J Electron Test (2016) 32:377–383
DOI 10.1007/s10836-016-5587-z

High Performance Significance Approximation Error Tolerance Adder for Image Processing Applications

R. Jothin[1] · **C. Vasanthanayaki**[2]

Received: 1 March 2016 / Accepted: 21 April 2016 / Published online: 29 April 2016

Abstract Addition is one of the fundamental arithmetic operations which are used extensively in many VLSI systems such as microprocessors and application specific DSP architectures. In this paper, the Significance Approximation Error Tolerant Carry Select Adder (SAET-CSLA) is constructed, which is efficient in terms of accuracy, power and area. While considering the elementary structure of an image processing applications, it is a combination of the multipliers and delays, which in turn are the combination of the adders. This research paper describes the Algorithmic strength reduction technique which leads to a reduction in hardware complexity by exploiting the significances. This transformation is basically implemented for the reduction in the power consumption and area efficient of Very Large Scale Integration (VLSI) design or iteration period in a programmable Digital Signal Processing (DSP) implementation. The significance approximation error tolerant adder is designed using full adder and approximate full adder cells with reduced complexity at the gate level. The performance of 16 bit conventional Carry Select Adder (CSLA), 16 bit Error Tolerant carry select Adder (ET-CSLA) and proposed Significance Approximation Error Tolerant Carry Select Adder (SAET-CSLA) are compared. For all the 2^{16} input combinations, comparison is made between existing and proposed CSLA adders and the error tolerance analysis is carried out for accuracy improvement. Application of image processing is carried out using proposed SAET-CSLA.

Keywords Carry select adder · Significance approximation · Error tolerance · Digital signal processing · Image blending

Responsible Editor: V. D. Agrawal

✉ R. Jothin
joejothin@gmail.com

[1] Department of Electronics and Communication Engineering, KGiSL Institute of Technology, Coimbatore, India

[2] Department of Electronics and Communication Engineering, Government College of Technology, Coimbatore, India

1 Introduction

In Digital signal processing, the filter is used to remove some unwanted component or feature from a signal thereby improving the quality of signal [14]. It alters the amplitude and/or phase characteristics of a signal in a desired manner with respect to its frequency. The primary function of the filters is to confine a signal into a prescribed frequency band, to decompose a signal into two or more sub-bands, to modify the frequency spectrum of a signal and to model the input-output relationship of a system [9]. Filters are extensively used in the signal processing and communication system in the applications such as noise reduction, echo cancellation, speech and waveform synthesis etc.

Approximation adders are widely used in the different types of the digital filter applications. In some of the applications, the conventional adder circuit is used to operate at the high accuracy rates, whereas in image processing applications, an approximation adder circuit is used in the low-power circuit, operating at a high speed for moderate accuracy rates. The approximation techniques were developed specifically for the digital image processing that can be found in parallel (or block) processing to either increase the effective throughput or to reduce the power consumption of the original image processing systems [7].

Traditionally, the application of the high quantity digital data processing system involves more number of same architectures of approximation adder hardware units for parallel

Table 1 Truth table for full adder

Inputs			Conventional FA	
A	*B*	*C*	*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

processing implementation. Choosing the adders with more number of hardware breadth rather than the depth not only reduces the delay, but also the total power consumption. A lot of the design methods of the low power approximation adders were proposed in the literatures [5, 16]. Some applications need an approximation adder to operate at the high frequencies such as video processing, whereas some other applications require high throughput and accuracy with a low-power circuit such as multiple-input and multiple-output systems [3, 10].

The content of this paper is organized as follows. Section II discusses about the existing conventional CSLA and ET-CSLA. Section III explains the proposed Significance Approximation Error Tolerant Carry Select Adder. Section IV presents comparison of simulation results and Section V deals with Image processing application using proposed SAET-CSLA. Finally, conclusions are drawn in Section VI.

2 Existing Conventional 16 Bit CSLA and 16 Bit Et-CSLA Adders Theory

2.1 Conventional 16 Bit CSLA

A carry-select adder is divided into sectors, each of which except for the least-significant performs two additions in parallel, one assuming a carry-in of zero, the other a carry-in of one. A four bit carry select adder generally consists of two ripple carry adders and a multiplexer. The carry-select adder is simple but rather fast, having a gate level depths of adding two

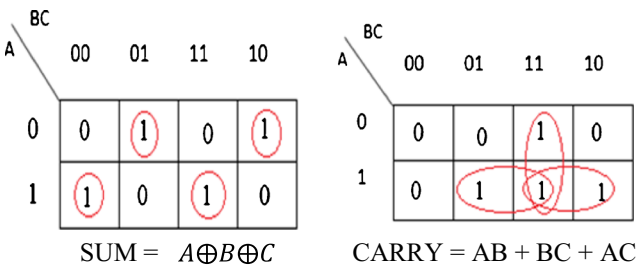


Fig. 1 K-map minimization of full adder

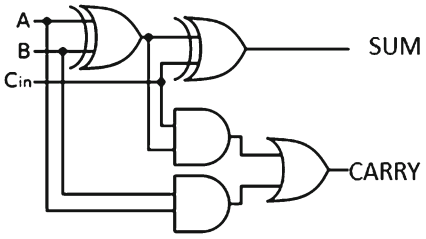


Fig. 2 Logic implementation of full adder

n-bit numbers with a carry select adder is done with two adders (two ripple carry adders) in order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one [1]. After the two results are calculated, the correct sum, as well as the correct carry, is selected with the multiplexer once the correct carry is known. A carry-select adder speeds 40 % to 90 % faster than RCA by performing additions in parallel and reducing the maximum carry path. The Full Adder (FA) is useful for additions that have multiple bits in each of its operands. It takes in three inputs and creates two outputs, a sum and a carryout [6, 8, 11, 14]. Full adder logic has 13 basic logic gates (AND, OR and NOT gates), which are used to implement the primarily gate level structure (2 Exor gates (2*5) +3) and 6 logic delays. The inputs have the same weight 2^i, the sum output has a weight of 2^i, and the carryout output has a weight of $2^{(i+1)}$ [12, 13].

The truth table and k map minimization of full adder is shown Table 1 and Fig. 1 respectively. The minimized expression for the implementation is obtained from k map. The logical gate implementation of sum and carry expression is shown in Fig. 2. Using FA truth table (Table 1), further comparisons with approximation logics are done.

For 16 bit conventional carry select adder shown in Figs. 3, 424 basic logic gates are used. In 16 Bit dual ripple carry adder it requires (7 RCA* 4FA in each RCA) = 7* 4*13 = 364 gates and for 3 numbers of 10:5 multiplexer (3* 5*(2:1 mux)), it requires (3*5*4) =60 gates.

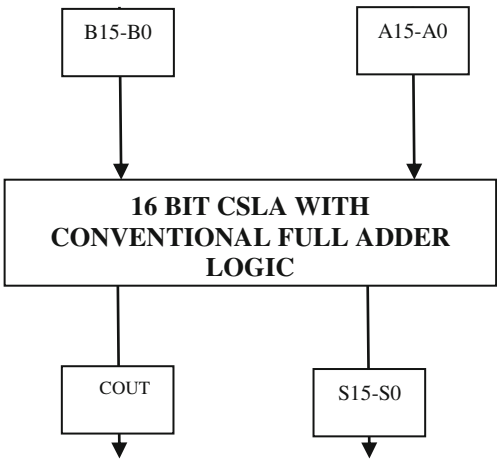


Fig. 3 Conventional 16 bit CSLA

Table 2 Truth table for approximation full adder

Inputs			Approximation FA	
A	*B*	*C*	*SUM*	*CARRY*
0	0	0	1 (Error)	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	0 (Error)	1

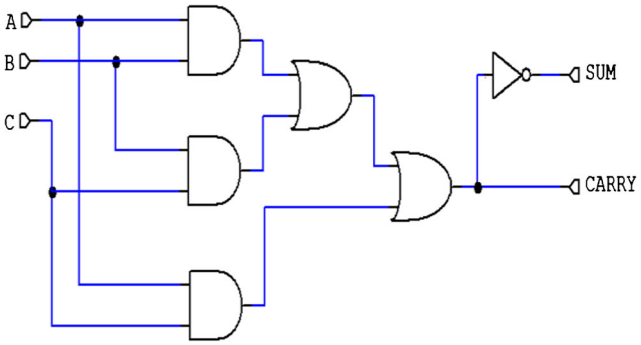


Fig. 5 Logic implementation of approximation full adder

2.2 16 Bit Error Tolerant CSLA

Approximate adders are adders with conventional architectures which run in an approximate output. With this output, erroneous sums may be produced at the savings of energy required to execute the computation. The result presented in this report lead to procedure for allocating the available energy budgets among the adders modules so as to minimize the expected error. For simplicity only the uniform distribution of the input is considered [5].The exact magnitude of error is dependent on time at which the outputs are sampled and the propagation time from the approximate adder to the particular output. But the quantity are interested in the significance of the approximate adder which is the amount by which the error at the approximate adder is amplified (or reduced) when it has propagated to the output [15].

Approximate hardware provides a result faster than an exact unit, and has applications to design at both ends of the performance spectrum. A low performance design might accept the potential errors, or be tolerant of them to some threshold; a high performance design can use the approximate result to speculatively execute instead of ideally waiting for the exact result to become available [1]. Throughput performance was more sensitive to correctness than the speedup in machine cycles of an approximate unit compared to an exact unit. Thus it is improving the correctness of the approximate arithmetic units, and extending them to more frequent and longer latency floating point operations.

In the approximation logic, approximation is chosen since it gives minimum errors when compared to other approximations [1]. It shows only 2 errors in Sum and no error in carry as shown in Table 2. K map minimization for approximation Full adder gate level implementation is done and represented in Fig. 4. Approximation Full adder logic has 2 errors in sum output and no error in carry output, where all the input bits are either ‘0’s or ‘1’s. Whereas for all other sum and carry values, there is no change in outputs.

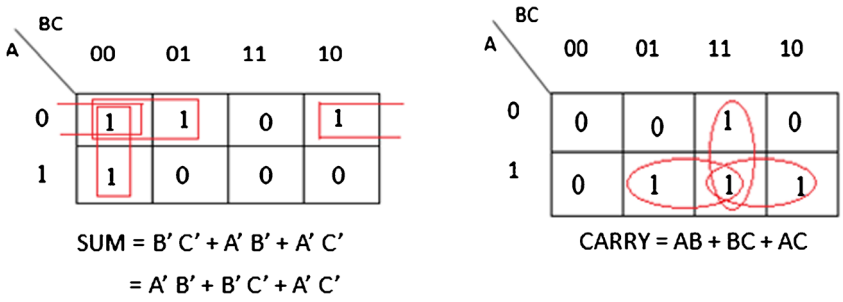
Logic implementation of Approximation full adder shown in Fig. 5, has 6 basic logic gates and 4 delays instead of 13 basic logic gates and 6 delays of conventional full adder.

Block diagram for 16 bit Error Tolerant Adder is shown in Fig. 6. In this approximate adder, approximation logic of full adder is applied for each block of Ripple Carry Adder. When the approximation is applied, only the two combinations (000 or 111) of inputs result in error, all the remaining combinations are correct.

For the ET-CSLA, when all the inputs are 1 then sum is an error (i.e., instead of 1 sum is 0) and when all the inputs are 0 then sum is an error (i.e., instead of 0 sum is 1). The ET-CSLA has 228 gates which are considerably less than conventional carry select adder. In 16 Bit dual ripple carry adder in the ET-CSLA, it requires (7 RCA* 4 AFA in each RCA) 7*4*6 = 168 gates and for multiplexer, it requires 60 gates.

Hence an area reduction of 46.22 % is obtained for Error Tolerant Carry Select adder when compared with conventional carry select adder.

Fig. 4 K-map minimization of approximation full adder



3 Proposed Significance Approximation Error Tolerant Carry Select Adder

Error-tolerance deals with the use of defective circuitry that occasionally produces errors, yet provides acceptable performance to end users when executing certain applications.

The motivation for using such devices is the related increase in effective yield, and hence lower cost parts [2].

This proposed significance approximation error tolerance concept shown in Fig. 7, is an integration of conventional CSLA and ET- CSLA for the system which improves the accuracy of output results. However, for many digital signal processing (DSP) systems that process signals relating to human senses such as hearing, sight, smell, and touch, e.g., the image processing and speech processing systems, the error-tolerant circuits may be applicable [4, 5, 16].

The different terminologies used in this work are as follows:

Overall error, OE = Difference between Rc and RE, where RE, is the result obtained by the adder and Rc denotes the correct result. Percentage of error Tolerance = [OE/Rc]/ 100 % (all the results are represented as decimal numbers) [5].

Accuracy (ACC): In the scenario of the error tolerant design, the accuracy of an adder is used to indicate how "correct" the output of an adder for a particular input. It is defined as: ACC = (1-(OE/Rc))/100 %. Its value ranges from 0 to 100 %. Minimum Acceptable Accuracy (MAA): Although some errors are allowed to exist at the output of an ETA, the accuracy of an acceptable output should be "high enough" (higher than a threshold value) to meet the requirement of the whole system. Minimum Accuracy of an adder is higher than the minimum acceptable accuracy is called acceptable result.

Acceptance Probability (AP): Acceptance probability is the probability that the accuracy of an adder is higher than the minimum acceptable accuracy.

The dividing strategy is chosen to design proposed SAET-CSLA with area efficiency and minimum acceptable accuracy to be 99 %. Having considered the above, the 16 bit adder is divided by 8 bit in to the accurate part and 8 bit in to the inaccurate part. If the accurate part has less than 8 bit, the minimum acceptable accuracy and the acceptance probability will be decreased. If the accurate part has more than 8 bit, the minimum acceptable accuracy and the acceptance probability will be 99 % and area will be inefficient. The addition of the higher order bits (accurate part) of the input operands is performed from LSB to MSB and normal addition method is applied.

3.1 Accurate Part

In this part, the normal 8 bit conventional Carry select adder is applied to preserve its correctness since the higher order bits play an important role than lower order bits for better accuracy.

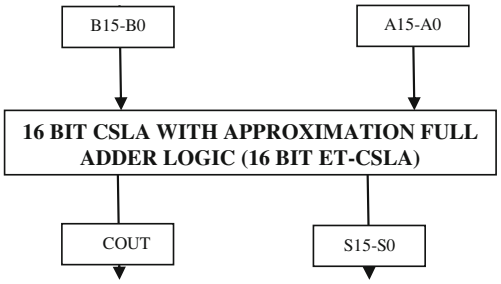


Fig. 6 16 bit error tolerant CSLA

3.2 Inaccurate Part

In this part, 8 bit ET-Carry select adder is chosen for inaccurate part of circuit since it is simplest one for most power saving and requires less hardware circuitry. The proposed SAET-CSLA adder has 340 gates which are considerably fewer than conventional CSLA and slightly more than ET-CSLA. In 8 Bit dual ripple carry accurate part, adder requires (4 RCA* 4FA in each RCA) 4*4*13 = 208 gates, inaccurate part of adder requires (3 RCA* 4 AFA in each RCA) 3*4*6 = 72 and for multiplexer it requires 60 gates.

Hence an area reduction of 19.81 % is obtained for proposed SAET-CSLA adder when compared with conventional CSLA.

3.3 Accuracy Calculation

The addition should actually yield 65,528 from two operands 32,761 and 32,767 if normal arithmetic has been applied. The

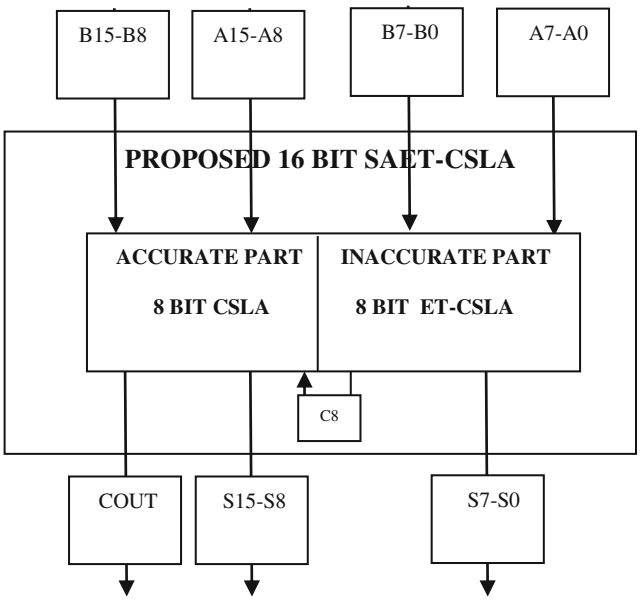


Fig. 7 16 bit proposed SAET-CSLA

Table 3 Error tolerance comparision of adders

S. No	Input data 1	Input data 2	Sum		% of error in 16 bit ET-CSLA	Range of error in 16 bit ET-CSLA	Sum	% of error in proposed 16 bit SAET-CSLA	Range of error in proposed 16 bit SAET-CSLA
			16 bit conventional adder	16 bit ET-CSLA			Proposed 16 bit SAET-CSLA		
1	23,445	18,330	41,785	41,057	1.74	Minimum	41,825	0.095	Minimum
2	18,429	24,510	42,999	40,967	4.74		42,755	0.006	
3	23,441	17,690	41,131	41,199	0.0016		41,101	0.001	
4	23,424	16,410	39,834	49,151	23.38	Medium	39,935	0.003	
5	16,413	22,530	38,943	49,151	26.2		39,167	0.006	
6	20,480	30,974	51,454	36,863	28.3		51,455	0.000	
7	22,528	28,890	51,418	31,863	38.03	Maximum	51,455	0.001	
8	23,447	29,978	53,425	32,993	38.24		53,473	0.089	
9	23,545	32,026	55,571	32,775	41.02		55,559	0.021	
10	24,477	32,030	56,507	32,995	41.06		56,547	0.070	
11	32,663	30,490	63,153	32,993	47.75		63,201	0.001	
12	30,962	32,707	63,669	32,829	48.4		63,549	0.002	
13	32,761	32,767	65,528	32,768	49.9		65,280	0.004	

total error generated can be computed by the following equation [1]:

$$\text{Overall Error (OE)} = |\text{Rc} - \text{Re}|$$

Where, Re (=65280) is the result obtain from SAET-CSLA adder and Rc denotes the correct result.

$$\text{OE} = |65528 - 65280| = 248$$

$$\text{OE Tolerance} = (248/65528) = 0.0037$$

$$\text{Accuracy} = (1-\text{OE}/\text{Rc}) \times 100 = (1-248/65528) \times 100 = 99.927\%$$

(Accuracy of an adder with respect to the two input operands).

4 Simulation Results Comparision

From Tables 3 and 4, it is obvious that the error tolerance performance of the adder for all input operands are greatly improved using SAET-CSLA. The minimum accuracy range can be rising from 50 % to 99 %. Although SAET-CSLA achieves high performances in accuracy for all input operands ranges from 0 to 2^{16}–1 and consumes slightly more energy than ET-CSLA. In this proposed design, the higher range of input operands could be playing a more important role to represent the high intensity digital signal. Therefore all ranges of input operands are considered while calculating the accuracy in SAET-CSLA. But the existing ET-CSLA is nearly 50 % less accurate for the higher range of input operands.

5 Application of Proposed SAET-CSLA in Image Processing

Image blending operation is performed based on image addition principle, but different weights are to be given for images so that it gives a feeling of blending or transparency. Images are added as per the equation below:

$$G(x,y) = (1-\alpha)F_1(x,y) + \alpha F_2(x,y)$$

Where, α is the blending ratio which determines the influence of each input image in the output. By varying the values of α from 0 to 1, a cool transition between one image to another is performed. This operator forms a blend of two input images of the same size. Similar to pixel addition, the value of each pixel in the output image is a linear combination of the

Table 4 Comparision of adders

Word size	Adder	Area (gate count)	Delay (ns)	Power (mw)
16 bit	Conventional CSLA	424	24.686	87
	ET-CSLA	228	18.851	79
	Proposed SAET-CSLA	340	22.187	84

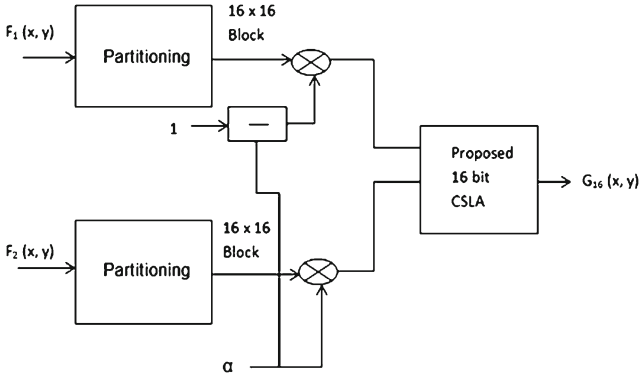


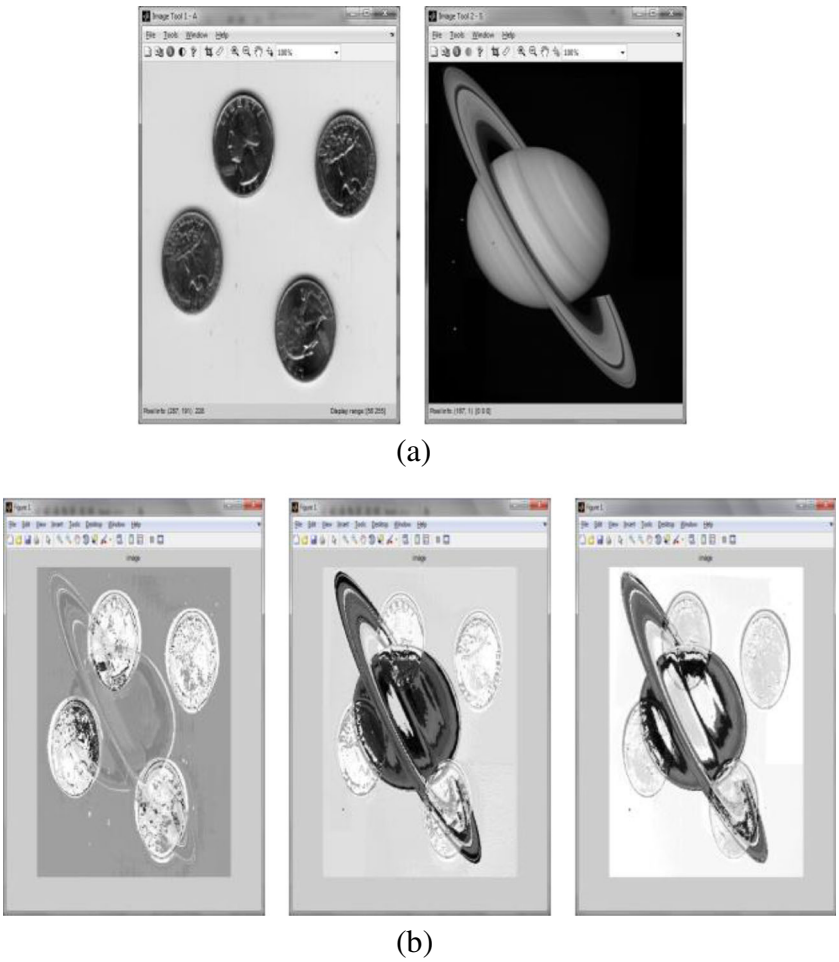
Fig. 8 Architecture of image blending using proposed 16 bit SAET – CSLA

corresponding pixel values in the input images. The coefficients of the linear combination are user-specified and they define the ratio by which to scale each image before combining them. These proportions are applied such that the output pixel values do not exceed the maximum pixel value. F_1 and F_2 are the two input images. In some applications F_2 can also be a constant, thus allowing a constant offset value to be added to a single image. α can either be a constant factor for all pixels in the image or can be determined for each pixel separately using a mask. The size of the mask must then be identical with the size of the images.

Architecture of image blending is shown in Fig. 8. F1 and F2 are the two input images (512 × 512) are separately given to two different partition blocks. The partitioned images are further multiplied by a multiplication factor. One of the images is multiplied by the factor α and other by the factor 1-α. The multiplied images are then added with the proposed 16 bit SAET-CSLA. α is the variable which determines the measure of blending of output resolution.

Image blending application is performed using proposed SAET-CSLA and the input and outputs are shown in Fig. 9. Verilog code is used to construct the adders and the image blending operation is performed in MATLAB using proposed SAET-CSLA by varying the intensity factor to get the high accuracy images.

Variation in α alters the foreground and background pixels of blended image parameters such as sharpness, brightness and contrast. From the Fig. 9b, it is noted that increasing the value of α enhances the quality of the blended image.



(a)

(b)

Fig. 9 Image blending **a** images F1 and F2 **b** blended images for α = 0.2, 0.6 and 0.8

6 Conclusion

SAET-CSLA adder structure is applicable to general purpose design, with a few exceptions. The need for the instant response and increasingly the huge data sets, the adder should be high accuracy and fast. In this proposed adder, minimum error tolerance is obtained by conventional full adder logic in the accurate part in order to increase the accuracy at the most significance level. The error tolerant adder is designed using approximate full adder cells with reduced complexity at the gate level in the inaccurate part. 16 bit conventional CSLA, ET-CSLA and proposed 16 bit SAET-CSLA are implemented. For all possible 2^{16} input combinations, performance analysis is done and the average error computed is less than 1 %. The error of the proposed adder can be reduced by analyzing the relation between the input bit patterns and approximation logic. The SAET-CSLA design can be a potential solution for all range of input operands to get the high accuracy resolution image outputs for the image processing applications. But by sacrificing some average accuracy more than 30 %, and the minimum accuracy level is 50 % for high input operands the ET-CSLA can attain an improvement in both the power consumption as well as speed and area efficient performance.

References

1. Gupta V, Mohapatra D, Park SP, Raghunathan A, Roy K (2011) IMPACT: imprecise adders for low-power approximate computing. Proc IEEE/ACM Int Symp Low-Power Electron Des 409–414
2. He Y, Chang CH, Gu J (2005) An area efficient 64-bit square root carry-select adder for low power applications. Proc IEEE Int Symp Circ S 4:4082–4085
3. Hegde R, Shanbhag N (1999) Energy-efficient signal processing via algorithmic noise-tolerance. Proc IEEE/ACM Int Symp Low Power Electron Des 30–35
4. Jayanthi AN, Ravichandran CS (2012) Design of error tolerant adder. Am J Appl Sci 9(6):818–824
5. V Muralidharan, M Jagadeeswari (2012) An enhanced Carry elimination adder for low power VLSI applications. Int J Eng Res Appl 2(2) 1477–1482
6. Neeharika R, Venkanna, Kavitha M (2012) Design of low-power high-speed truncation-error tolerant adder and its application in digital signal processing. Int Eng Res Appl 2(2): 939–944
7. Parhi KK (2007) VLSI digital signal processing systems: design and implementation, 255–279, Wiley
8. Rawat K, Darwish T, Bayoumi M (2002) A low power and reduced area Carry select adder. Proc 45th Midwest Symp Circuit 1:467–470
9. Saeed V (2007) Vaseghi. Wiley, Multimedia Signal Processing
10. Saxena P, Purohit U, Joshi P (2013) Analysis of low power, area-efficient and high speed fast adder. Int J Adv Res Comput Commun Eng 2(9):356–375
11. Shim B, Sridhara S, Shanbhag N (2004) Reliable low-power digital signal processing via reduced precision redundancy. Proc IEEE Trans VeryLarge Scale Integr Syst 12(no. 5):497–510
12. Singh S, Kumar D (2011) Design of area and power efficient modified carry select adder. Int J Comput Appl 33(No. 3):975–987
13. Varatkar G, Shanbhag N (2006) Energy-efficient motion estimation using error-tolerance. Proc IEEE/ACM Int Symp Low Power Electron Des 113–118
14. Yadav S, Mehra R (2014) Analysis of FPGA based recursive filter using optimization techniques for high throughput. Int J Adv Res Technol 3:341–343
15. Kim Y, Kim L-S (2001) A low power carry select adder with reduced area. Proc IEEE Int Symp Circ S 4:218–221
16. Zhu N, Goh WL, Yeo KS (2009) An enhanced low-power highspeed adder for error-tolerant application. Proc IEEE Int Symp Integr Circuits 69–72

R. Jothin received Diploma in Electronics and Communication engineering from the Sankar Institute of Polytechnic, Tirunelveli, India and received the B.E. and M.E. "(VLSI Design)" in Electronics and Communication engineering from Government College of Technology, Coimbatore, India. He is currently doing a Ph.D. in Anna University, Chennai, India. He has 20 years of industry experience and 3 years of academic experience. His research is focussed on High performance VLSI architectures for image processing applications.

C. Vasanthanayaki received the B.E. degree in electronics and communication engineering and M.E. degree in computer science, from Government college of Engineering, Tirunelveli, India, in 1987 and 1997 respectively. She received the Ph.D. degree in the information and communication engineering. Since 1988, she has been with the Government College of Technology, Coimbatore, India.