# Design of Efficient Approximate Reverse Carry Propagate Adder

**Telagamalla Gopi**
Asst. Professor, Dept of ECE
Annamacharya Institute of Technology And Science's, Hyderabad,
Telangana State, India

***Abstract-*** *Adders plays vital role in many digital signal processing applications and An efficient Reverse carry signal propagates from MSB bit to LSB bit, here carry input signal has more significance compared to output signal. Here in this paper we propose three different designs of the reverse carry propagate full-adder cell with different delay, energy, and accuracy configurations. And also we use some hybrid structure in the n bit adder design were least half of the adder design is implemented with this RCPA and most significant half will be added by using some accurate high speed adder like Kogge stone adder which can enhance the adder speed. The Hybrid adders realized utilizing these structures are studied and compared those with conventional approximate adders using Xilinx ISE 14.7 with Verilog HDL coding.*

***Keywords***- Full Adder, digital signal processing (DSP), Reverse carry propagate adder (RCPA), Kogge-Stone adder (KSA).

## I. INTRODUCTION

Adder blocks, which are the main components in arithmetic units of DSP systems, are power hungry and often form hot-spot locations on the die. These facts have been the motivations for realizing this component using the approximate computing approach. Prior researches on approximate adders have taken two general approaches of focusing on error weight and error probability reductions. Power consumption reduction and speed improvement are the key goals in the design of digital processing units, especially the portable systems. Normally, an increase in the speed is achieved at the cost of more power Consumption for exact processing units.

One of the approaches to improve both the power and speed is to sacrifice the computation exactness. This approach, which is approximate computing, may be used for the applications where some errors maybe tolerated. The first approach is based on a hybrid structure adder where two different parts, exact MSB's, and approximate least significant bits (LSBs) are utilized. The error appears in the carry input of the exact most significant bit (MSB) part and the summation in the LSB part.It limits the error weight to the weight of the carry input of the MSB part. Since normally most of the activities occur in the LSB part. In the second approach, pure approximate adder structures are employed. For these adders, reducing the error probability of the summation as well as reducing the power and delay is the key design criteria.

In this approximate adder propagates the input carry from the higher significant bitto lower significant bit to form the carry output. In this type of adder, the propagation is performed by introducing a forecast signal acting as an output signal. The MSB half will be added with the help of parallel prefix adder which can have better speed of operation. Lower part will be added with the help of approximate reverse carry propagate adder.

## II. LITERATURE SURVEY

Designing an adder with carry propagation will take more delay so if we avoid the carry propagation while designing an adder will effects the better adder performance so we go for reverse carry propagate adder in which the carry is propagated in reverse direction.

**Ripple carry adder:** It is simplest adder among all adders but slowest adder, it requires o(n) and delay of o(n),where n represent the operand size

**Carry look ahead adder:** It has good area o(nlogn) and good delay of o(logn),but suffers from irregular layout design

**Carry select adder:** It has area of o(n) and delay with $o(n^{l+2/l+1})$ ,and it is the best adder in terms of area and delay

**Carry save adder:** Requires area o(n) and delay of o(logn)
Carry select adder is ths fast adder as it reduces computation time for operation among all adders but suffers from fanout limitation. The sorting problem is defined as the rearrangement of N input values so that they are in ascending order, merge sort method uses divide and conquer algorithm and uses recursion to perform sorting

Among all the exact adder structures, a Ripple carry adder (RCA) has the worst power and area utilizations. It, however, suffers from a large delay. To improve the speed and energy efficiency of this adder, some prior works have sacrificed the accuracy. Approximate RCA structure is implemented .in this input operands are divided in to two parts exact MSB computation part and inexact LSB computation part, in LSB inexact part includes carry free addition part. Hybrid adders with large sizes for the approximate part which determines the critical path of the adder. The design parameters of the adder depend on the width of the approximate part. This is especially advantageous in the case of hybrid adders with large sizes for the approximate part which determines the critical path of the adder

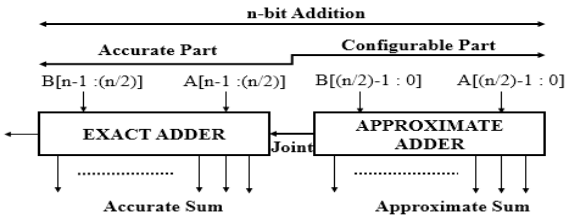


**Fig.1: n-bit adder structure**

Based on our requirement we have three design types of reverse carry propagate adders (DESIGN-I, DESIGN-II, DESIGN-III for16-bit or 32-bit) the designs. By using these proposed adders we can implement 3 types of reverse propagate adders and are used in the place of approximate adders .As mentioned before, the weight of the carry decreases as the carry propagates in a counter-flow manner.

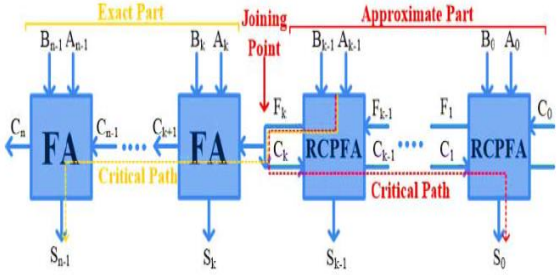


**Fig.2 Architecture of an n-bit adder with RCPA**

The proposed RCPFAs may be used in hybrid adders whose general *n*-bit structure based on the RCPFA's. Obviously, the design parameters of the adder depend on the width of the approximate part.

We focus on the hybrid adders where the use of the approximate reverse carry propagate full-adder (RCPFA) is suggested. The approximate adder propagates the input carry in a counter-flow manner, i.e., from the higher significant bit to lower significant bit to form the carry output.In this type of addition weigt of carry decreases as it propagates. The MSB half will be added with the help of parallel-prefix adder which can have better speed of operation and lower part will be added with the help of approximate reverse carry propagate adder.

In hybrid structures two different parts are used to find exact MSB bit's and approximate LSB bit's, eeror appers in carry input of exact MSB bits's part and summation part of LSB bit's part, so it limits the weight of the error to the weight of carry input In the second approach, pure approximate adder structures are employed. For these adders, reducing the error probability of the summation as well as reducing the power and delay is the key design criteria.

## III. PROPOSED HYBRID ADDER

In hybrid adders with two different n/2 parts are used to find exact MSB bit's and aprroximate LSB bit's,in this adder error appears in carry input of exact MSB part and summation part of LSB part, so weight of error is limitted to weight of carry. For these adders, reducing error probability, area and power all are considered to design adder. In hybrid adders approximate RCPA is suggested, where input carries counter–flow maner i.e from MSB bit to LSB bit and so called approximate reverse carry adder (RCPFA).The MSB part will be added with the help of parallel-prefix adder which can have better speed of operation and LSB part will be added with the help of approximate reverse carry propagate adder.Adder with carry propagation makes more delay, so if make carry propagation in reverse direction better performance of adder and speed can be achieved.

As the generation and propagation of the carry input of the MSB's through small LSB bit's of full adder, short delay causes large amount of error, if carry is propagated in reverse direction error probability can be decreased.

Each exact FA generates its carry output and sum signals using

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

Where $A_i$ ($B_i$) is the $i^{th}$ bit of the input A(B),$C_i$($C_{i+1}$)is the carry input (output), and Si is the $i^{th}$ bit of the sum.

Based on this equation, the output signals in the $i^{th}$ bit position depends on the $i^{th}$ bits of the inputs *A* and *B* and the carry output of the previous position $(Ci)$. By moving the term $C_i$($C_{i+1}$)to the left (right) side of the equation, one may write

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

Considering (2), one may think of a full adder as a structure which operation depends on the carry output of the (i+1) $n^{th}$ bit position $C_{i+1}$ and its input operand bits.

These full adders have four inputs and three outputs.The inputs are the input operands (Ai and Bi), the carry output of the next bit position ($C_{i+1}$), and a forecast signal (Fi). The RCPFA determines the summation result (Si), carry (Ci), and the forecast signal ($F_{i+1}$) as its output signals. As mentioned before, the advantage of the RCPA is that the value of the error is in the direction of decrease in the bit significance.

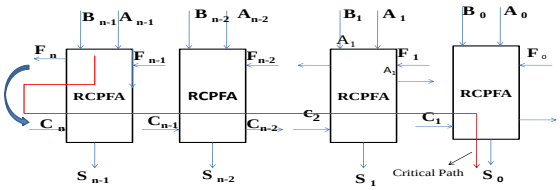


**Fig.3: RCPFA Adder**

The concept in parallel prefix adders is to compute a small group of intermediate prefixes and then find the large group prefixes, until all the carry bits are computed. Parallel prefix addition of the operands A and B of width n is done in three steps as given below

1. Pre Processing
2. Carry Generation
3. Post processing

**Pre-processing stage**

In this stage we compute the generate and propagate signals are used to generate carry input of each adder.A and B are inputs. These signals are given by the equation 1&2.

$$P_I=A_i \text{ XOR } B_i \quad \text{(3)}$$
$$G_I=A_iXB_i \quad \text{(4)}$$

**Carry generation stage**

In this stage we compute carriers corresponding to each bit. Execution is done in parallel form.After the computation of carriers in parallel they are divided into smaller pieces.Carry operator contain two AND gates, one OR gate.

$$P_{(i-k)} = P_{(i-j)}.P_{(j-i-k)} \quad \text{(5)}$$
$$G_{(i-k)} = G_{(i-j)} + (G_{(j-i-k)}.P_{(i-j)}) \quad \text{(6)}$$

This carry is generated by using different cell structures called Gray cell, Black cell and Buffer cell.By using this cell structures we can calculate final carry and are same for all parallel prefix adders but the design of carry generation is different.

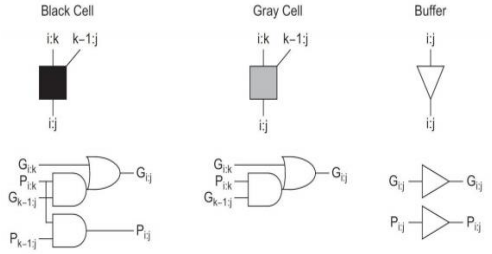


**Fig.4: Black Cells, Gray Cells and Buffer cell used for carry generation stage**

**Post Processing**

The sum bits are generated either by using simple XOR gates or by the use of conditional sum adders. In conditional sum adders, for each bit position, two tentative sums will be generated and the correct one will be selected when the relevant carry for that bit arrives. By implementing the adder with two sub adders like Kogge-stone adder and reverse carry propagate adder we can have better in terms of delay. Here in this paper we present 16-bit and 32-bit adders.

## IV. RESULTS AND DISCUSSION

By implementing the adder with two sub adders like Kogge-stone adder and reverse carry propagate adder we can have better in terms of delay. Here in this paper we present 16-bit and 32-bit adders with 3 different designs of reverse carry propagate adders. The results are presented in the below table

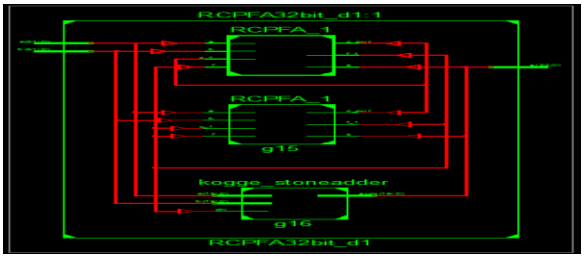


**Fig.5: RTL Schematic of proposed 32 bit adder**


**Fig.6: Simulation output of proposed 32 bit Adder**

## V. CONCLUSION

In this paper we have designed and implemented an n-bit adder with a hybrid architecture where we use two types of adders for implementation of large adder. We use approximation in lower side addition that is least significant side and accurate addition in higher side that is most significant side. The accurate adder we used in this project is a parallel prefix adder called Kogge stone adder and the approximate adder we used is the RCPA we proposed in this paper. We have designed and implemented an n-bit adder with a hybrid architecture where we use two types of adders for implementation of large adder. Parallel prefix kogge-stone adder is used for exact bit calculation i.e msb bits's and approximate carry propagate adder is used for least significant bits i.e lsb bit's. This design reduces delay but decreses accuracy **a**nd occupies more area

## REFERENCES

[1] S. Mittal, "A survey of techniques for approximate computing," ACM Comput. Surv., vol. 48, no. 4, pp. 62-1–62-33, Mar. 2016.

[2] B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in Proc. 49th Annu. Design Autom. Conf. (DAC), Jun. 2012, pp. 820–825.

[3] T. Moreau, A. Sampson, and L. Ceze, "Approximate computing: Making mobile systems more efficient," *IEEE Pervasive Comput.*, vol. 14, no. 2, pp. 9–13, Apr. 2015.

[4] AungMyo San, Alexey N. Yakunin, "Reducing the Hardware Complexity of a Parallel Prefix Adder" 978-1-5386-4340-2/ 2018 IEEE

[5] J. Kung, D. Kim, and S. Mukhopadhyay, "On the impact of energyaccuracy tradeoff in a digital cellular neural network for image processing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 7, pp. 1070–1081, Jul. 2015.

[6] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Dual-quality 4:2 Compressors for utilizing in dynamic accuracy configurable multipliers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 4, pp. 1352–1361, Apr. 2017.

[7] D. Mohapatra, V. K. Chippa, A. Raghunathan, and K. Roy, "Design of voltage-scalable meta-functions for approximate computing," in Proc. Design, Autom. Test Eur., Mar. 2011, pp. 1–6.

[8] Madanayake*et al.*, "Low-power VLSI architectures for DCT/DWT: Precision vs approximation for HD video, biomedical, and smart antenna applications," *IEEE Circuits Syst. Mag.*, vol. 15, no. 1, pp. 25–47, 1st Quart., 2015.

[9] S. Ghosh, D. Mohapatra, G. Karakonstantis, and K. Roy, "Voltage scalable high-speed robust hybrid arithmetic units using adaptive clocking," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 9, pp. 1301–1309, Sep. 2010.

## AUTHOR PROFILE

**Telagamalla Gopi** Received B-Tech degree from Scient Institute of Technology ,Hyderabad.Completed M-Tech with VLSI system design from Sree dattha Institute of Engineering and Science,Hyderabad.Currently working as Assistant Professor in Annamacharya Institute of Technology and Sciences,hyderabad.Has 8 years of teaching experience.Areas of interest includes digital signal processing and VLSI.