Journal of Engineering Sciences
Vol 11, Issue 6,June/2020
ISSN NO:0377-9254

# A PERFORMANCE EVALUATION OF MODIFIED SPECULATIVE APPROXIMATE ADDERS FOR 32-BIT

S.Sireesha[1], T. Lalith kumar[2]
[1]Dept. of ECE, Annamacharya Institute Of Technology & Science, Kadapa, Andhra Pradesh, India, sirisireeshareddy1@gmail.com..
[2]Dept. of ECE, Annamacharya Institute Of Technology & Science, Kadapa Andhra Pradesh, India, lalith.tappeta@gmail.com.

***ABSTRACT-*** **Approximate Adder require less area with that it have less propagation delay than perfect Adder. In this paper, we aimed full adder with one OR gate for approximate computing. Aimed edifice, segmenting the adder in few unintersection analogy cubes, in which the import output of the anterior cube is support by import input of the current summation cubes and those sub adders are ripple carry adders. In this ripple carry adder structure, carry estimator is used to estimate the import input of the modern cubes by anterior import output due to that delay will be reduced. From existing, measure will be for import chain calculated by aiming in to two cubes will produce the import output with fewer average delay. In this design, Full adder sum is constructed by 1or gate and one x-or gate and the import output is constructed by 2 AND gates and one OR gate. Due to this design the area as well as delay will be reduced. The effectives of the aimed style is weigh with the anterior style applied conditions of area and delay.**

***IndexTerms*** **— Approximate Computing, Less delay, Less area, Ripple Carry Adder(RCA).**

## 1. INTRODUCTION

As the electronics usages are probably incrementing day to day life. Humans are more focusing on the electronic devices, due to their flexibility which are the fast production in the industry. In electronics system the main entity of the internally is electronic components. The overview covered by the broad spectrum, through design and development of an electronic system to earn the new production by its proper function, service life and disposal. It's time to reveal the fusion of million MOS transistors into a single chips which was introduced in 1970's framed as the integrated circuit, processed by VLSI originated by Mohammed M. Atalla and Dawn Kalnng at Bell Labs(1959). Mainly the framed MOS transistors process Ic's are proposed in 1960 due to the MOS integrated circuit chip. Now a days electronics are the huge industry attained in a last few periods, due to fast progesses in very large scale designs and the applications are count at high which are related to the integrated circuits for high action of computing , controls, telecommunications and digital image processing rising fastly.

## 2. LITERATURE SURVEY

A. B. Kahng, S. Kang, et al.2012'[2] concluded to over whelm consequences of over erection, many recent apparatus for variation-resilient edifice permit timing errors and manage edifice reliability dynamically. Extremely othing the lack of correctness for erections may dramatically reduce costs of manufacturing, verification and test .

H. Esmaeilzadeh, A. Sampson, et al.2012'[3] concluded ago erectioning ASICs for the massive platform of quickly changing, normal-purpose applications is instantly impossible, practitioners are increasingly turning to programmable

M. Bilal, S. Masud, et al.2012'[5] proposed sum of squared error (SSE) and sum of abextremely lute difference (SAD) are two most outstanding error metrics for template sequencing in multimedia applications. Mathematically, the SSE and SAD mid two vectors "X" and "Y " are determined as follows:

$$SSE(X,Y) = X - Y = Mi=0\,Nj=0\,(Xij - Yij)\,2 \quad (1)$$

$$SAD(X,Y) = |X - Y| = Mi=0\,Nj=0\,|Xij - Yij| \quad (2)$$

Extremely "Xij " and "Yij " are the elements of 2-D M × N vectors "X" and "Y ," respectively, while (Xij − Yij ) is framed as the "error." Processing elements lack ofd for abextremely lute- and squared error determination are viewed in Fig. 1

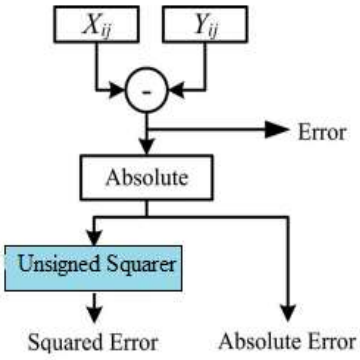


Fig. 1. extremely lute and squared error computations in hardware.

Y. Kim, Y. Zhang, et al. in 2013'[8] suggested Specifically, the key of several DSP and neuromorphic applications lies in processing particular kernel functions, which allows a powerful part of silicon area and computation time .

B. K. Mohanty, S. K. Patel, et al. in 2014'[4] suggested low power, area-efficient, and high-playance VLSI systems are increasingly utilized in handy and mobile devices, multistandard wireslighter receivers, and biomedical instrumentation.

M. Kamal, A. Ghasemazar, et al. in 2014'[6] suggested this, however, may be attained via the utilization of extensible process extremely rs which have emerged in the field of embedded computing as a promising arts. In this art, by incrementing the instruction set architecture (ISA) of the baseline embedded process extremely, the risky portions of the running application on the process extremely is accelerated.

H. Jiang, C. Liu, et al. in 2017'[1] proposed Mutually circuit erections, adders and multipliers have been a target because they play a vital task in calculating the action and power dissipation of countable compute-intensive applications.

## 3. EXISTING SYSTEM

The regular view of an speculative play for the approximate adder of n-bit improved the carry play by a carry estimator arm is showed in Fig. 3.1.

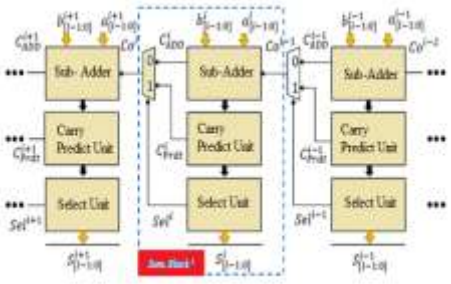


Fig.2 ordinary design of an approximate adder fusion with carry estimation arm.

In regular BBCSA (see Fig. 3.1), the carry output of the *ith* summation cube ( is obtained from

$$CO^i = \overline{Sel^i}.C^i_{Add} + Sel^i.C^{Ii}_{Prdt} \quad (7)$$

where the $Sel^i(C^i_{Prdt})$ determined by (Carry estimator)arm. Now, this work, we aimed to calculate the exact $Sel^i C^i_{Prdt}$ and $C^i_{Exact}$ using

$$Sel^i = K^{i+1}_0 + G^i_{l-1} \quad (8)$$

$$C^i_{Prdt} = G^i_{l-1} \quad (9)$$

$$C^i_{Add} = P^i_{l-1}G^i_{l-2} + P^i_{l-1}P^i_{l-2}G^i_{l-3} + \cdots + \prod_{k=1}^{l-1} P^i_k.G^i_0 \quad (10)$$

where the $K^{I+1}_0$ is the 1st bit position of the kill signal for the (*i*+1)th cube will be formed as(*i.e.*,$\overline{a^{i+1}_0}.\overline{b^{i+1}_0}$ ), $G^i_{l-1}$ is the produce signal to last bit position of the *ith* cube (*i.e.*$a^i_{l-1}.b^i_{l-1}$ ), and $P^i_{l-1}$ is the produce signal of the last bit position of the *ith* cube($a^i_{l-1} \oplus b^i_{l-1}$ ).

Thus the problem occurred in the ripple carry adder are deeply discussed in the 4 ways which are shown in below cases

- In the first case ($K^{i+1}_0$= 0 and $G^i_{l-1}$= 0), since $G^i_{l-1}$ = 0, the probability of this case will be half of the chance to reduce from the delay.

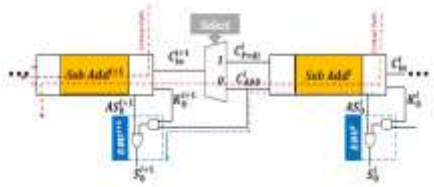

Fig. 3. pattern for existing adder with Error Recovery Arm (ERU**).**

- In the second case ($K^{i+1}_0$ =0 and $G^i_{l-1}$ =1), because $G^i_{l-1}$ is 1, the speculated carry ($C^i_{Prdt}$

) is correct. Therefore, for this case, the $C_{Prdt}^{i}$ is selected as the $C_{in}^{i+1}$.

- In the third case $(K_0^{i+1} = 1 \text{and } G_{i-1}^{i} = 0)$ because $K_0^{i+1}$ is 1, in this case have chance to the delay through the error occurance.
- In the final case $(K_0^{i+1} = 1 \text{ and } G_{i-1}^{i} = 1)$ equal to the middle case frame as the second option, by this no chance to occurance of the delay.

An error recovery arm produces the first addition bit for *ith* cube $S_0^i$ as

$$S_0^{i+1} = \left(K_0^{I+1}.C_{Add}^{i}\right) + \left(P_0^{i+1} \oplus C_{in}^{i+1}\right) \qquad (11)$$

Note that $P_0^{i+1} \oplus C_{in}^{i+1} (C_{in}^{i+1} = C_{Prdt}^{i})$ from approximate addition output of 1st bit of the (*i*+1)*th* cube denoted as *A* in Fig. 3.

## 4. PROPOSED SYSTEM

### 4.1. Full adder In RCA:

If the full adder two inputs then the carry input cine ill acts as the third input. The output carry is identical as cout and the ordinary output is identical as view the truth-table

Table 1: Utility table of full adder

| Inputs | | | Outputs | |
|---|---|---|---|---|
| A | B | Cin | Sum | Carry |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

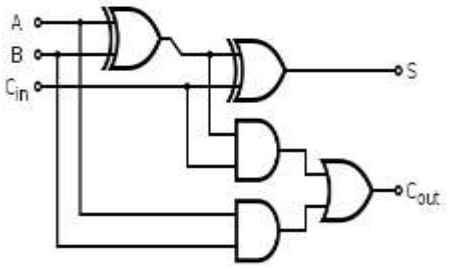


Figure 4. full adder logic diagram

### 4.2 Full adder with OR In RCA:

In this case ,we are placing 32-bit Rca. By the requirement of the or gate instead of the xor gate will reduce the delay and the area.
The proposed block diagram is shown in figure 5.Here the proposed work can be implemented in as follows.

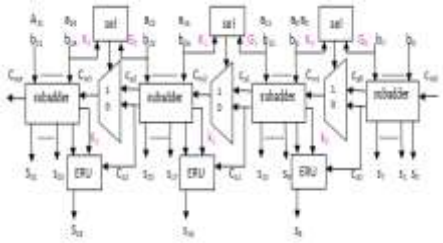

Fig.5 Generalized proposed block diagram.

Generally, RCA implemented with FA which is shown in figure 6.In full adder, sum and carry expressions are

$$Sum = A \; xor \; B \; xor \; C$$
$$Carry = (A \; and \; B) or (B \; and \; C) or (C \; and \; A)$$

For approximation we replace one of the xor gate in sum with or gate, viewed in figure 4.2 the sum equation becomes

$$Sum = A \; or \; B \; xor \; C$$

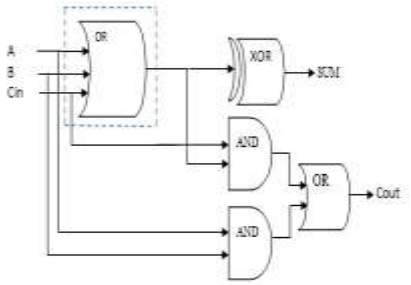


Fig.6 Full Adder with or gate

By replacing the blue dotted line with OR gate. Through this structure the area as well as delay will be decreases.

Table 2: utility table of full adder with or gate

| Inputs | | | Outputs | |
|---|---|---|---|---|
| A | B | Cin | Sum | Carryout |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

## 5. RESULTS:

Table 3:comparision of adders

| Parameters | exisiting32-bit | Proposed 32-bit |
|---|---|---|
| Area | 114 | 60 |
| Delay | 12.684ns | 8.602ns |

### 5.1 MATLAB APPLICATION

The images of the x-rays are not provide in the single image due to the area and the delay consumption. The fusion of the images are take place in this due to the area and delay complexities are not achieved in this images. the purpose of the image fusion will reduce the data of that cost and built the images with appropriate results those are understandable by the human as well as machines.

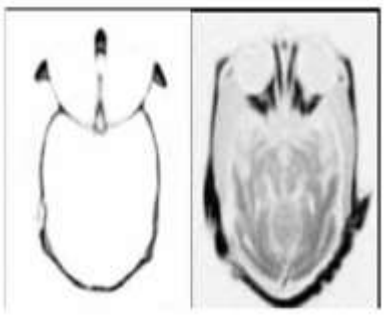

Fig.7 Digital image application Input Images of proposed method .

Fig.8 Fused digital Image application which is the output image of the proposed method.

## 6. CONCLUSION

Approximate Adder require less area and have less propagation delay than exact Adder. In this paper, we propose full adder with one OR gate for approximate computing. In this ripple carry adder structure, carry predictor is used to predict the carry input of the present blocks by the previous carry output due to that delay will be reduced which was occurred in RCA. In this design, Full adder sum is constructed with one or gate and one x-or gate. Due to this design the area as well as delay will be reduced. The effectives of the proposed method is concluded with the previous method in terms of area and delay by the fused images of digital image processing applications.

## REFERENCES

[1] H. Jiang, C. Liu, L. Liu, F. Lombardi and J. Han, “A review, classification and comparative evaluation of approximate arithmetic circuits,” ACM JETCAS, vol. 13, no. 4, Article no. 60, 2017.

[2] A. B. Kahng, S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," In Proc. DAC, 2012, pp.820-825.
[3] H. Esmaeilzadeh, A. Sampson, L. Ceze, and D.Burger. "Neural Acceleration for General-Purpose Approximate Programs," In Proc. of Micro, pp.105-115, 2012.
[4] B. K. Mohanty, S. K. Patel, "Area–delay–power efficient carry select adder", IEEE TCAS-II, vol. 61, no. 6, pp. 418-422, Jun. 2014.
[5] M. Bilal, S. Masud, and S. Athar, "FPGA Design for Statistics-Inspired Approximate Sum-of-Squared-Error Computation in Multimedia Applications," IEEE TCAS-II, vol. 59, no. 8, pp. 506-510, 2012.
[6] M. Kamal, A. Ghasemazar, A. Afzali-Kusha, and M. Pedram, "Improving efficiency of extensible processors by using approximate custom instructions," in Proc. DATE, 2014.
[7] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "RAP-CLA: A reconfigurable approximate carry look-ahead adder," IEEE TCAS-II, vol. 65, no. 8, pp. 1089–1093, 2018.
[8] Y. Kim, Y. Zhang, and P. Li. "An energy efficient approximate adder with carry skip for error resilient neuromorphic VLSI systems," In Proc. ICCAD, 2013, pp. 130–137.
[9] J. Hu and W. Qian, "A new approximate adder with low relative error and correct sign calculation," In Proc. IEEE DATE, pp. 1449-1454, 2015.
[10] W. Xu, S. S. Sapatnekar, and J. Hu. "A Simple Yet Efficient Accuracy Configurable Adder Design," In Proc. ISLPED, 2017.
[11] R. Ye, T. Wang, F. Yuan, R. Kumar and Q. Xu, "On Reconfiguration-Oriented Approximate Adder Design and Its Application," In Proc. ICCAD, 2013, pp. 48-54.
[12] M. Shafique, W. Ahmad, R. Hafiz and J. Henkel, "A low latency generic accuracy configurable adder," In Proc. DAC, 2015, pp. 1-6.