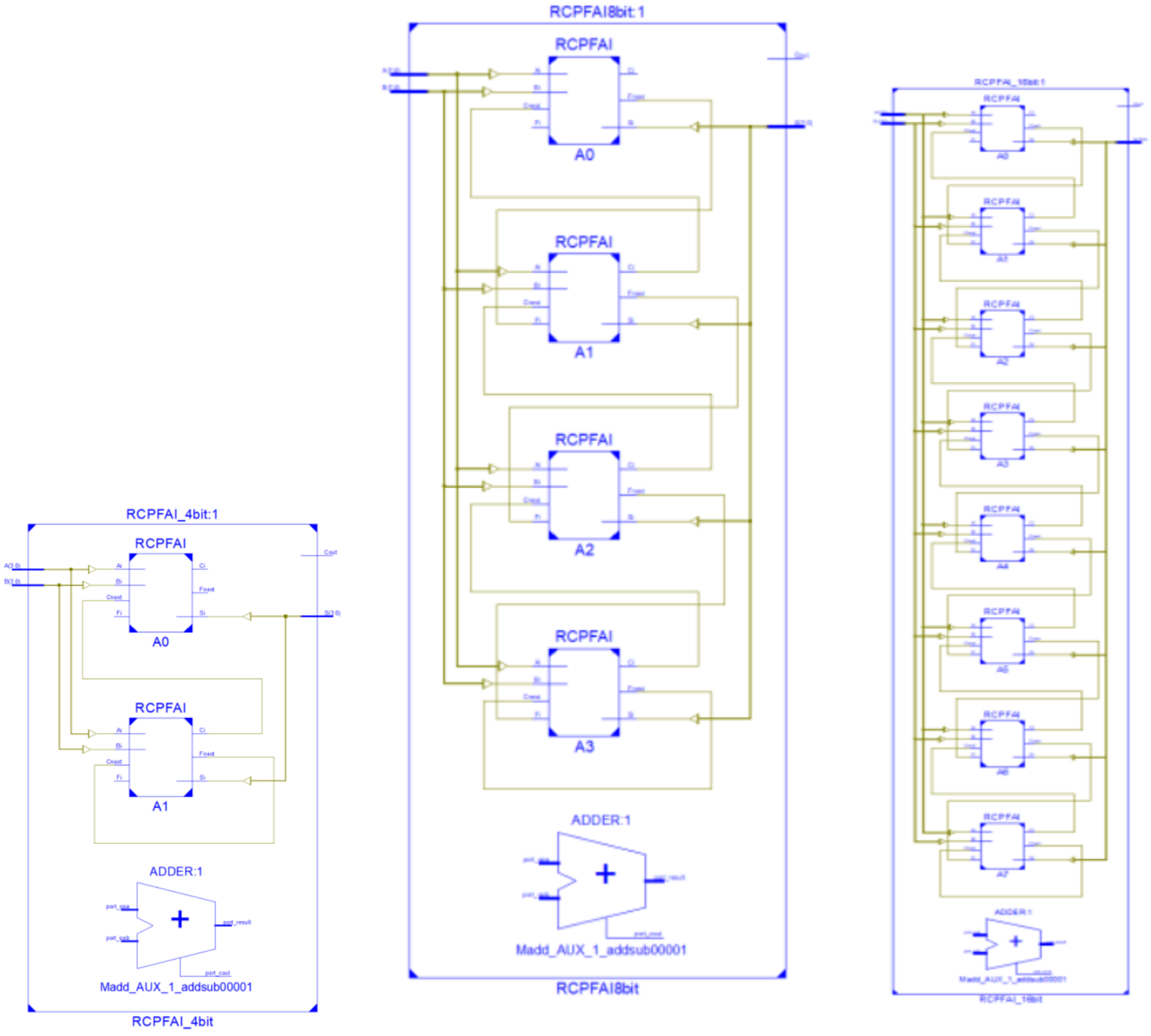


Figure 5.1 RTL schematic of SRP-FA-I structure

Using SRP-I, different three approximate adders with four-bit, eight-bit and sixteen-bit adders. In SRP-I_4BIT, we have taken 4-bit operands for A [3:0] and B [3:0] then for the MSBs, normal summation will take place and for the LSB's, approximate computation will take place using SRP-FA-I. Figure 5.2 shows the RTL schematic of SRP-I_4BIT.



a) Four-bit SRP-FA-I (b) Eight-bit SRP-FA-I (c) Sixteen-bit SRP-FA-I

Figure 5.2 RTL schematic of SRP-I- a) four-bit b) eight-bit and c) sixteen-bit.

Similarly, for SRP-I_8BIT, normal addition will take place for bits $S_H[7:4]$ and $S_L[3:0]$ will be calculated using SRP-I. The RTL schematic of SRP-I_8BIT is shown in Figure 5.2b. In the similar manner, in SRP-I_16BIT, the sum for the first eight bits that is $S_H[15:8]$ will be normal addition of bits but for the rest eight bits of LSBs will be added using SRP-I structure as shown in Figure 5.2(c).

In the next section, the RTL schematics of the SRP-FA using F_{i+1} has been discussed.

5.3.2 For SRP-FA considering F_{i+1}

Considering different values of F_{i+1}, the RTL schematics of the SRP-FA-A has been shown below.

5.3.2.1 *SRP-FA-A*

The proposed SRP-FA structures have been implemented on Xilinx using Verilog.

Its RTL schematics have been included in this report. The RTL schematics of SRP-FA-A has been shown in Figure 5.3.

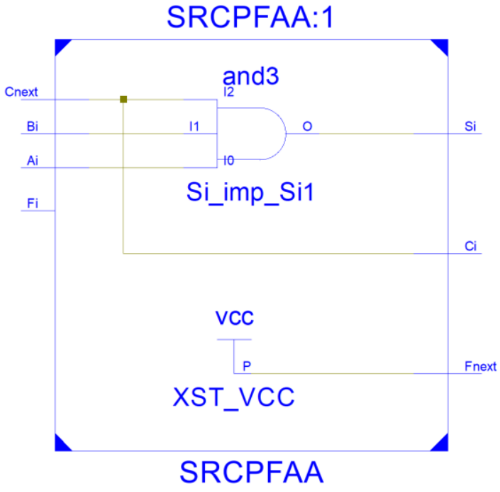


Figure 5.3 RTL schematic of SRP-A

Using SRP-A, we have made three approximate adders with four-bit, eight-bit and sixteen-bit adders. In SRP-A_4BIT, we have taken four-bit operands for A [3:0] and B [3:0] then for the MSBs, normal summation will take place and for the LSB's, approximate computation will take place using SRP-A. Figure 5.4 a) shows the RTL schematic of SRP-A_4BIT.

Similarly, for SRP-A_8BIT, normal addition will take place for bits S_H [7:4] and S_L [3:0] will be calculated using SRP-A. The RTL schematic of SRP-A _8BIT is shown in Figure 5.4 b).

In the similar manner, in SRP-A _16BIT, the sum for the first eight bits that is S_H 15:8] will be normal addition of bits but for the rest eight bits of LSBs will be added using SRP-A structure as shown in Figure 5.4 c.

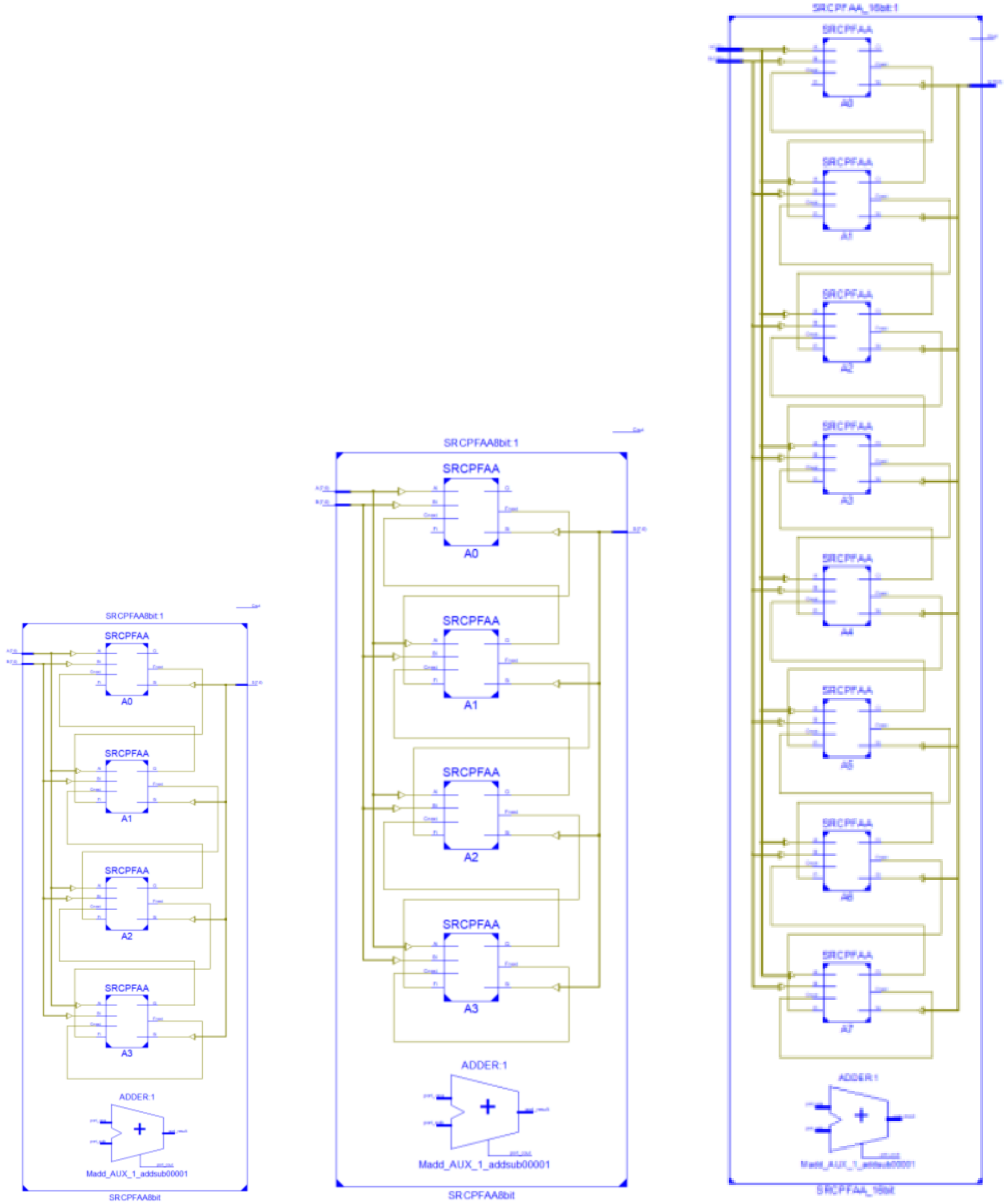


Figure 5.4 RTL schematic of SRP-FA-A a) four-bit b) eight-bit and c) sixteen-bit.

5.3.3 For low error rate simplified reverse carry-propagate adder

In the low-error rate SRP-adder (LER-SRP), the main focus is to improve the error-rate of the proposed adder. This LER-SRP adder is used when the MSB's are zero and thus the LSB's would be calculated using the RCA unlike approximate adder (SRP) in rest of the cases. Thus, improving the error rate.

Below mentioned are the different LER-SRP adders namely- four-bit LER-SRP, eight-bit SRP and sixteen bit SRP adders as shown in Figure 5.5.

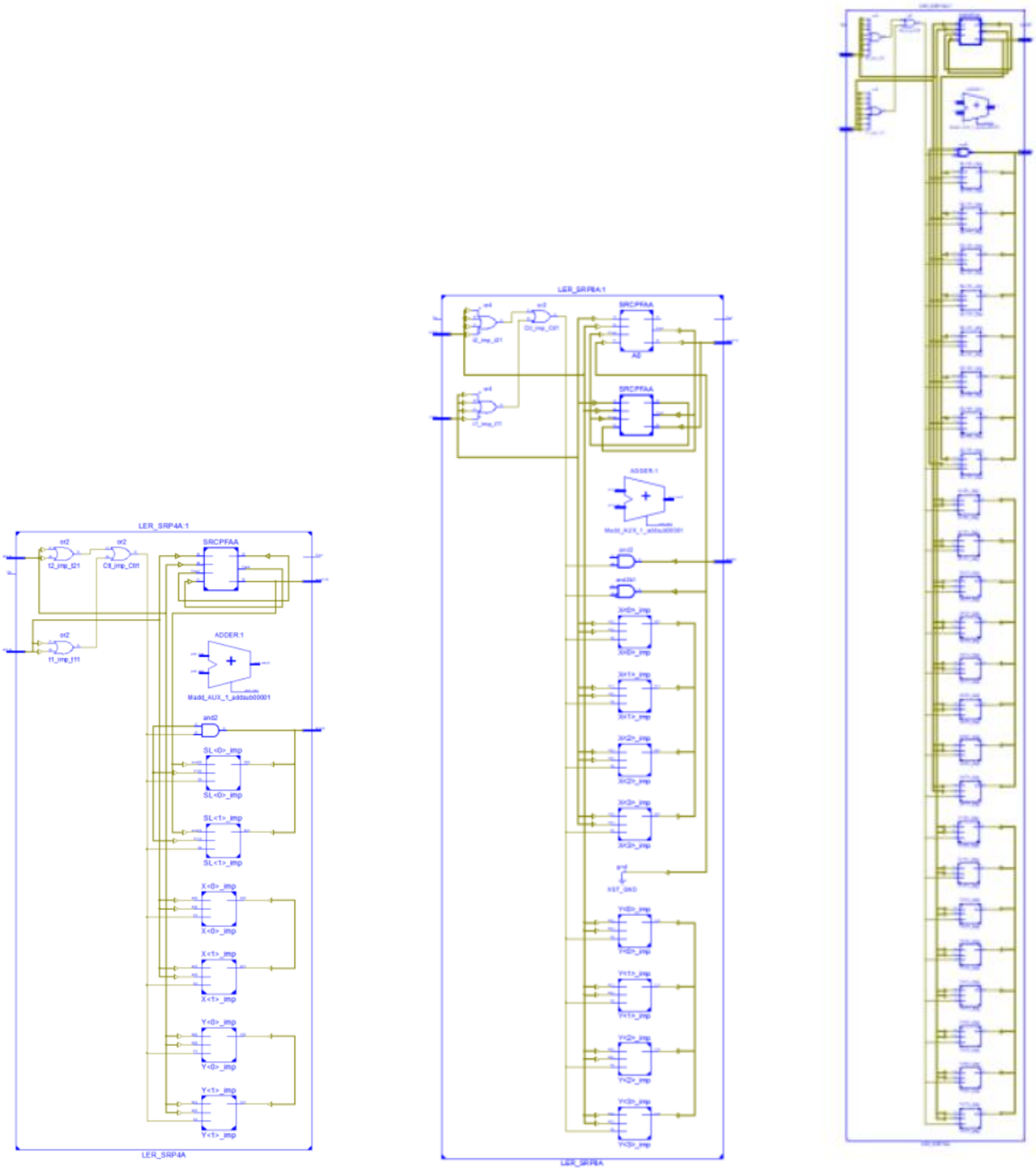


Figure 5.5 RTL schematic of LER-SRP-FA-A a) four-bit b) eight-bit and c) sixteen-bit.

5.4 DESIGN METRICS ANALYSIS

On implementing the proposed structures on Xilinx, the logic utilization has been plotted containing the number of LUTs and Delay. Table 5.1 shows the comparison of the implementation results of SPR-FA and RCPFA.

5.4.1 Implementation Results using Xilinx

The proposed SRP adders have been implemented on Xilinx using Verilog [48]. Various parameters like the number of LUT's and the delay have been calculated and the results are shown in Table 5.1

Table 5.1 Logic utilization table showing the comparison of SRP-FA with the state-of-the-art RCPFA: 5.1a) For eight-bit, b) for sixteen-bit.

Design	**8-bit**		**16-bit**	
	No. of LUTs	**Delay(ns)**	**No. of LUTs**	**Delay(ns)**
RCPFA1 [38]	**2**	**5.776**	23	12.261
RCPFA2[38]	12	8.959	27	10.649
RCPFA3[38]	11	7.898	22	11.090
Proposed SRP-FA-I	3	5.847	16	7.165
Proposed SRP-FA-II	11	6.869	**15**	**7.165**
Proposed SRP-FA-III	11	8.881	24	8.156
Proposed SRP-FA-IV	13	7.898	**15**	10.045
Proposed SRP-FA-V	20	7.978	16	7.165
Proposed SRP-FA-A	15	6.869	16	7.165
Proposed SRP-FA-B	11	7.898	16	7.165
Proposed SRP-FA-C	11	8.959	16	7.165
Proposed SRP-FA-D	**2**	**5.776**	15	7.165
Proposed LER-SRP-FA-I	26	12.927	45	12.328
Proposed LER-SRP-FA-II	25	12.927	44	12.328
Proposed LER-SRP-FA-III	29	12.927	53	12.328
Proposed LER-SRP-FA-IV	24	12.927	44	12.328
Proposed LER-SRP-FA-V	26	12.927	45	12.328
Proposed LER-SRP-FA-A	26	12.927	45	12.328
Proposed LER-SRP-FA-B	26	12.927	45	12.328
Proposed LER-SRP-FA-C	26	12.927	37	12.328
Proposed LER-SRP-FA-D	25	12.927	37	12.328

From the table, it can be concluded that the minimum no. of LUTs for SRP-FA-D is 2 similar to RCPFA1, also the delay is same that is 5.776 n sec. But for the sixteen-bit numbers, the no. of LUTs have been greatly reduced to a minimum of 15 with an improved delay of 7.165 n sec.

5.4.2 Simulation results on Synopsys design compiler

After getting satisfied results of implementation, the proposed structures have been simulated on the Synopsys Design Compiler [45] and the calculated power dissipation, area and delay have been shown below.

Table 5.2 Comparison of the simulation results of eight bit and sixteen-bit SRP-FA structures with the state-of-the-art-adders on Synopsys design compiler.

Adder	**8-bit adder**			**16-bit adder**		
Architectures	**Area(μm^2)**	**Power(μW)**	**Delay(ns)**	**Area(μm^2)**	**Power(μW)**	**Delay(ns)**
RCA	180	138.9	0.29	234.72	95.85	0.71
ACA [28]	372.60	441.12	0.19	481.68	435.90	0.27
RCPFA1[38]	165.23	172.00	0.19	301.32	175.90	0.37
RCPFA2[38]	113.40	126.00	0.19	230.40	138.00	0.35
RCPFA3[38]	**101.16**	**112.50**	**0.17**	**229.68**	**126.20**	**0.35**
Proposed SRP-FA-I	62.28	110.70	0.17	144.00	150.10	0.25
Proposed SRP-FA-II	**59.76**	**102.60**	**0.17**	**132.12**	**142.10**	**0.25**
Proposed SRP-FA-III	65.88	112.80	0.17	143.28	147.20	0.25
Proposed SRP-FA-IV	63.36	111.50	0.17	138.24	145.00	0.25
Proposed SRP-FA-V	62.28	110.70	0.17	137.16	144.40	0.25
Proposed SRP-FA-A	93.96	144.80	0.17	154.08	181.00	0.21
Proposed SRP-FA-B	113.76	120.60	0.17	216.72	177.40	0.23
Proposed SRP-FA-C	**99.36**	**110.10**	**0.17**	**180.00**	**158.00**	**0.23**
Proposed SRP-FA-D	109.08	129.70	0.17	219.60	178.30	0.23
Proposed LER-SRP-FA-I	242.64	116.40	0.35	291.24	134.40	0.45
Proposed LER-SRP-FA-II	172.80	103.20	0.33	299.88	134.20	0.45
Proposed LER-SRP-FA-III	219.24	118.10	0.33	282.24	140.01	0.43
Proposed LER-SRP-FA-IV	203.04	104.10	0.33	268.56	125.70	0.45
Proposed LER-SRP-FA-V	223.59	117.40	0.33	290.16	135.90	0.45
Proposed LER-SRP-FA-A	172.08	129.80	0.35	278.28	133.00	0.25
Proposed LER-SRP-FA-B	208.08	109.00	0.33	431.28	180.20	0.39
Proposed LER-SRP-FA-C	176.40	91.15	0.35	392.04	169.40	0.39
Proposed LER-SRP-FA-D	247.68	125.60	0.33	446.04	191.30	0.39

Using the simulation results of the proposed SRP adders and the low-error-rate SRP adders, an analysis has been shown below:

5.4.2.1 Simplified Reverse Carry-Propagate (SRP) adders

Using the simulation results of the proposed SRP adders and on comparing them with the state-of-the-art adders, for eight-bit numbers, at a constant delay of 0.17 n sec, **SRP-FA-II** is providing an improved power-area-delay product of **46%** over RCPFA-III. Also, about **85.6%** improvement over RCA and **96.6%** improvement over ACA has been provided by SRP-FA-II. For 16-bit numbers, SRCP-FA-II is providing an improved power-area-delay product of **53.73%** over RCPFA-III, **70.6%** improvement over RCA and about **91.7%** improvement over ACA.

In SRP-FA-C, for eight-bit numbers, at a constant delay of 0.17 n sec, it is providing an improved power-area-delay product of **1%** over RCPFA-III. Also, about **74%** improvement over RCA and **94%** improvement over ACA has been provided by SRP-FA-C. For 16-bit numbers, SRP-FA-C is providing an improved power-area-delay product of **35%** over RCPFA-III, **59%** improvement over RCA and about **88.4%** improvement over ACA.

5.4.2.2 Low-error rate simplified reverse-carry propagate (LER-SRP) adder

On comparing the simulation results of the proposed sixteen-bit SRP adders and LER-SRP adders, the area of the sixteen-bit SRP adder- SRP-FA-II has been found to be 132.12 μm^2 and an increased area of 229.88 μm^2 for LER-SRP-FA-II because of the increase in the circuitry. The power dissipation is a constant of 135.20 µW at the cost of increased delay of 0.20n sec.

5.5 QUALITY METRICS ANALYSIS

The performance metrics of the proposed adders can be calculated using MATLAB. These include calculating the design metrics like MED, NED, PSNR and SSIM.Thereafter the proposed adders have been embedded in the Gaussian smoothening filter-BG and the outputs have been included.

5.5.1 Quality metrics calculation using MATLAB

Apart from error metrics like mean error and mean square error, following other error metrics would also be calculated- Mean Error Distance (MED), Normalized Error Distance (NED), PSNR, SSIM. The quality metrics of different SRP-FA structures that have been compared to state-of-the-art adders is shown below in Table 5.3.

Table 5.3 Quality metrics table of different SRP-FA structures as compared to state-of-the-art adders.

	8-bit		**16-bit**	
Adder Name	**MED**	**NED**	**MED**	**NED**
ETA [24]	93.19	0.2128	153.109	0.6004
ACA [28]	7.390	0.1155	123.753	0.0302
RCPFA1[38]	8.008	0.4215	130.150	0.4080
RCPFA2[38]	**3.806**	**0.2500**	**63.612**	**0.2504**
RCPFA3[38]	15.012	0.5004	255.399	0.5008
Proposed SRP-FA-I	3.783	0.2522	63.8558	0.2504
Proposed SRP-FA-II	3.789	0.2526	63.8628	0.2514
Proposed SRP-FA-III	3.773	0.2515	64.0757	0.2513
Proposed SRP-FA-IV	3.748	0.2499	59.4881	0.2342
Proposed SRP-FA-V	3.878	0.2586	59.86	0.2357
Proposed SRP-FA-A	13.28	0.5030	223.81	0.5844
Proposed SRP-FA-B	3.77	0.2518	63.7537	0.2500
Proposed SRP-FA-C	3.76	0.2509	63.6348	0.2495
Proposed SRP-FA-D	3.77	0.2515	65.5755	0.2580
Proposed LER-SRP-FA-I	3.75	0.2501	63.6179	0.2495
Proposed LER-SRP-FA-II	3.74	0.2409	63.4432	0.2498
Proposed LER-SRP-FA-III	3.75	0.2505	63.8911	0.25
Proposed LER-SRP-FA-IV	**3.71**	**0.247**	**59.42**	**0.23**
Proposed LER-SRP-FA-V	3.85	0.2571	59.61	0.23
Proposed LER-SRP-FA-A	13.19	0.5738	223.19	0.5807
Proposed LER-SRP-FA-B	3.74	0.2497	63.45	0.249
Proposed LER-SRP-FA-C	3.75	0.2499	63.52	0.2491
Proposed LER-SRP-FA-D	3.74	0.2500	65.45	0.2567

From the design metrics table, it can be analyzed that the minimum MED has been provided by LER-SRP-FA-IV, for the eight-bit adder, the low error rate SRP adder that is 3.712 and MED for the sixteen-bit numbers is 59.42.

5.5.2 Quality metrics using Gaussian smoothening filter

In order to do quality analysis in the application, the Gaussian smoothening filters (GSF) [41] embedded with proposed adders and existing approximate adders are implemented and quality metrics such as PSNR and SSIM have been calculated and compared. The simulation results of the GSF embedded with proposed and existing adder are shown in Table 5.4.

Table 5.4 Quality metrics of GSF embedded with proposed and existing approximate adders.

	16-bit	
Adder name	**PSNR**	**SSIM**
ETA [24]	18.7546	0.8196
ACA [38]	18.7751	**0.5772**
RCPFA-I [38]	18.8486	0.8201
RCPFA-II [38]	18.9271	0.8222
RCPFA-III [38]	18.7546	0.8196
Proposed SRP-FA-I	18.9271	0.8222
Proposed SRP-FA-II	18.9271	0.8222
Proposed SRP-FA-III	18.9271	0.8222
Proposed SRP-FA-IV	18.9271	0.8222
Proposed SRP-FA-V	18.9271	0.8222
Proposed SRP-FA-A	18.7546	0.8196
Proposed SRP-FA-B	18.9271	0.8222
Proposed SRP-FA-C	18.9271	0.8222
Proposed SRP-FA-D	18.9212	0.8221
Proposed LER-SRP-FA-I	18.9271	0.8222
Proposed LER-SRP-FA-II	18.9271	0.8222
Proposed LER-SRP-FA-III	18.9271	0.8222
Proposed LER-SRP-FA-IV	18.9271	0.8222
Proposed LER-SRP-FA-V	**18.9464**	0.8223
Proposed LER-SRP-FA-A	18.7758	0.8209
Proposed LER-SRP-FA-B	**18.9464**	0.8233
Proposed LER-SRP-FA-C	18.9464	0.8233
Proposed LER-SRP-FA-D	18.9212	0.8221

It can be analyzed from the design metrics table that an improved PSNR of 18.9464 has been obtained from LER-SRP-FA-B that is a low-error rate SRP-adder.

Now to further obtain the improved results, the quality metrics are obtained by implementing all the proposed RCPA structures on MATLAB and simulating with the Lena image. Figure 5.1 shows the noisy Lena image filtered with state-of-the-art adders.

Figure 5.6 Lena image (256x256) soothed by GSF embedded with 16-bit a) ACA, b) ETA c) RCPFA-I d) RCPFA-III e) LER-SRP-I f) LER-SRP-V g) LER-SRP-A h) SRP-D

CHAPTER-6

CONCLUSION

In this report, two simplified reverse carry propagate (SRP) adders and one low-error rate SRP adder, have been proposed that are taking a reduced number of gates. The structures are proposed in the following two manners: a) When taking different values for forecast signal (F_i), the number of LUT's have been reduced to 15 for sixteen-bit SRP-FA-II keeping the delay constant (that is 7.165 ns) over the RCPFAs. b) When taking different values for F_{i+1}, the number of LUT's have been reduced to a minimum of 2 gates for eight-bit SRP-FA-D keeping the delay constant (that is 5.776ns) over the RCPFAs. Also, LER-SRP-FA-B has the best PSNR among all the proposed adders. Also, the MED has been reduced to 3.712 for eight-bit LER-SRP-FA-IV and 59.42 for sixteen-bit LER-SRP-FA-IV.

Using the simulation results of the proposed SRP adders and on comparing them with the state-of-the-art adders, for eight-bit numbers, at a constant delay of 0.17 ns, **SRP-FA-II** is providing an improved power-area-delay product of **46%** over RCPFA-III. Also, about **85.6%** improvement over RCA and **96.6%** improvement over ACA has been provided by SRP-FA-II. For 16-bit numbers, SRCP-FA-II is providing an improved power-area-delay product of **53.73%** over RCPFA-III, **70.6%** improvement over RCA and about **91.7%** improvement over ACA.

In 8-bit SRP-FA-C, at a constant delay of 0.17 ns, it is providing an improved power-area-delay product of **1%** over RCPFA-III. Also, about **74%** improvement over RCA and **94%** improvement over ACA has been provided by SRP-FA-C. For 16-bit numbers, SRP-FA-C is providing an improved power-area-delay product of **35%** over RCPFA-III, **59%** improvement over RCA and about **88.4%** improvement over ACA.

REFERENCES

[1] O. McSorley, "High speed arithmetic in binary computers," *IRE proceedings* vol.49, pp.67-91, 1961.

[2] Y.-K. Chen, J. Chhugani, P. Dubey, C. Hughes, D. Kim, S. Kumar, V. Lee, A. Nguyen, and M. Smelyanskiy, "Convergence of recognition, mining, and synthesis workloads and its implications," *Proceedings of the IEEE*, vol. 96, no. 5, pp. 790-807, 2008.

[3] Melvin A. Breuer and Haiyang Zhu, "Error-tolerance and multi-media," in *Proc.* of the 2006 *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp.45-52,2006.

[4] Bharat Garg, Chaitanya Goteti, and G. K. Sharma, "A Low-Cost Energy Efficient Image Scaling Processor for Multimedia Applications", *International Symposium on VLSI Design and Test (VDAT-2016*), pp. 1-6, 2016

[5] Melvin A. Breuer, "Let's think analog," in *Proc. of the IEEE Computer Society Annual Symposium on VLSI*, pp.67,2005.

[6] Nawandar A, Garg B, and Sharma G. K., "RICO: A low power Repetitive Iteration CORDIC for DSP applications in portable devices", *System Architecture, Elsevier 2016*, Vol 70, pp. 82-92.

[7] Mittal, Sparsh. "A survey of techniques for approximate computing." *ACM Computing Surveys (CSUR) 48.4*, 2016, pp.62.

[8] R. Venkatesan, A. Agarwal, K. Roy and A. Raghunathan, "Macao: Modeling and Analysis of circuits for approximate computing," in *Processings of the International Conference on Computer-Aided Design. IEEE Press,2011*, pp.667-673.

[9] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *Test Symposium (ETS)*,2013 18th IEEE European. IEEE, 2013, pp.1-6.

[10] Garg B, Bharadwaj N K, and Sharma G K, "Energy scalable approximate DCT architecture trading quality via boundary error-resiliency", *System-on-Chip Conference (SOCC), 27th IEEE Int.*, 2014, Las Vegas, USA.

[11] Kaushal V, Garg B, Jaiswal A, and Sharma G K, "Energy Aware Computation Driven Approximate DCT Architecture for Image Processing" *In VLSI Design (VLSID)*, 2015 28th Int. Conf. on, pp. 357-362. IEEE, 2015.

[12] Jaiswal A, Garg B, Kaushal V, and Sharma G K, "SPAA-Aware 2D Gaussian Smoothing Filter Design Using Efficient Approximation Techniques", *In VLSI Design (VLSID)*, 2015 28th International Conference on pp. 333-338, IEEE, 2015.

[13] Garg B, and Sharma G K, "PAID: Process Aware Imprecise DCT Architecture Trading Quality for Energy Efficiency", *Journal of Low Power Electronics*, Vol. 11, No. 2 (2015): 121-132.

[14] 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.

[15] Khuraijam Nelson Singh and H. Tarunkumar, "A review on Various Multipliers Designs in VLSI", *IEEE Indicon*, vol.7, pp-214-218, 2018.

[16] P. Kulkarni, P. Gupta, M. Ercegovac, Trading Accuracy for Power with an Undersigned Multiplier Architecture. In *Proc. IEEE international Conference on VLSI Design*, pp.-346-351, Jan-2011.

[17] Garg B, and Sharma G K, "Low Power Signal Processing via Approximate Multiplier for Error-Resilient Applications", *In Industrial and Information Systems (ICIIS-2016),* 11th Int. Conference on, pp. 1-6, 2016

[18] Garg B, and Sharma G. K., "ACM: An Energy-efficient Accuracy Configurable Multiplier for Error-resilient Applications", *Journal of Electronics Testing*, Vol. 33, 2017.

[19] V. Gupta, D. Mohapatra, S.P. Park, A. Raghunathan and K. Roy, "Impact: Imprecise adders for low-power approximate computing," in *Low Power Electronics and Design (ISLPED) 2011 International Symposium* on, Aug 2011, pp.409-414.

[20] V. Gupta, D. Mohapatra, A. Raghunathan and K. Roy, "Low-power digital signal processing using approximate adders," *IEEE Trans. Computer-Aided Design Integrated Circuits Syst.*, vol.32, no.1, pp. 124-137, Jan2013

[21] Celia D, Vinita Vasudevan and Nitin Chandrachoodan, "Optimizing power accuracy trade-off in approximate adders", *in DATE,*2018, pp.-1-10.

[22] Garg B, and Sharma G K, "Block Matching Algorithm for Deriving Quality-Tuneable Motion Estimation Architecture", In *Industrial and Information Systems (ICIIS-2016*), 11th Int. Conference on, pp. 1-6, 2016

[23] V. Chippa, S. Chakradhar, K. Roy and A. Raghunathan, "Analysis and characterization of inherent application resilience for approximate computing," *Design Automation Conference (DAC),*2013.

[24] N. Zhu, W.L. Goh and K.S. Yeo, "An enhanced low-power high-speed adder for error-tolerant application," in *Integrated Circuits, ISIC'09. Proceedings of the 2009 12th International Symposium on. IEEE,*2009, pp.69-72.

[25] N. Zhu, W.L. Gosh, 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,” *Very Large-Scale Integration (VLSI)Systems, IEEE Transactions* on, vol.18, no.8, pp.1225-1229,2010.

[26] N. Zhu, W.L. Goh, G. Wang and K.S. Yeo, “Enhanced low-power high-speed adder for error-tolerant application,” in *SoC Design Conference (ISOCC) 2010 International. IEEE*, 2010, pp. 323-327.

[27] N. Zhu, W. Goh and K. Yeo, “An Enhanced Low-Power High-Speed Adder for Error-Tolerant Application: *Processing International Symposium on Integrated Circuits*, 2009, pp.69-72.

[28] A.B. Kahng and S. Kang, “Accuracy-configurable adder for approximate arithmetic designs,” in *Proc. 49th ACM/EDA/IEEE Design Automation Conference (DAC)*, San Francisco, CA, USA,2012, pp.850-825.

[29] R. Ye, T. Wang, F. Yuan, R. Kumar and Q. Xu,” On reconfiguration-oriented approximate adder design and its application,”, in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD)*, San Jose, CA, USA, 2013, pp. 48-54.

[30] M. Shafique, W. Ahmad, R. Hafiz and J. Henkel, “A low latency generic accuracy configurable adder,” in *Proc. 52nd ACM/EDA/IEEE Design Automation Conference (DAC)*, San Francisco, CA, USA,2015, pp.7-11.

[31] Xiao ling Chen, Ahmed M. Eltawil, Fadi J. Kurdahi, “Low latency approximate adder for highly correlated input streams”, in *IEEE 35th International Conference on computer design*,2017, pp.1063-6404.

[32] V. Benara and S, Purini, “Accurus: A Fast Convergence Technique for Accuracy Configurable Approximate Adder Circuits,” in *IEEE Computer Society Annual Symposium on VLSI*,2016

[33] Ayad Dalloo, Ardalan Najafi and Alberto Garcia-Ortiz, “Systematic design of an approximate adder: the optimized lower part constant-OR adder”, in *IEEE transactions on VLSI systems*, august 2018, pp.1063-8210.

[34] A.K. Verma, P. Brisk and P. Ienne, “Variable Latency Speculative Addition: A New Paradigm for Arithmetic Circuit Design”, *Proc. DATE,* pp.62-69,2008.

[35] J. Hu and W. Qian, “A new approximate adder with low relative error and correct sign calculation,” in *Proc. Design Automation Test Europe Conf. Exhibit. (DATE*), Grenoble, France,2015, pp. 1449-1454.

[36] O. Akbari, M. Kamal, A.A. Kusha, M. Pedram, “RAP-CLA: A Reconfigurable Approximate Carry Look-Ahead Adder,” in *IEEE Transactions on Circuits and Systems-II: Express Briefs*, Vol.65, No.8, August 2018, pp.1549-7747.

[37] Wenbin Xu, Sachin S. Sapatnekar, Jiang Hu,” A simple yet efficient accuracy-configurable adder (SARA)”, in *IEEE transactions on VLSI,* June 2018, pp.1063-8210.

[38] Masoud Pashaeifer, Mehdi Kamal et al., “Approximate reverse carry propagate adder for energy-efficient DSP applications”, in *IEEE transactions on VLSI systems*, 2018, pp.1063-8210.

[39] O. Bedrij, “Carry select adder”, *IRE Trans. On Electronic Computers*, vol. EC-11, pp.340-346,1962.

[40] P. Balasubramanian, C. Dangetal “Approximate ripple carry and carry lookahead adders-a comparative analysis”, in *proc IEEE,* Oct 2017, pp.1-15.

[41] Garg B and Sharma G K, "A quality-aware Energy-scalable Gaussian Smoothing Filter for image processing applications" *Microprocessors and Microsystems Elsevier* (2016) Vol. 45, pp. 1-9

[42] Garg B, Dutt S, and Sharma G K, "Bit-width-aware constant-delay run-time Accuracy Programmable Adder for error-resilient applications" *Microelectronics Journal, Elsevier* 2016, Vol. 50, pp. 1-7.

[43] B.K. Mohanty and S.K. Patel “Area-delay-power efficient carry-select adder”, *IEEE Trans. Circuits Syst. II, Exp. Briefs,* vol.61, no. 6, pp. 418-422, Jun.2014.

[44] Garg B. and Sharma G K, “A process tolerant low-power adder architecture for image processing applications”, *Turkish Journal of Electronics and Computer Engineering,* Vol. 27 (3), pp. 1839-1854, 2019

[45] Xilinx ISE, Xilinx Inc.

[46] Design Compiler, Synopsys Inc.

[47] MATLAB, MathWorks.

[48] Samir Palnitkar. *Verilog HDL*, Sunsoft Press, 1996.

LIST OF PUBLICATIONS

[1] Yashoda Bisht and Bharat Garg, "A Novel High-Performance Reverse Carry Propagate Adder for Energy Efficient Multimedia Applications", communicated to 5th IEEE International Symposium on Smart Electronics Systems 2019.

Report

ORIGINALITY REPORT

12%	7%	5%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	etd.lib.nsysu.edu.tw Internet Source	4%
2	repository.tudelft.nl Internet Source	2%
3	digital.library.unt.edu Internet Source	1%
4	Lecture Notes in Computer Science, 2004. Publication	1%
5	Yang Yang, Fan Jian jun, Hong Yuan, Jin Shu xin, Yang Yu fei. "Chapter 32 Research of Navigation Constellation Independent Punctuality Based on Different Configuration of Satellite Clocks", Springer Science and Business Media LLC, 2015 Publication	1%
6	Tongxin Yang, Tomoaki Ukezono, Toshinori Sato. "A Low-Power Yet High-Speed Configurable Adder for Approximate Computing", 2018 IEEE International	1%

15/07/19
(Supervisor)

Yashoda
15.07.19

Symposium on Circuits and Systems (ISCAS), 2018
Publication

---

7 Sarvenaz Tajasob, Morteza Rezaalipour, Masoud Dehyadegari, Mahdi Nazm Bojnordi. "Designing Efficient Imprecise Adders using Multi-bit Approximate Building Blocks", Proceedings of the International Symposium on Low Power Electronics and Design - ISLPED '18, 2018
Publication

1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

15/07/19
(Supervisor)

Yashoda
15.07.19