Demonstration and Verification of Low power and area efficient error tolerant adder

***Abstract---* Adder plays a major role in an image processing application . This adder will add two images to blend into a single image. To add that an adder with accurate bit and inaccurate bit is required. And one of the major concerns in image processing is the amount of time taken to process the image , the size of the adder and the power required to do the image processing. In this paper, these concerns can be solved by doing some modifications in the adder . Especially, In the Improved Low Power And Area Efficient Error Tolerant Adder(ILETA)[1] , The accurate bit is taken as the Low power and area efficient carry select adder(LCSLA)[1] and the Modified Full Adders(MFA) and an Modified Half Adder(MHA) are taken in the inaccurate part. This inaccurate bit, The changes are to be done in transistor level in order to reduce the delay, size and power. Compared to the normal full and half adders in the inaccurate part , This MFA and MHA will require low area, power and delay. In this paper, a novel Full adder and half adder are to be designed with lowest number of gate and transistor counts as possible. Designing the adders with low gate and transistor counts will reduce the space required for the inaccurate bit of the ILETA[1] to be designed in hardware. These reduced number of gate and transistor counts will also reduce the delay and power required to process. This transistor level design is designed in the Linux operated computing system where the application is cadence virtuoso-64. The design of the Modified Full Adder(MFA) and Modified Half Adder(MHA) based inaccurate bit adder are to be made and the results are compared with the normal full adder and half adder based inaccurate bit adder, some modified Approximate adders (conventional Modified Approximation(CMA)[11], Multiplexer Based Approximate Full Adder-1(MBAFA-1)[4], Modified Approximation-3(MA-3)[11], Modified Approximation-4(MA-4)[11], Multiplexer Based Approximate Full Adder-2( MBAFA-2)[4])**

***Keywords---*** **Accurate bit adder and Inaccurate bit adder, Improved Low Power And Area Efficient Error Tolerant Adder(ILETA[1]), Low power and area efficient carry select adder(LCSLA)[1], Modified Full Adders(MFA),**

**Modified Half Adder(MHA), Linux operated computing system where the application is cadence virtuoso-64, Time taken, Area required and Power required.**

**I .INTRODUCTION**

Digital signal processors (DSPs) are used to process digital signals in error-tolerant applications. DSP uses arithmetic units like adders and multipliers to carry out the arithmetic and logic operations. The performance and computational complexity of the processors are determined by these arithmetic units. To improve the efficiency of DSP processors, these arithmetic units must be optimized. Multi-bit high-speed adders are needed in DSP processors to process large amounts of data. A carry select adder is the fastest multi-bit adder currently available, despite its complex hardware. The adders can be realized with lesser complexity in hardware by adopting approximate computing. By using approximate computing, the hardware needed to implement the adders can be made simpler. By sacrificing the accuracy of the desired outputs to meet the requirements of error-tolerant applications, approximate computing offers the best design approach for arithmetic and logical units. It is compatible with both the software and hardware levels of abstraction. By removing some algorithmic levels, the Software level of abstraction is attained. By changing the hardware architecture, the hardware-level of abstraction is attained. In full adders, carrying generation causes a computational delay which is eliminated by implementing a modified full adder (MFA).

ILETA[1] is the most efficient adder in time consumption, size and power used and it is mainly used for the image processing application. The hardware level of approximation is implemented in this paper[2] and designed four different adder approximations. CSLA[3] is the adder that selects the sum or its inversion which is based on the incoming carry and also a novel logic of half sum using EX-OR gates is designed in this paper. SAET-CSLA[4] is majorly designed to handle the accurate part , as its design combines both accurate and inaccurate part. ETA[5] is also an inaccurate approximation adder which high speed performance and delay time performance. Low power full Adder[6] focuses on reducing the power of a full adder using two EX-OR gates and one 2x1 mux. Approximate EX\_OR and EX-NOR based adders[7] are designed which consumes very less power and provides better performance and the error remains almost similar. Different styles of full adder and half adder[8] are designed and the most efficient design is derived in this paper. HPETA[9] is designed using the MBAFA adders and achieves the high speed and area efficiency by minimizing the critical path delay. HSETA[10] is designed using conventional CLSA’s and MFA’s which is majorly focusing on image blending evaluation. Low power DSP using Approximate adders[11] suggests reduced transistor level complexity to achieve the low power adders which is used in multimedia circuits.

LETA has two components in it accurate 8-bit adder which consists of LCSLA[1] and the other component is inaccurate 8 bit adder using modified full and half adders.In this, the changes are made on the inaccurate part of the LETA[1] which has modified full adder and modified half adder .The novelty is to modify the full and half adder which will takes very less number of transistors as possible. For this the paper uses transmission gate logic for xor and and gate.

Many other adders were designed to check and compare the results of delay and power. The adders designed are CMA[11], MA-3[11], MA-4[11], MBAFA-1[4], MBAFA-2[4]

**II EXSISTING ADDERS**

**A. Conventional Modified Adder (CMA)[11]**

This CMA is the popular representation of the Modified full adder which consists of 24 transistors. This CMA is designed from the actual derivation of SOP of the full adder. Due to its high transistor count it takes high processing time and power to process the image. But the advantage of CMA is it has high accuracy as compared to the other adders.

**B. Modified Approximation-3 (MA-3)[11]**

This MA-3 has been derived from the above conventional adder. As the further simplification of that design leads to MA-3[11]. This design has very low number of transistors (13). So the delay size and power required to process is very low, But the accuracy while designing the LETA[1] is very less.

**C. Modified Approximation-4 (MA-4)[11]**

Further simplification of MF-3 will leads to MF-4[11]. This design has the same number of transistorsas the MF-3[11]. but the size is further decreased as compared to the MF-3[11]. But in this MF-4[11] also the accuracy was very less.

**D. Multiplexer Based Approximate Full Adder-1 (MBAFA-1)[4]**

The MBAFA-1 is designed based on the logic formulations X=B+C, Y=BC, CARRY =A, SUM =A’.X+A.Y. This design is achieved by using 1 2x1 mux, 1 OR gate, 1 AND gate. This design has very low delay, but the power and size of the design is very high when compared to the other designs.

**E. Multiplexer Based Approximate Full Adder-2 (MBAFA-2)[4]**

MBAFA-2[4] design is achieved by using the further modification of MBAFA-1[4] . This MBAFA has 1 AND gate , 1 3-input OR gate, 1 2x1 mux. This design also has very low delay, But the power and size of this design is higher than other designs.

**III PROPOSED WORK**

**A.TRANSMISSION GATE LOGIC**

It uses the complementary characteristics of the PMOS and NMOS transistor. Nmos is active when it has logic 1 and it is inactive at logic 0. Same the PMOS is active when it has logic 0 and it is inactive at logic 1.This logic can be created by placing the NMOS in parallel with the PMOS. The control signals to the transmission gate logic are complementary.

**B.MODIFIED XOR GATE**

In this design, the xor gate is designed with less transistor count. To reduce the transistor count the xor gate is designed with the help of transmission gate logic .The xor gate can be created by two transmission logic based circuits together as seen in figure . For giving the input Abar and Bbar the inverter is connected to the A and B. A normal XOR gate requires a minimum of 10 transistors but the Xor gate using transmission gate logic requires only 6 transistors.

**C.MODIFIED AND GATE**

For an AND gate the minimum number of transistors required is 6 but an AND gate using transmission gate logic requires only 4 transistors. Two transistors for the circuit and the remaining two is for the inverter the signal A.

A diagram of a circuit

Description automatically generated

***Fig 1. Conventional Modified Adder (CMA)[11]***

A diagram of a circuit

Description automatically generated

***Fig 2. Modified Approximation-3 (MA-3)[11]***

A diagram of a circuit

Description automatically generated

**Fig 3. Modified Approximation-4 (MA-4)[11]**

A diagram of a circuit

Description automatically generated***Fig 4. Multiplexer Based Approximate Full Adder-1 (MBAFA-1)[4]***

A diagram of a circuit

Description automatically generated

***Fig 5. Multiplexer Based Approximate Full Adder-2 (MBAFA-2)[4]***

A diagram of a square with lines and letters

Description automatically generated with medium confidence

***Fig.6 Transmission gate logic circuit***

A diagram of a circuit

Description automatically generated

***Fig.7 modified XOR gate using transmission gate logic***

A diagram of a square with lines and text

Description automatically generated

***Fig.8 modified AND gate using transmission gate logic***

|  |  |  |  |
| --- | --- | --- | --- |
| **GATES** | **LOGIC** | **NO. OF TRANSISTORS** | **ERROR PERCENTAGE** |
| NORMAL XOR GATE | A’B+B’A | 10 | 0% |
| MODIFIED XOR GATE | A’B+B’A | 6 | 0% |
| NORMAL AND GATE | A.B | 6 | 0% |
| MODIFIED AND GATE | A.B | 4 | 0% |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.NO** | **A** | **B** | **NORMAL XOR GATE** | **MODIFIED XOR GATE** | **NORMAL AND GATE** | **MODIFIED AND GATE** |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 1 | 1 | 1 | 0 | 0 |
| 3 | 1 | 0 | 1 | 1 | 0 | 0 |
| 4 | 1 | 1 | 0 | 0 | 1 | 1 |

***Table .1 comparison of normal and Modified XOR and AND gates***

***Table .2 comparison of truth tables of normal and Modified XOR and AND gates***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **normal** | **CMA** | **MA-3** | **MA-4** | **MBAFA-1** | **MBAFA-2** | **proposed** |
| A B C | S Ca E | S Ca E | S Ca E | S Ca E | S Ca E | S Ca E | S Ca E |
| 0 0 0  0 0 1  0 1 0  0 1 1  1 0 0  1 0 1  1 1 0  1 1 1 | 0 0 0  1 0 0  1 0 0  0 1 0  1 0 0  0 1 0  0 1 0  1 1 0 | 0 0 0  1 0 0  1 0 0  0 1 0  1 0 0  0 1 0  0 1 0  1 1 0 | 1 0 -1  1 0 0  0 1 0  0 1 0  1 0 0  0 1 0  0 1 0  0 1 +1 | 0 0 0  1 0 0  0 0 +1  1 0 0  0 1 0  0 1 0  0 1 0  1 1 0 | 0 0 0  1 0 0  1 0 0  1 0 -1  0 1 +1  0 1 0  0 1 0  1 1 0 | 0 0 0  1 0 0  1 0 0  1 0 -1  1 0 0  1 0 -1  0 1 0  1 1 0 | 0 0 0  1 0 0  1 0 0  0 0 +1  1 1 -1  0 1 0  0 1 0  1 1 0 |

***Table 3.* *Comparison of Different Adders***

**D.PROPOSED MODIFIED HALF ADDER**

As discussed earlier, the modified half adder is used in place of the normal half adder which will reduce the transistor count . The modified half adder requires the same number of gate counts (one XOR and one AND gate). But the number of transistors used will be reduced based on the transmission gate logic based design. The XOR gate is to produce the sum and the AND gate is to provide the carry. An inverter is used to invert the input A.

**E.PROPOSED MODIFIED FULL ADDER**

Full adder plays an important role in this image processing adder application. The normal full adder has either two half adders and one OR gate or two Xor ,Two AND and one OR gate. Where the sum is the addition of two half adders and the carry is the addition of two AND gates and one OR gate. But in the modified full adder circuit the Full adder is designed with two XOR gates only , where the sum

is the addition of two XOR gates and the carry is directly taken from the first input. By taking the carry as the first input , there may be some error(an interchange of one 1 and one 0) but mostly the carry won’t affect the adder's design as the 16 bit adder’s inaccurate part does not require any carry,It depends on the previous input.

Where the normal full adder has 2 AND gates, 2 XOR gates and 1 OR gate. In this the must depend on the output from the XOR gate.

**A diagram of a diagram

Description automatically generated**

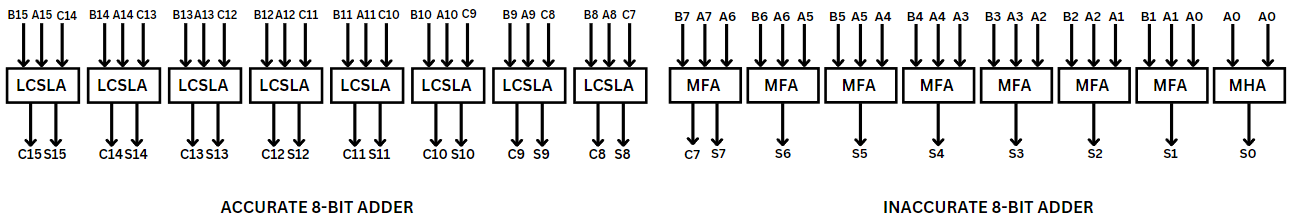
***Fig 9. Modified Full Adder***

**IV. ADDER DESIGNS**

In this paper an area efficient error tolerant adder is designed based on the LETA[1]. These adders have two different parts, namely an accurate part or MSB and an inaccurate part or LSB. The change is done in the inaccurate bit or LSB of the adder.

**A. LOW POWER AND AREA EFFICIENT ERROR TOLERANT ADDER(LETA)**

This adder has a LCSLA[1] in the accurate bit or MSB of the adder and the Modified Full Adders(MFA) and Modified half adder(MHA) in the inaccurate part or LSB of the adder . The LCSLA[1] is designed using 7 basic gates and 1 multiplexer. The accurate part is generally designed with 8 LCSLAs[1] and the inaccurate part is designed with 7 MFAs and 1 MHA. The accurate part of the LETA[1] requires 88 basic gates and the inaccurate part of the LETA requires 60 basic gates. Totally 148 basic gates are required to design the LETA[1] .The MFAs in the inaccurate part of the LETA[1] uses the gets two inputs from the external source and the third is the first input of the previous MFA. The LCSLA[1] gets two inputs from the external source and the third input is the carry of the previous LCSLA[1].

******

***Fig 10. Low Power and Area Efficient Error Tolerant Adder (LETA)[1]***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **DESIGN** | | **ADDERS** | | | | **TOTAL GATES** |
| **ADDERTYPE** | **NO. OF ADDER BLOCKS** | **TOTAL GATES PER ADDER BLOCK** | **TATAL GATES** |
| **LETA** | **ACC** | LCSLA[1] | 8 | 11 | 88 | 148 |
| **INACC** | MFA,  MHA | 7,1 | 8,4 | 56,4 |

***Table 4. LETA[1] properties***

**V. COMPARISON OF NORMAL VS PROPOSED ADDER**

**A. NORMAL INACCURATE BIT**

The inaccurate bit of the already existing LETA[1] has seven MFAs and one MHAs. The inputs are given in 8 bits for the MFAs and MHA. Half adder gets the input directly, where the full gets to inputs directly from the external source and another one input is the first input of the previous full adder respectively and the carry of the final MFA is given to the first LCSLA[1].In normal inaccurate bit the design was change in gate level , where the number of gate counts decreases. Then the time delay , area and power required for the adder also decreases. The normal adder doesn’t require any gates to generate carry from the final MFA.

**B. PROPOSED INACCURATE BIT**

The proposed inaccurate bit is the same as the normal inaccurate bit of the LETA[1] but the design of the normal inaccurate bit of the adder is optimized in gate level and the proposed inaccurate bit of the adder was optimized in the transistor level But in transmission gate logic the carry can’t be taken directly from the first input. And for carry an OR gate needs to be connected to the first input and the other input must be considered as logic 1, so that whatever the third input of the last MFA will be taken as the carry also.

|  |  |  |  |
| --- | --- | --- | --- |
| **Adders** | **No. of transistors for a adder** | **No. of Adders required** | **Total no. of adders** |
| Normal adder | 22, 16 | 7,1 | 170 |
| CMA[11] | 27,16 | 7,1 | 205 |
| MA-3[11] | 13,16 | 7,1 | 107 |
| MA-4[11] | 14,16 | 7,1 | 114 |
| MBAFA-1[4] | 18,16 | 7,1 | 142 |
| MBAFA-2[4] | 20,16 | 7,1 | 156 |
| Proposed adder | 12,10 | 7,1 | 94 |

***Table 5. Transistor Count of Adders***

|  |  |  |
| --- | --- | --- |
| **Adders** | **Delay** | **Power** |
| Normal adder | 322.1E^-12 | 2.335E^-6 |
| CMA[11] | 375.6E^-12 | 1.085E^-6 |
| MA-3[11] | 367E^-12 | 583.2E^-9 |
| MA-4[11] | 64.60E^-12 | 695.9E^-9 |
| MBAFA-1[4] | 133.9E^-12 | 1.088E^-6 |
| MBAFA-2[4] | 122.1E^-12 | 675.8E^-6 |
| Proposed adder | 12.93E^-12 | 448.8E^-9 |

***Table 6. delays and average powers of adder***

**VI. RESULT AND ANALYSIS**

To measure the delay and power required by the adder to process, the inputs must be considered in 8-bit and the following results are verified with both normal inaccurate bit and the proposed inaccurate bit. For the verification of inaccurate 8-bit adder two inputs are chosen to the adder and the following results are verified.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **INPUT** | **SET-1** | | | | | | | | | **SET-2** | | | | | | | | |
| **CIN** |  | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |  | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| **A** |  | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |  | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| **B** |  | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |  | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| **S** | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| cout | s7 | s6 | s5 | s4 | s3 | s2 | s1 | s0 | cout | s7 | s6 | s5 | s4 | s3 | s2 | s1 | s0 |

***Table 7. Result Analysis Table-1***

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **INPUT** | **SET-3** | | | | | | | | | **SET-4** | | | | | | | | |
| **CIN** |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **A** |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **B** |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **S** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| cout | s7 | s6 | s5 | s4 | s3 | s2 | s1 | s0 | cout | s7 | s6 | s5 | s4 | s3 | s2 | s1 | s0 |

***Table 8. Result Analysis Table-2***

**A. RESULT:**

A graph with different colored lines

Description automatically generated

***Fig 11. Result-1 of inaccurate bit of ILETA[1]***

A graph with different colored lines

Description automatically generated

***Fig 12. Result-2 of inaccurate bit of ILETA[1]***

**B. POWER AND TIME DELAY ANALYSIS**

The adder was designed in the linux computed operating system with the virtuoso application. The power can be measured by taking the average power required for the transistor to perform the adder operation. The time delay can be measured by choosing the first output sum(s1) and the carry. For measuring the time delay the rising or falling edge are taken and compared. The comparison is made between different modified approximation adders to prove that the proposed adder is better and more efficient. To verify that two different combinations are tested and the result is written below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Combination-1** | | **Combination -2** | |
|  | **Delay** | **Power** | **Delay** | **Power** |
| Normal | 263.9E^-12 | 13.48^E-6 | 14.125E^-12 | 14.18^E-6 |
| CMA | 314.1E^-12 | 6.909^E-6 | 21.22E^-12 | 7.233^E-6 |
| MA-3 | 291.3E^-12 | 4.419^E-6 | 60.83E^-12 | 5.323^E-6 |
| MA-4 | 118.7E^-12 | 7.945^E-6 | 13.02E^-12 | 6.446^E-6 |
| MBAFA-1 | 83.01E^-12 | 6.562^E-6 | Not reflected | 13.02^E-6 |
| MBAFA-2 | 124.2E^-12 | 7.391^E-6 | Not reflected | 15.68^E-6 |
| Proposed | 95.77E^-12 | 4.184^E-6 | 11.33E^-12 | 2.722^E-6 |

***Table 9. Delay and Power comparison of ILETA[1] Design of Different Adders.***

**VII. CONCLUSION**

We have reduced the transistor count from 156 to 100 in the inaccurate 8- bit part of the LETA[1] by modifying the full adders and half adders .So the power required will be reduced to 4.184^E-6 and 2.722^E-6 due to the reduced count of the transistor. And the time delay is also reduced to 95.77E^-12 and 11.33E^-12. As a human nature we won’t focus much on the clarity of the image in the image processing application , we will consider only the delay and the power consumed. So this circuit will helps to reduce the power and delay of the image processing application.

**VIII. REFERENCES**

[1] Low power and area efficient error tolerant adder for image processing application, M Priyadharshni Sundaram Kumaravel Received: 22 August 2019 Revised: 13 December 2019 Accepted: 29 December 2019 DOI: 10.1002/cta.2744

[2] Dutt S, Nandi S, Trivedi G. Analysis and design of adders for approximate computing. ACM Trans Embed Comput Syst (TECS). 2018;17(2):40

[3] Kumar VN, Raj PP, Lakshminarayanan G, Sellathurai M. Low power and area efficient carry select adder. J Low Power Electron. 2014;10(4):593-601

[4] Jothin R, Vasanthanayaki C. High performance significance approximation error tolerance adder for image processing applications. J Electron Test. 2016;32(3):377-383.

[5] Zhu N, Goh WL, Zhang W, Yeo KS, Kong ZH. Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing. IEEE Trans Very Large Scale Integr (VLSI) Syst. 2010;18(8):1225-1229.

[6] Design and Analysis of Low Power Full Adder March 2017, Conference: National Conference CSPCT'17 ,At: MADURAI

[7] Yang Z, Jain A, Liang J, Han J, Lombardi F. Approximate XOR/XNOR-based adders for inexact computing. In: 2013 13th IEEE International Conference on Nanotechnology (IEEE-NANO 2013) IEEE; 2013:690-693.

[8] Half Adder Using Different Design Styles: A Review on Comparative Study September 2020, Journal of Advancements in Robotics ,ISSN: 2455-1872 ,Volume 7, Issue 3 ,[www.stmjournals.com](http://www.stmjournals.com)

[9] Jothin R, Vasanthanayaki C. High performance error tolerant adders for image processing applications. IETE J Res. 2018:1-12.

[10] Geetha S, Amritvalli P. High speed error tolerant adder for multimedia applications. J Electron Test. 2017;33(5):675-688.

[11] Low-Power Digital Signal Processing Using Approximate Adders. Vaibhav Gupta, Debabrata Mohapatra, Anand Raghunathan, Fellow, IEEE, and Kaushik Roy, Fellow, IEEE. IEEE transactions on computer-aided design of integrated circuits and systems, vol. 32, no. 1, january 201312