



AREA-EFFICIENT PARALLEL ADDER WITH FAITHFUL APPROXIMATION FOR IMAGE AND SIGNAL PROCESSING APPLICATIONS

V. KARUNAKARAN¹, C. SENTHILKUMAR, M.E., DOEACC²

1 PG SCHOLAR, 2 ASSISTANT PROFESSOR

MASTER OF ENGINEERING IN VLSI DESIGN

AVS ENGINEERING COLLEGE, SALEM, TAMILNADU, INDIA

ABSTRACT

Design of low-power and area-efficient portable complementary metal– oxide–semiconductor processors for image and signal processing applications demand reduction in transistor switching and count. Adder is the fundamental block of all arithmetic operations performed in processing units. In this study, an error-tolerant parallel adder with faithful approximation is proposed that can optimize area and accuracy. In the proposed parallel adder, for n bit input and m bit adder block, least $n/2m$ blocks are designed with approximate logic using carry by-pass addition algorithm and most $n/2m$ blocks are designed with exact logic using carry select addition algorithm. Least significant approximate part of the adder is designed with either exact full adder (EFA) or fault-tolerant full adder (FTFA) cells. This confines the maximum error in the proposed-EFA and proposed-FTFA designs to be not more than unit bit value with weights $2[(n/2m) - 1]m$ and $2n/2$, respectively. Two different FTFA cells are proposed and implemented in the approximate blocks.

KEYWORDS: Parallel Adder, Human Senses & Digital Image

1. INTRODUCTION

Addition is the fundamental operation in arithmetic units used in digital image processing, signal processing, multimedia processors and so on. All these applications demand multi-bit adders and high delay in carry propagation is a great issue while using long ripple carry adder (RCA). A fast and reliable operation in digital system is performed by resident parallel adders. Parallel adders are preferred for VLSI implementation since they rely on the use of simple cells and possess regular connections between them. The ripple-carry adder (RCA) is the simplest form of adder. Two numbers using two's-complement representations can be added by using the circuit shown in Figure. A Wd -bit RCA is built by connecting Wd full-adders so that the carry-out from each full-adder is the carry-in to the next stage. The sum and carry bits are generated sequentially, starting from the LSB. The carry-in bit into the rightmost full-adder, corresponding to the LSB, is set to zero, i.e., $(c_{Wd} = 0)$. The speed of the RCA is determined by the carry propagation time which is of order $O(Wd)$. Special circuit realization of the full-adders with fast carry generation is often employed to speed the operation. Pipelining can also be used. Multiple full adder circuits can be cascaded in parallel to add an N -bit number. For an N -bit parallel adder, there must be N number of full adder circuits. A ripple carry adder is a logic circuit in which the carry-out of each full adder is the carry in of the succeeding next most significant full adder. It is called a ripple carry adder because each carry bit gets rippled into the next stage. In a ripple carry adder the sum and carry out bits of any half adder stage is not valid until the carry in of that stage occurs.

Though many approaches are proposed in literature to improve speed in parallel addition, it compromises area at the other end. Approximate computing is a recent technique which gained much attention in the design of arithmetic data path units for error resilient applications. Image and video processing applications and application circuits involving processing of human sensing signals like hearing, touch and smell mostly employ approximate circuits since small error in the output does not affect the visual quality or hearing sense. The significant advantage of approximation logic is that they realize area- and energy-efficient architectures suitable for portable battery-powered devices. Error-tolerant adders with approximation logic in least significant inaccurate part are proposed in.

Another approach for multi-bit addition uses block-based architecture. In block-based adders, the Sum bit at any significant position is estimated from approximate carry bits computed from immediate least significant bit (LSB) inputs. Examples of such type of adders are almost correct adder, error-tolerant adder, carry skip Adder and carry save approximate adder. The block based adders demonstrate better in terms of speed compared to approximate LSB adders. Speculative adders proposed in exhibit reduced critical path delays to sub- logarithmic level by exploiting the trade-off between reliability and performance. Speculative adders combine speculation with error correction to achieve high accuracy and low area overhead over traditional approximate designs.

2. LITERATURE REVIEW

2.1 Jothin, R., Vasanthanayaki, C.: 'High speed energy efficient static segment adder for approximate computing applications', *J. Electron. Test.*, 2017, 33, (1), pp. 125–132

Real time high quantity digital data computing design needs to achieve high performance with required accuracy range. The constraints involved with high performance are low power consumption, area efficiency and high speed. This paper proposes a design of high speed energy efficient Static Segment Adder (SSA), which improves the overall performance based on static segmentation. Accuracy Adjustment Logic (AAL) is incorporated to improve the accuracy derived from negating lower order bytes of input operands. In this paper, an integration of static segment method and accuracy adjustment logic is used to achieve computational accuracy for error tolerant applications. The proposed adder design enables to provide high speed and energy efficiency through the static segmentation method. Image enhancement operation is carried out using proposed SSA design. In this method, 99.4% overall computational accuracy for 16-bit addition even with 8-bit adder can be achieved.

In many Very Large-Scale Integrated (VLSI) systems such as application- specific Digital Signal Processing (DSP), the circuit is implemented for filtering, encryption or time to frequency or frequency to time domain transformations. Most critical functional units of these architectures, which performance is totally depended upon are adders. If adders are too slow or consume more energy, then the performance of the design will be degraded. Approximate addition has been carried out as a means of achieving area, power and speed improvements at the cost of accuracy in the field of digital video and audio signal processor design.

2.2 Hafiz Md. Hasan Babu and Ahsan Raja Chowdhury, "Design of a Reversible Binary Coded Decimal Adder by Using Reversible 4-Bit Parallel Adder", 18th International Conference on VLSI Design (VLSI Design 2015), January 2015, Kolkata, India, pp-255-260.

Different stages of Brent -Kung and Kogge-Stone adders. The proposed style reduces of prefix operation by victimization a lot of number of Brent-Kung stages that reduces the quality, semiconductor space and power consumption significantly. The operation undergone during this technique is that once high operation speed is required, tree structures like parallel-prefix adders square measure used. The strategy employed in this technique is Parallel Prefix Adder in Associate in Nursing FPGA Perform computation that any previous state can perpetually be reconstructed given an outline of the present state. Simulation results of forward & backward computation of 4*4 reversible TSG & Fred kin gate. The gate is then went to design four-bit Carry Skip Adder block. The adder design designed victimization TSG & Fredkin gate square measure abundant optimized as compared to existing four bit Carry Skip Adder in terms of low power dissipation. Methodology used for coming up with reversible gate is Tanner Tool Version- 13 & technology file zero.35 microns. The operation that undergone is that the method of carry- skip adder (also familiar as a carry- bypass adder) is adder implementation that improves on the delay of a ripple- carry adder. The strategy employed in this method is Carry skip adder victimization TSG & Fred kin reversible gate.

2.3 Liu, C., Han, J., Lombardi, F.: ‘An analytical framework for evaluating the error characteristics of approximate adders’, *IEEE Trans. Comput.*, 2019, 64, (5), pp. 1268–1281.

Reversible logic is gaining important thought because the potential logic style for implementation in fashionable engineering and quantum computing. They are implemented with marginal impact on physical entropy. A unique programmable reversible computer circuit is conferred, verified and its implementation within the style of a reversible ALU is incontestable. Implementations of the Kogge-Stone adder with sparsity-4, eight and sixteen were designed, verified and compared. The improved sparsity-4. Kogge-Stone adder with ripple-carry adders was selected and its enforced within the design of a 32-bit ALU is incontestable. Similar to the carry-skip adder, however computes generate signals additionally as cluster propagate signals to avoid looking ahead to a ripple to see if the cluster generates a carry. The strategy used is increased carry look-ahead adder for novel reversible ALU.

3. EXISTING METHOD

For multi-bit addition, conventional RCA is more suitable due to low hardware area and regular structural arrangement of FA cells; however, it incurs high propagation delay. RCA takes in two n -bit operands as input and produces $n + 1$ -bit result and is constructed using n FA cells which are cascaded together, with the carry-out bit of one FA tied to the carry-in bit of the next FA cell. Hence, the propagation delay of RCA represented as t_{RCA} will be,

$$t_{RCA} = n \cdot t_{FA} \quad (1)$$

where, t_{FA} is the delay in carry generation of 1-bit FA cell. From (1) it is evident that the delay of RCA depends on the carry generation delay of FA cell and it increases in $O(n)$ where n is the length of RCA. Minimizing the carry propagation delay (t_{FA}) of the FA cell will minimize t_{RCA} . Though many approaches are proposed to minimize propagation delay at gate level, the fan out of the gate tends to be poor. To overcome propagation delay problem in high bit adder structures, architectural-level modifications are proposed. Approaches to reduce carry propagation delay include CSLA carry by-pass adder, carry look-ahead adder. Carry look-ahead adder will produce results with shorter propagation delay, however it occupies huge area. CSLA and its recent developments are a good choice in optimizing hardware area and delay for parallel implementation of multi-bit addition at block level.

3.1 Conventional CSLA algorithm

CSLA consists of array of RCA pair and 2:1 Muxes. Each RCA block consists of cascaded single bit full adders. For fast addition, the addition problem is broken into smaller groups. For $n = 16$ bits and 4 bits in single RCA block, CSLA will have four RCA pairs and addition is performed in all RCA pairs assuming carry-in (C_{in}) to be 0 and 1. MUX units provide the *sum* bits from RCA array using carry-out (C_{out}) from previous block. If the carry propagation time of 1-bit FA and propagation delay Mux unit are represented as t_{FA} and t_{mux} , respectively,

3.2 Modified CSLA

Modifications to CSLA to reduce area and power consumption are proposed in [21]. In this design, BEC is used to realize the function of RCA array with $C_{in} = 1$. A detailed study and implementation of BEC in CSLA is done in [5, 21]. The significance of BEC-CSLA is that it adopts a simple and efficient gate-level modification to reduce silicon area and power consumption. The logic expressions of 5-bit BEC are given in (4)–(8).

The following are the functional symbols used \sim NOT, $\&$ AND, \wedge XOR.

3.3 ERROR-TOLERANT ADDERS

As an enhancement to CSLA design for error-tolerant application, used error-tolerant FA cells to reduce gate count. Here, the *sum* output is realized as an inversion of *carry*. Logic expressions for *carry* and *sum* outputs of error-tolerant FA cell used in the design are given by (10) and (11), respectively. Gate-level implementation of the fault-tolerant adder cell is shown in Fig. 2. From the logic equations it is evident that this error-tolerant FA introduces two errors in *sum* and no error in *carry* logic with probability $P(\text{Error}) = 1/4$. For $n = 16$, ET-CSLA uses 228 logic gates. Hence, an area reduction of 46% is gained when compared to the conventional CSLA

4. PROPOSED METHOD

4.1 Proposed high-speed parallel adder

In the proposed hybrid parallel algorithm, for n bits in input operands and m bits in RCA block, the number of blocks will be n/m . Based on CSLA, high-speed addition is performed on most significant $n/2m$ blocks. In the least significant $n/2m$ blocks, second RCA array with $C_{in} = 1$ is eliminated to reduce hardware area and this introduces an approximation in the logic. The maximum error due to the approximation confines to unit bit value (UBV) with weight $2^{[(n/2m) - 1]m}$. This maximum error is tolerable for error resilient applications in digital image processing like blending T and de-noising filters.

The implementation of the algorithm for $n = 16, m = 4$ with two approximate blocks that the inputs to the adder unit are $A [15:0]$ and $B [15:0]$ and the outputs are $Sum [15:0]$ and a $Carry$. Adder array in the approximate part comprises of RCA units with inputs $A [3:0]$ and $B [3:0]$, $A [7:4]$ and $B [7:4]$ and outputs $Sum [3:0]$, $Sum [7:4]$. An AND logic on carry signals $C [3], C [7]$ of approximate part adder is used as the Select signal for first pair MUXes of accurate block. Accurate part of adder comprising of exact FA cells has inputs $A [11:8], B [11:8]$ and $A [15:12], B [15:12]$ and outputs represented as $O [11:8], C [11]$ and $O [15:12], C [15]$. These outputs are passed through BEC units for excess one conversion.

MUXes are used to select either adder outputs or BEC outputs based on previous block carry to produce final $Sum [15:12]-Sum [11:8]$ and a $Carry$. Functionality of the proposed parallel adder in digital image and signal processing applications are verified for two cases viz., exact full adder (EFA) and proposed FTFA in approximate blocks.

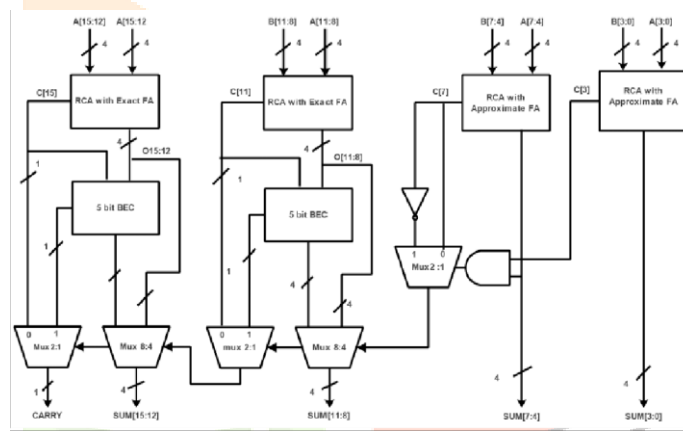


Fig: 1 Block implementation of proposed parallel adder for n=16

4.2 Proposed approximate parallel adder design with exact FA

In the proposed design using EFA hereafter referred as proposed- EFA, the accurate part and least significant approximate part of the adder are implemented with EFA cell. The logic expressions for Sum and Carry outputs of exact FA cell are given by (16) and (17), respectively.

$$Sum = A \oplus B \oplus C_{in} \quad (16)$$

$$Carry = A \cdot B + B \cdot C_{in} + A \cdot C_{in} \quad (17)$$

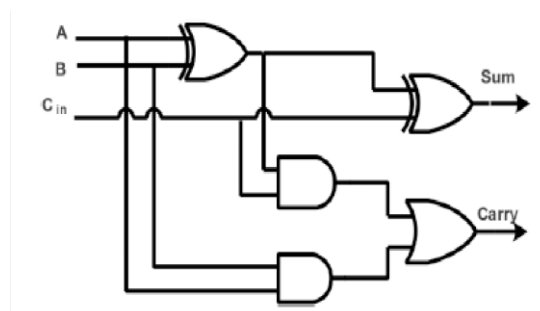


Fig:2 Logic diagram of exact FA

Functionality of the proposed parallel adder in digital image and signal processing applications are verified for two cases viz., EFA and proposed FTFA in approximate blocks. A detailed discussion on the logic design of the proposed adders is as follows. For $n = 16$, unlike SAET the accurate part consists of 2 RCA blocks, 2 BEC units and 10 multiplexers (2:1MUX) units, whereas the approximate part consists of 2 RCA blocks along with an ANDgate and 2:1 MUX unit. The function of RCA block is to generate Sum for $C_{in} = 0$ and function of BEC unit is to generate excess-1 on the RCA outputs, while the MUXes generate final output from Carry bit of previous block.

4.3 Proposed Approximate Parallel Adder with Fault Tolerant FA

In the proposed design using Fault tolerant FA cell, the accurate part is designed using exact FA cell and approximate part is designed using FTFA cells. To validate the effectiveness of the proposed algorithm in optimizing area and accuracy, two different FTFA cells are used. In proposed-FTFA1 design, FTFA cell introduces two errors in Sum and one error in Carry with probability $P(\text{Serror}) = 1/4$ and $P(\text{Cerror}) = 1/8$, respectively. Logic expressions for Sum and Carry outputs of FTFA cell are given by (18) and (19), respectively, and gate-level implementation of FTFA1

$$\text{Sum} = A \oplus (B + C_{in}) \quad (18) \quad \text{Carry} = A + (B \cdot C_{in}) \quad (19)$$

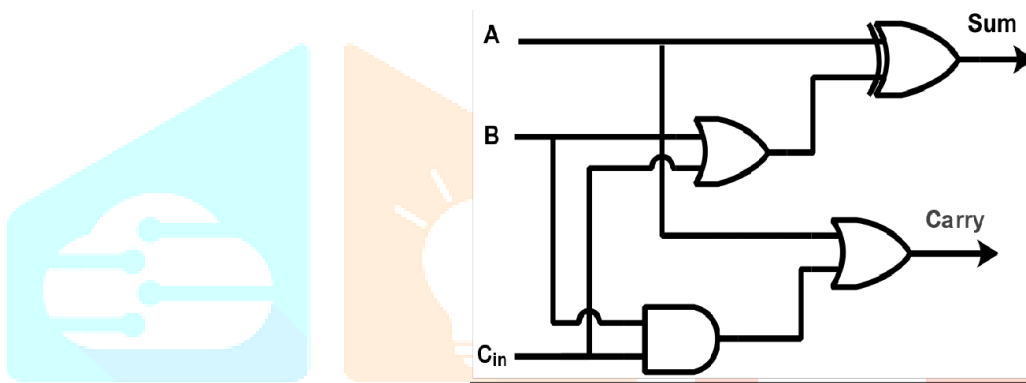


Fig. 3 Logic diagram of FTFA 1

In proposed-FTFA2 design, FTFA cell introduces 2 errors in both Sum and Carry outputs with probability $P(\text{Serror//Cerror})$ of $1/4$ each. Logic expressions for Sum and Carry outputs of FTFA cell are given by (20) and (21), respectively, and gate-level implementation of FTFA2 is shown,

$$\text{Sum} = (A \oplus B) * (\sim C_{in}) \quad (20)$$

$$\text{Carry} = C_{in} \quad (21)$$

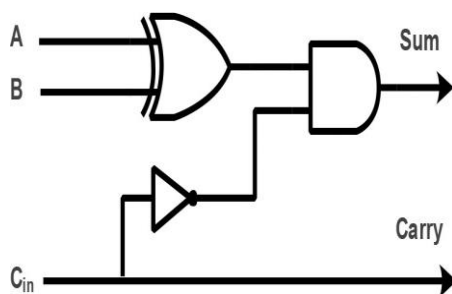


Fig. 4 Logic diagram of FTFA 2

4.4 Error comparison

An important metric to evaluate the performance of approximate computing designs is percentage of accuracy which can be estimated in terms of error percentage.

Overall error (OE): $= |R_c \sim R_e|$, OE is the difference between the correct result R_c and the result obtained by the adder R_e . Percentage of error tolerance $= [OE/R_c]/100\%$.

Accuracy (ACC): It indicates how 'correct' the output of an adder is for the given input. $ACC = (1 - (OE/R_c))/100\%$. Its value ranges from 0 to 100%.

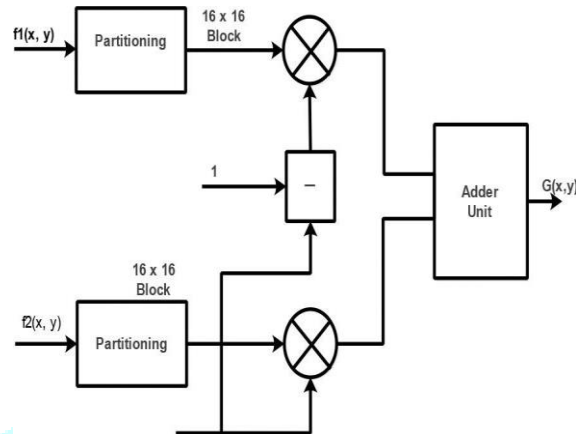


Fig: 5 Block diagram of image blending system

An illustration demonstrating output and error metrics estimation of various adder designs for the sample inputs 23,445 and 18,330. Based on the above methodology error metrics of different designs are estimated with random set of inputs and the percentage of average error is calculated based on accuracy.

4.5 Area, power and delay comparison

The proposed faithful parallel adder and state-of-the-art similar designs used for comparison viz., We have used conventional CSLA architecture as standard for performance comparison of all the error-tolerant approaches. Performance measure in terms of area, delay, power dissipation in terms of power and power-delay product (PDP) of the adder. However, the maximum error is significantly larger in ET-CSLA compared to all other designs. Proposed designs viz., proposed-EFA, proposed-FTFA1 and proposed-FTFA2 demonstrate an area reduction of 4, 11.4 and 19.9%, respectively, compared to SAET-CSLA, but the maximum errors of all the designs are limited to $2n/2$. Also note that PDP of proposed-EFA is higher than ET-CSLA and SAET-CSLA, however it can limit maximum error to significantly negligible value.

5.RESULT

In this section we describe the design procedure and the design procedure and the architecture of DES, Triple DES and Image Processing. The Verilog model was synthesized with Quartus II software targeted for device and simulated with modelsim. Then also the verilog model was synthesized with Xilinx Software targeted for 3E, Vertex 5 and Vertex E device and simulated modalism. To Design an Implementation of highly secure Triple DES Encryption standard technology was chosen because it provides some important advantages over general purpose processors and application.

5.1 SCREENSHOTS

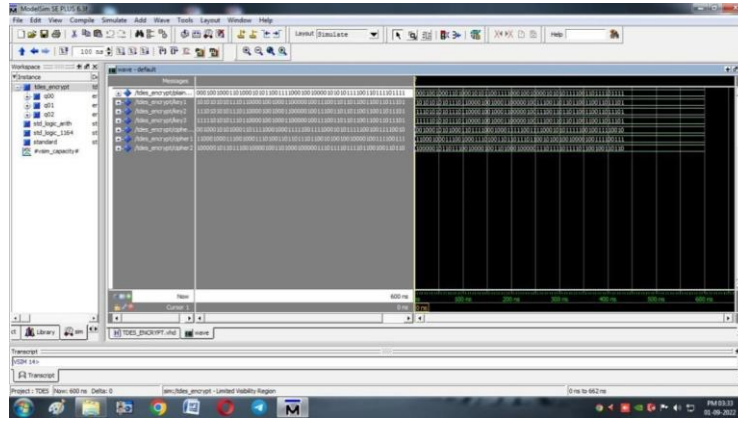


Fig : 6 MATLAB DECRYPTION

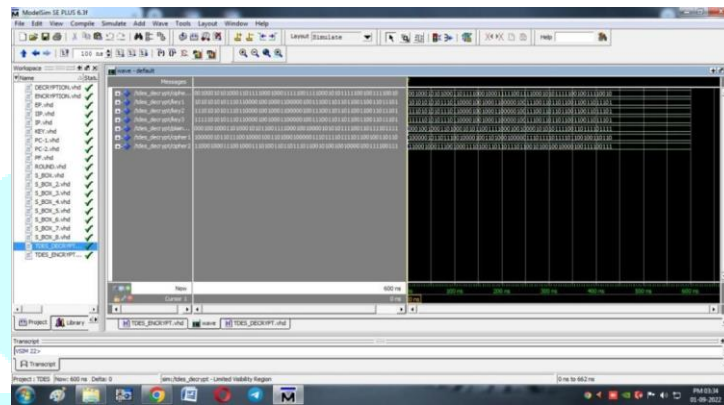


Fig: 7 DECRYPTION

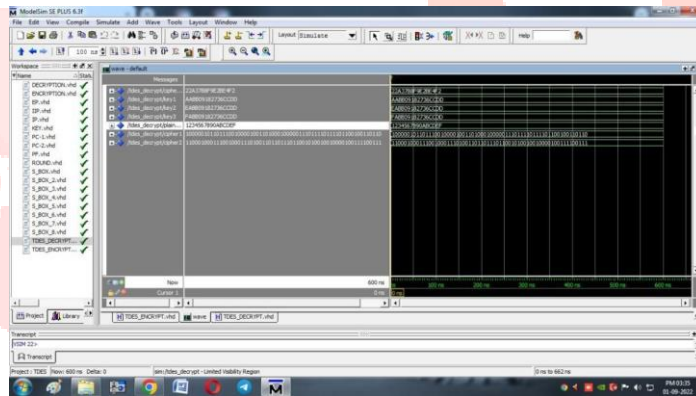


Fig: 8 DECRYPTION

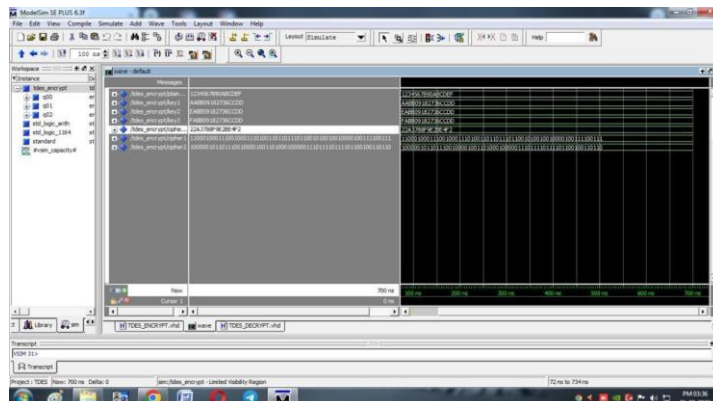


Fig: 9 ENCRYPTION

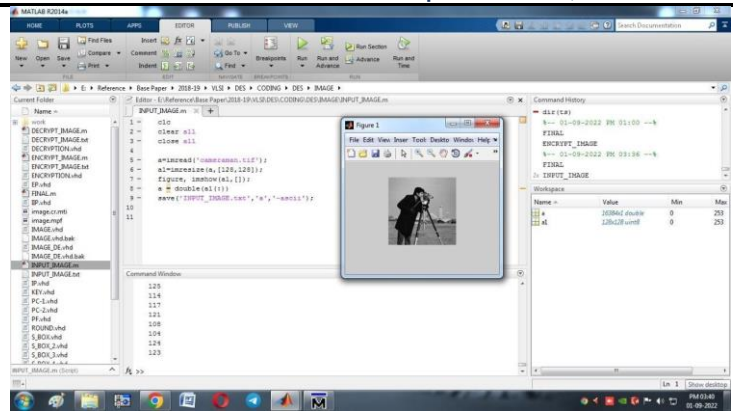


Fig: 10 SAMPLE IMAGE

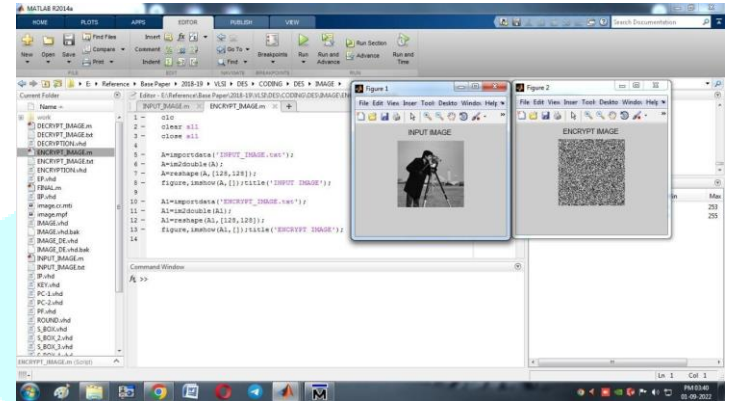


Fig: 11 IMAGE ECRYPTION

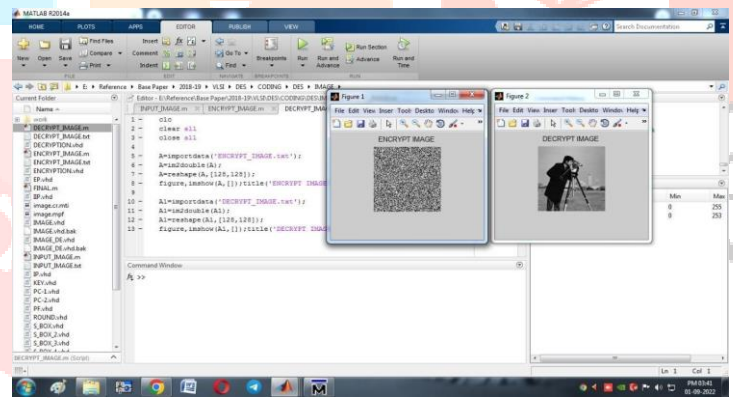


Fig: 12 IMAGE DECRYPTION

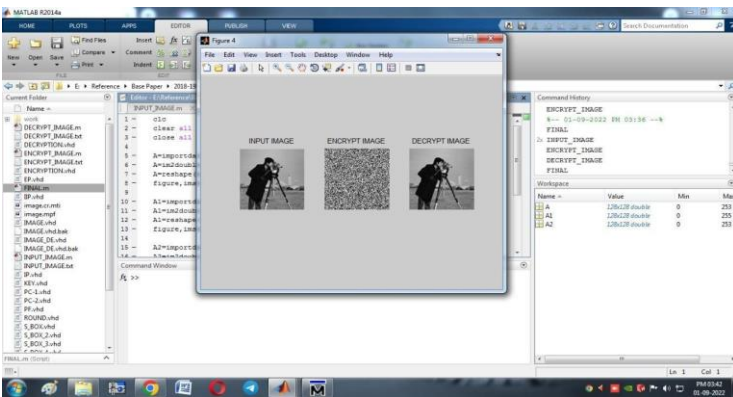


Fig: 13 ENCRYPTED IMAGE TO DECRYPPT ORIGINAL IMAGE

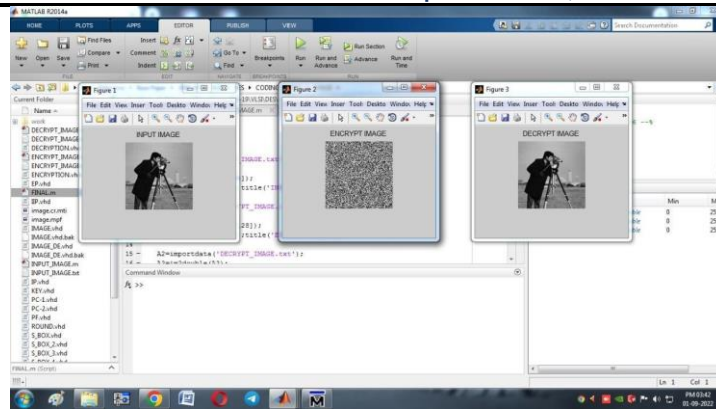


Fig : 14 ENCRYPT IMAGE TO DECRYPT ORIGINAL IMAGE

6. PROJECT FLOW

A project is a collection mechanism for an HDL design under specification or test. Even though you don't have to use projects in Modelsim, they may ease interaction with the tool and are useful for organizing files and specifying simulation settings. The following diagram shows the basic steps for simulating a design within a Modelsim project.

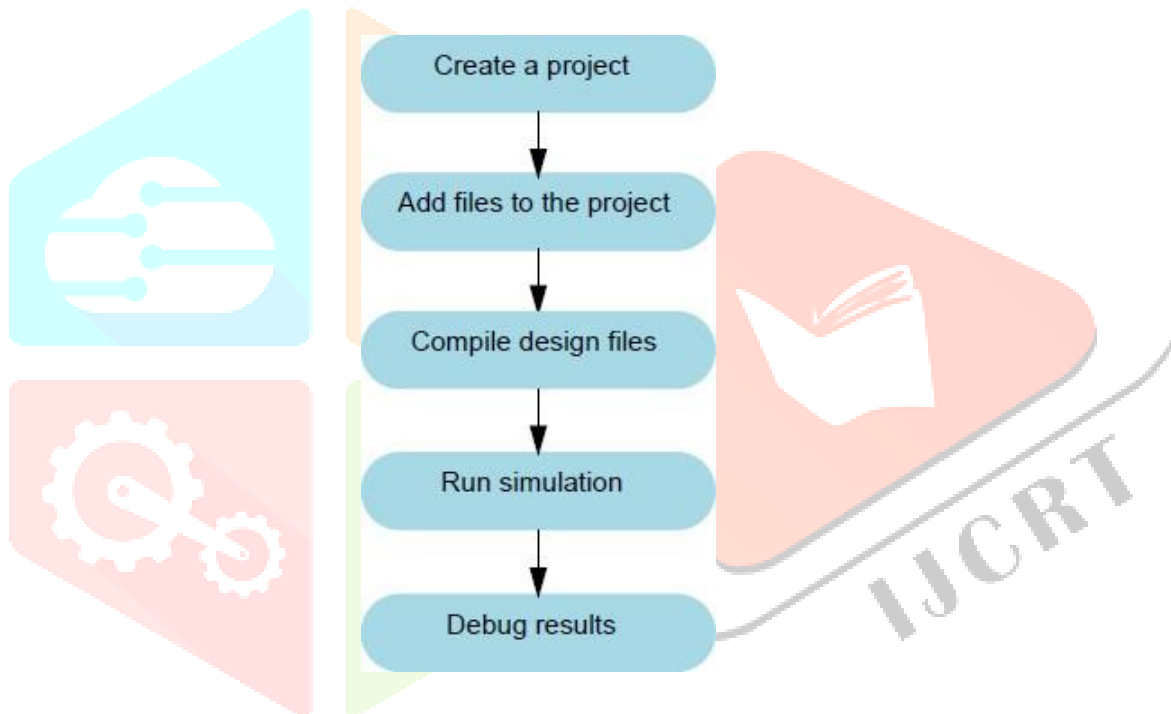


Fig: 15 Fig Project flow

The flow is similar to the basic simulation flow. However, there are two important differences:

You do not have to create a working library in the project flow; it is done for you automatically.

Projects are persistent. In other words, they will open every time you invoke Modelsim unless you specifically close them.

7. CONCLUSION

In this paper, we have proposed an area-efficient parallel adder combining carry bypass and carry select addition algorithms with faithful approximation for error-tolerant applications. Implementation of the proposed parallel algorithm in $n=16$ and $m=4$ revealed that the proposed-FTFA2 design performs better in terms of area and PDP while proposed-EFA performs in terms of accuracy. However, proposed- FTFA1 can optimize both accuracy and PDP. Functionality of the proposed parallel adder designs is evaluated in image blending and de-noising filter applications. Visual evaluation of the output images reveals that the proposed- EFA produces more accurate outputs similar to exact systems while proposed-FTFA1 and proposed-FTFA2 produce significantly better outputs than SAET-CSLA and ET-CSLA designs.

8. REFERENCE

- [1] Jothin, R., Vasanthanayaki, C.: 'High speed energy efficient static segment adder for approximate computing applications', J. Electron. Test., 2017, **33**, (1), pp. 125–132.
- [2] Jothin, R., Vasanthanayaki, C.: 'High performance significance approximation error tolerance adder for image processing applications', J. Electron. Test., 2016, **32**, (3), pp. 377–383.
- [3] Priyadarshini, K.M., Kiran, N.R., Tejasri, N., et al.: 'Design of area and speed efficient square root carry select adder using fast adders', Int. J. Sci. Technol. Res., 2014, **3**, (6), pp. 133–138.
- [4] Vasicek, Z., Sekanina, L.: 'Evolutionary approach to approximate digital circuits design', IEEE Trans. Evol. Comput., 2014, **19**, (3), pp. 432–444.
- [5] Palubinskas, G.: 'Mystery behind similarity measures MSE and SSIM'. IEEE Int. Conf. on Image Processing, Paris, France, October 2014, pp. 575–579
- [6] Reddy, D.O., Yadav, P.R.: 'Carry select adder with low power and area efficiency', Int. J. Eng. Res. Dev., 2012, **3**, (3), pp. 29–35.
- [7] Jayanthi, A.N., Ravichandran, C.S.: 'Design of an error tolerant adder', Amer.J. Appl. Sci., 2012, **9**, (6), pp. 8–18
- [8] Gupta, V., Mohapatra, D., Park, S.P., et al.: 'IMPACT: imprecise adders for low-power approximate computing'. Proc. of the 17th IEEE/ACM int. symp. on Low-power electronics and design, Fukuoka, Japan, August 2011, pp. 409–414
- [9] Ramkumar, B., Kittur, H.M.: 'Low-power and area-efficient carry select adder', IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 2011, **20**, (2), pp.371–375
- [10] Devi, P., Girdher, A., Singh, B.: 'Improved carry select adder with reduced area and low power consumption', Int. J. Comput. Appl., 2010, **3**, (4), pp. 14– 18.

