Dynamic Accuracy Configurable 64 Bit Multipliers using Dual-Quality 4:2 Compressors

Sneha Susi S
M.Tech in VLSI & Embedded System
ECE Department
Dr. Ambedkar institute of technology
Bangalore, Karnataka-560056

Meenakshi L Rathod
Assistant professor
ECE Department
Dr. Ambedkar institute of technology
Bangalore, Karnataka-560056

Abstract:
The compressor which is projected here has the capacity to change between the approximate and exact modes it has lower power, low accuracy, and high speed in the approximate phase. In exact as well as in approximate modes these dual quality compressors will exhibits different level of accuracy, delay and power dissipation. So during runtime their accuracy, speed, and power may variant vigorously by employing these compressors in design of parallel multipliers. The same compressors are used with different bit size like 8, 16, 32, 64 bits the tabular column is created between different bit levels to know the difference between them and the area, speed, and delay are also calculated for different bits. Error correction is calculated at the output for each and every bit. And this compressor are used in the image processing application here smoothing technique is considered.

Index Terms **— Adders, Compressors, Approximate Multipliers and Error Distance parameter.**

I. INTRODUCTION

The multiplier is one of the main components in many arithmetic blocks; they are used in signal processing. Parallel and Sequential are the architectures in the multipliers. Parallel architecture are nothing but the Dadda multipliers this will consume high power and are very much fast and the Sequential consist of higher latency and it requires the less power Hot spot are created on the die because of their large power and higher performance in the parallel multipliers, as we all know that speed and power consumption are the important parameters to consider so the optimization of this parameter is very much important and usually the optimization proceeds in such a way that one parameter is considered while considering the limitation for the other. For portable devices the speed is considered by using the limited power is a task need to be considered and we will consider the reliability also. But in early time, much of the work is on approximate multipliers and this also provide the lower delay, power, and the accuracy, but the proposed multipliers are also the approximate one which will give the fixed accuracy in the runtime. This reconfigurable accuracy is very much useful in giving the better quality of services while performing the operation, so by decreasing the accuracy of the system the other factors are also reduced like the power and delay. For both the modes that are for exact as well as the approximate modes, the general purpose processor may be used. This can be used by using the correction unit for the error detection in the output with this the approximate unit is used, but this correction unit may give more power, area, and delay in the circuit. And it also needs more than one clock cycle which in turn slows down the process. 4:2 approximate compressor which we have presented here have the capability to reconfigure and also it can switch between the approximate and the exact modes and this are mainly used in designing of the parallel multipliers, and this compressor which are proposed consist of two parts supplementary and approximate, and for better purposed in exact mode some units of approximate part along with supplementary part is active, but only approximate part is active in approximate mode.

II. RELATED WORK

For the error resistant applications the approximate design can reduce the complexity of design but increases the power efficiency and performance this mainly deals with the multipliers approximation [1]. To present the verity of probability terms in multipliers the partial products are changed and approximations logical difficulty speckled so that based on their chance the partial products are accrued. In 16 bit multipliers the two approximations which are proposed are used the final report called the synthesis report for the software shows that multipliers which are proposed have a power saving of 72% and 38% correspondingly and this are compared with the exact multipliers too and the mean relative error is also calculated and this proposed multipliers are used in the image processing application.

In both math's and science the addition and multiplication are the simple and main operation the speed of the digital signal processors are determined by the multipliers speed here we are going to give the brief description about four multipliers that are Dadda, Wallace, modified booth, Vedic multipliers and here the Vedic multipliers are used in diagonal multiplication, and used to reduce from multi bits to single bit, modified booth are used to describe the partial product process [2]. And when coming to Dadda multipliers it is used in minimum reduction process and has adders which have larger carry propagation so it has high speed. Next will be the Wallace multiplier this has adder which has smaller carry propagation because it reduces all partial products.

The earlier Vedic math's concepts are used in the design of the compressors which are used in design of the Vedic multipliers and are used in increasing the performance of the multipliers [3]. The basic parameters that are going too considered in designing of the multipliers are low power, less delay, less area, and high speed. The 16 sutras are used to solve

the Vedic mathematics. Here to get the reduced area, delay, power and for high speed we have used 4:2 and 7:2 compressors. The two full adders and 4:2 compressors are used in construction of 7:2 compressors. This are performed using Spartan 3 and timing, area, delay are intended.

Convolutional Neural Networks are being concentrated to give highlights, for example, continuous picture acknowledgment [4]. One of the key activities to help HW executions of this kind of system is the duplication. In spite of the high number of activities required by Convolutional Neural Networks, they ended up practical in the previous years due the high accessibility of registering power, display on gadgets, for example, Graphic Processing Units. Be that as it may, those usage are unacceptable for vitality compelled situations, for example, installed gadgets. FPGAs are programmable gadgets that are being considered as a low power elective for GPUs. This work proposes and executes a generator of two quick combinatorial multipliers: Dadda and Wallace tree. Our generator is equipped for creating basic depictions of the two plans for any operand width, an activity thought about unfeasible by hand. We assessed our generator utilizing two minimal effort FPGA stages, effectively found available.

III. EXISTING SYSTEM

A. Exact 4:2 Compressors

The 4:2 and 5:2 compressors are commonly used in reducing the delay of the partial product in parallel multiplier summation stage. Here we are using the 4:2 compressor and basics of exact 4:2 compressor are described here. The below figure 1 describes the approximate 4:2 compressor it has an input carry (Cin), (x1-x4) are the four inputs, and it has 2 outputs called carry and sum and finally the Cout which is also an output.

Here the exact 4:2 compressor internal design is shown it consist of 2 full adders which are serially connected it is described in the Figure 2 in the above design output sum and the input of all the loads are same, but Cout and carry loads of the output are higher by one binary bit position. The below formulas show the formulas for sum, carry, and Cout.

Sum=x1⊕x2⊕x3⊕x4⊕Cin,

carry=(x1⊕x2⊕x3⊕x4) Cin+(x1⊕x2⊕x3⊕x4)'x4 and

Cout= (x1⊕x2) x3+ (x1⊕x2)'x1

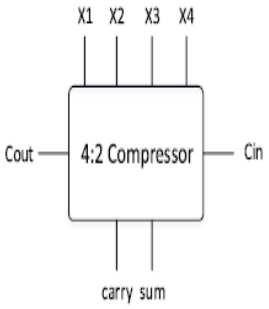

Figure 1: compressors 4:2 basic block diagram

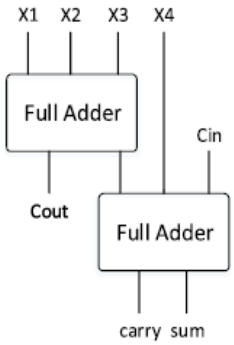

Figure 2: compressors 4:2 conventional design

B. Adders

Addition of two binary digits forms the adder. It is used in addition of two numbers. But this are used in binary addition, and this are commonly used in the calculation of table index, decoding of address, and in the decimals of binary code, etc. we all know the adders can be classified into half adders and full adders, here in this paper we have consider basic of full and half adder formulas it is given below

- Half adder truth table

S=A ⊕ B (Exclusive OR)

C=A.B (AND)

A	B	Sum	Carry-Out
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- Full adder truth table

S= A ⊕ B ⊕ Cin

C=AB + Cin (A ⊕ B)

A	B	Carry-In	Sum	Carry-Out
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

Table 1: truth table for full adder and half adder with formulas

IV. PROPOSED SYSTEM

The dual quality 4:2 compressors are explained in this section. It has the capacity to exchange between the correct and rough working modes. The compressors might be used in the dynamic quality configurable parallel multipliers. There are two structures in proposed compressors consist of two sections one is estimated and other is supplementary part. In the exact operating mode supplementary part and some approximate components are active and when we consider in approximate mode the approximate mode is active.

A. 4:2 Compressors dual quality

Block diagram of the compressor which is proposed here is designed in figure 3 the accuracy lies in two different modes approximate mode and exact mode. The control gated has been done for the supplementary part and approximate part is fragmented in the other mode. But we are going to use some

approximate components and supplementary part is considered in exact mode. But we are going to use the much components of approximate part in exact proposed operating mode this is very much useful in decrease of area and power. Unused component is turned OFF in the approximate part by using the power gating technique. The tri-state buffers are used to separate the approximate part and the primary outputs.

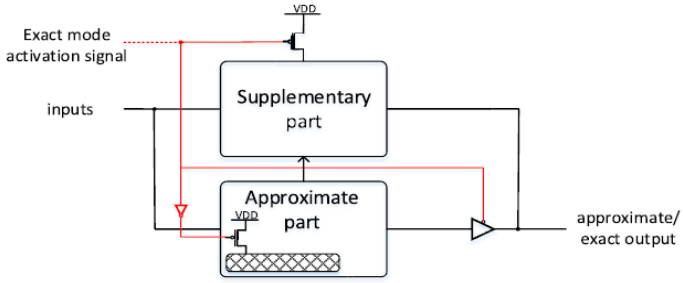

Figure 3: Proposed approximate 4:2 compressor block structure.

We can see from the above figure 3 that the hachured box which is there in the approximate part is not shared between supplementary and approximate parts.

i. **DQ4:2C1 design 1**

Approximate fragment of planned design 1 is shown in the figure 4(a), here the carry is straight associated for the input X4, and the sum is straight connected to the input X1, and in the same approximate part the output Cout is not considered. Here the approximate part is fast and uses the low power. And error rate is more.

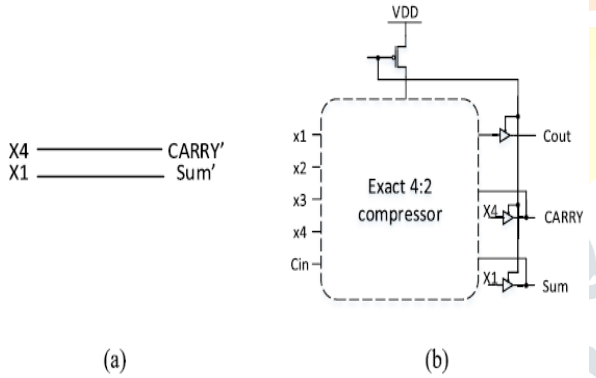

Figure 4: (a) Approximate fragment (b) complete design of DQ4:2C1

The complete design is shown in the Figure 4(b). The supplementary part in the given design is a 4:2 compressor with exact nature, this design has the delay same as the exact 4:2 compressor

ii. **DQ4:2C2 design 2**

Cout is not considered in the first design and because of this it has properly shortened the internal design of reduction stage in the multiplication, the error is additional in the succeeding design when we compare with the first design, here the output is straight connected to the input x3 in the approximate part as presented in the Figure 5 it shows the complete design and the approximate part of DQ4:2C2. Here in this second structure the relative error is lower and the error rate is same as the DQ4:2C1.

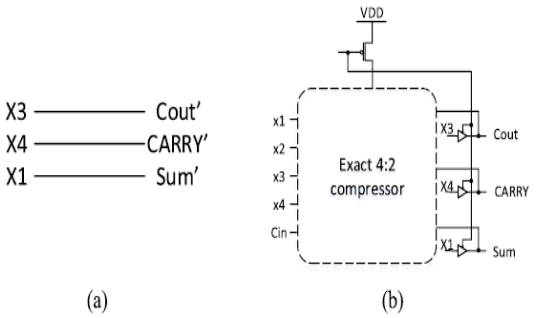

Figure 5: (a) Approximate part and (b) complete design of DQ4:2C2.

iii. DQ4:2C3 design 3

In the design2 the power and delay is reduced in the approximate operating mode when compared with the exact compressors. And lets coming to third design when accuracy is the main factor in approximate operational approach, it is enhanced my increasing the design complication of approximate fragment the inner design is presented in the below Figure 6. In this the accuracy is increased in the output sum. As same as the design 1 the Cout is not supported in the approximate part. The error rate in this design is reduced by 50%.

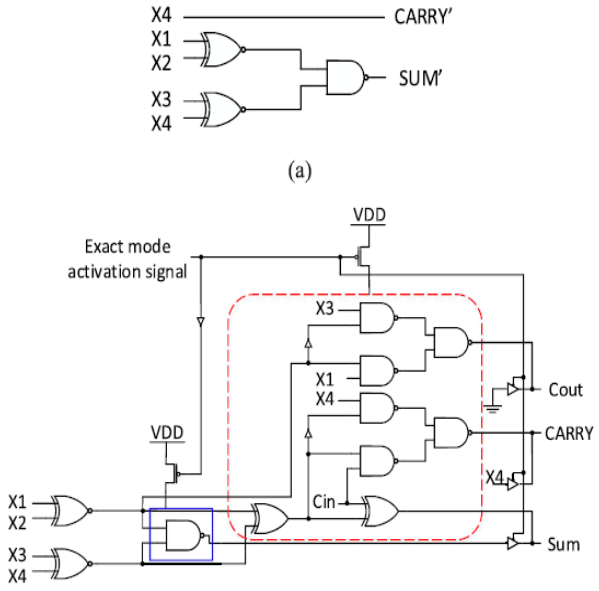

Figure 6: (a) Approximate part and (b) complete design of DQ4:2C3

iv. DQ4:2C4 design 4

Here the accuracy is more in the output carry when compared with the DQ4:2C3. But the delay and power consumption are more with the reduced error rate. The approximate part internal structure and the complete design are shown in the above Figure 7. The blue dotted line indicates the logic gates of approximate fragment are offed throughout the operation in the exact approach, and the red dashed line drawn as a form of rectangle indicates the supplementary part.

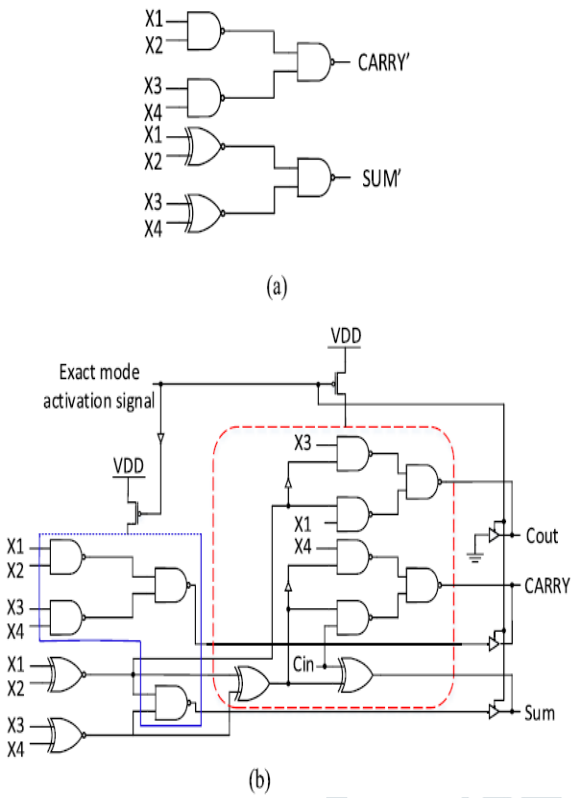

Figure 7: (a) Approximate fragment (b) general arrangement of DQ4:2C4

The quality of the output is resolute by error distance in short we called it as ED in this section it will give the error rate for the whole series of input which is applied to get the errors in the output; ED is defined as the difference between the approximate unit and the exact output. Here with ED we will consider some parameters they are (MRED) mean relative ED and (NED) normalized ED even this are very much useful in finding the output quality. For the error robust application the correctness of an approximate unit for a given quality of output can be determined.

B. Error distance parameter (ED)

For the complete range of input the occurrence of error in the output is called error rate. The error distance parameter gives the quality of output and it is the variance among the outputs of approximate unit and the exact unit.

The output quality is determined by the parameters which are in addition with ED they are (MRED) mean relative ED, (NED) normalized ED. And we should consider these parameters to judge the correctness of the error robust applications in there approximate part.

The definition of MRED can be stated as The ED difference between exact and approximate output with each and every patterns of input which is divided by the exact output, this is given by the formula below.

$$MRED = \frac{1}{2^{2N}} \sum_{i=1}^{2^{2N}} \frac{|ED_i|}{S_i}.$$

.....1

NED can be stated as like in the below formula this is used to compare the approximate multipliers with their independent sizes.

$$NED = \frac{MED}{D} = \frac{1}{2^{2N}} \sum_{i=1}^{2^{2N}} \frac{ED_i}{D}$$

.....2

For approximate multipliers the D is a possible maximum ED, but (2N-1)2 is considered for most of the approximate multipliers. So we can substitute D by (2N-1)2, the NED can be defined as below

$$NED = \frac{MED}{(2^N-1)^2} = \frac{1}{(2^N-1)^2} \sum_{i=1}^{2^{2N}} \frac{|ED_i|}{2^{2N}}$$

.....3

C. Proposed approximate Half Adder and Full Adder

i. Proposed Architecture of 8 Bit approximate adder

Proposed with a new architecture of half adder and full adder as we know for 8 bit addition there is total 7 full adder and 1 half adder is require. But in proposed approach we propose a new novel 8 bit architecture where we can put some error on LSB bit of adder. Here in approximate half and full adder there is no any carry generation unit. So on first LSB bit we are using proposed approximate half adder and on second LSB bit we use one approximate full adder for next third bit there is no any carry generate so there is no need to use one full adder so at the place of full adder we are using one half adder and after that we use 5 full adder. So as we can see with small error generation we can reduce the hardware requirement and we can make justice with SPAA matrices. The Figure 8 displays the Planned Approximate Half adder; figure 9 displays the Future Approximate Full Adder and

Proposed approximate half adder formula

Input..... x1, x2
Output..... sum, carry
Sum =x1|x2
Carry =x1&x2

Figure 8: Proposed Approximate Half Adder

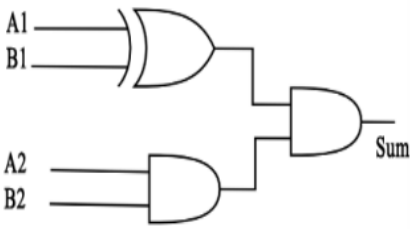

Figure 9: proposed approximate full adder

Proposed approximate full adder formula

Input..... x1, x2, x3

Output.....sum, carry

Wire.....w

w=x1|x2

Sum=w^x3

Carry=w&x3

V. DESIGN METHODOLOGY

A. The main steps involved in this project are given below

- Programming in Verilog is written by using formulas given for exact compressor, half adder, and full Adder; this will give the exact values.
- Then same like above programming in Verilog is written by using formulas given for proposed approximate compressors and approximate half adders, and approximate full adders.
- Here four approximations are used for compressors and that are compared with the exact one, And the best approximate compressor with lesser area and delay are selected.
- This exact compressor, half adder, full adder are used in exact multiplication and the approximate compressor, half adder, full adder are used in approximate multiplication.
- So there are four approximation in compressor this four are used one by one in approximate multiplication and are compared with the exact multiplication, and the best one is selected, Same code can be written for 8 bit, 16 bit, 32 bit, and 64 bit.
- This can be checked with the mat lab the quality of the image is checked by smoothing factor.
- Image processing is used in the mat lab, here image pixel is selected and that is multiplied with the filter.
- Then the quality of the image is checked in all approximate compressors and best approximation is selected based on the quality and clarity of the image.

B. Detail examination

At starting of the project we are going to the examine the accuracy of 4:2 compressors in approximate working mode and this examination is done by using in the Dadda design and this are matched with the Dadda multiplier which as an exact nature and here Dadda multiplier are understood by using the 2 proposed 4:2 compressors and the main aim is to design the best design with reduced area and delay and here we are going to compare the effectiveness of the approximate and proposed compressor with the exact compressors used in the Dadda multipliers both performance are compared in the same exact working mode, to know the working of the proposed approximate multipliers are used in the image processing applications and different bits of multipliers are constructed and in this different bits like 8, 16, 32, and 64 bits the difference between area and delay and error correction at the output are calculated.

C. Approximate Operating Mode

By the 8 bit multipliers we can see that the approximate proposed multipliers have the decreased delay and area and the correction of errors at the output are also reduced when we compare with the exact one. And as the length of the bit keeps on increasing the enhancement also increases now we will consider the multipliers of 16 bit here also the compressor of proposed approximate multiplier have a decreased area and delay when we compare with the Dadda multiplier which as an exact nature used in the compressors. Like this it will be the same for the 32 and 64 bit Dadda multipliers here also proposed approximate multipliers are more efficient than exact one here also the area and delay are decreased in approximate compressor then the exact one.

D. Exact Operating Mode

The multipliers using DQ4:2Cs structures have lesser delay, area and EPD values when comparing with the other proposed multipliers and the design constraints are more in approximate multipliers due to the presence of tristate buffers in the output.

The decreased delay and area for 8, 16, 32, 64 bit can be seen for proposed DQ4:2C design when compared with the other multipliers and H spice simulators are used in compressors which are proposed for analyzing the switching time expenses.

E. Image Processing Applications

To describe the proposed design application in real time they are mainly used in image processing applications they are image multiplication, sharping and smoothing here we are using 64 bit proposed multipliers which uses the approximate compressors.

The below formula gives the output as sharpened image.

$$Y(i,j) = 2.X(i+m, j+n) - \frac{1}{273}\sum_{m=-2}^{2}\sum_{n=-2}^{2} \times X(i+m, j+n)\cdot \text{Mask}_{\text{Sharpening},1}(m+3, n+3)$$

Input and output are denoted by X and Y and Mask Sharpening matrix is

$$\text{Mask}_{\text{sharpening}} = \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}.$$

The below equation is used for smoothing the resulted image

$$Y(i,j) = \frac{1}{60}\sum_{m=-2}^{2}\sum_{n=-2}^{2} \times X(i+m, j+n)\cdot \text{Mask}(m+3, n+3)$$

Where Mask Smoothing is given by

$$\text{Mask}_{\text{smoothing}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 4 & 4 & 4 & 1 \\ 1 & 4 & 12 & 4 & 1 \\ 1 & 4 & 4 & 4 & 1 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}.$$

In this area, ten standard benchmark pictures from were utilized. Pinnacle motion to-clamor proportion is constraints; it is utilized in examining the nature of inexact pictures. The picture quality seen by the human have more reliability than this constraint mentioned above a superior constraint for this object is (MSSIM) mean structural similarity index metric which works in light of estimating the basic comparability of the correct and approximate pictures. It depends on the rule that the human visual framework is fit for removing data in light of the picture structure. The articulation for this parameter which is somewhat mind boggling. So in this paper it describes about smoothing as primary to check the nature of the picture.

VI. RESULT AND DISCUSSION

The results for the 64 bit has been show below as like this the simulation results are observed for the 8, 16, 32 bits and the comparison table for this are tabulated in the column. This 64 bit compressor are used in image processing application, this paper shows the result for smoothing.

1. The below figure 10 shows the delay and area value generated from Xilinx software for 64 bit exact 4:2 compressor

multiplier64_accurate Project Status (07/28/2018 - 12:19:32)			
Project File:	susi.xise	**Parser Errors:**	No Errors
Module Name:	multiplier64_accurate	**Implementation State:**	Synthesized
Target Device:	xc6slx4-3tqg144	**• Errors:**	No Errors
Product Version:	ISE 14.7	**• Warnings:**	4 Warnings (0 new)
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	8286	2400	345%
Number of fully used LUT-FF pairs	0	8286	0%
Number of bonded IOBs	256	102	250%

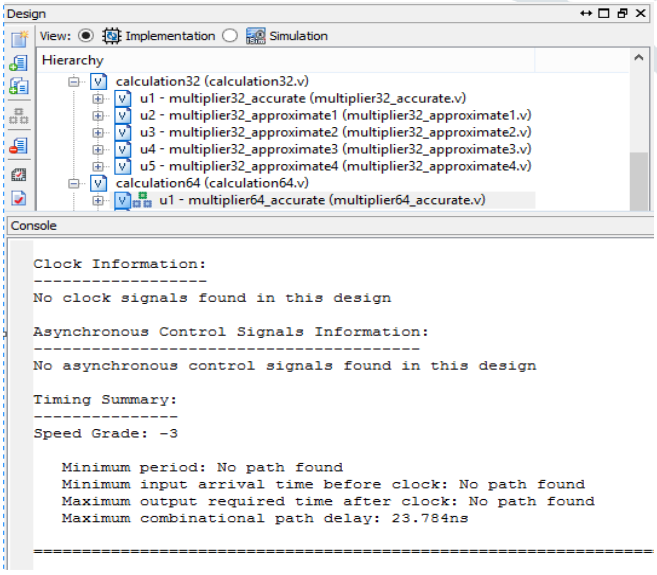

Figure 10: 64 bit Exact 4:2 compressor

2. The below figure 11 shows the delay and area value generated from Xilinx software for 64 bit approximate compressor DQ4:2C1.

multiplier64_approximate1 Project Status			
Project File:	susi.xise	**Parser Errors:**	No Errors
Module Name:	multiplier64_approximate1	**Implementation State:**	Synthesized
Target Device:	xc6slx4-3tqg144	**• Errors:**	No Errors
Product Version:	ISE 14.7	**• Warnings:**	1030 Warnings (0 new)
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	3662	2400	152%
Number of fully used LUT-FF pairs	0	3662	0%
Number of bonded IOBs	256	102	250%

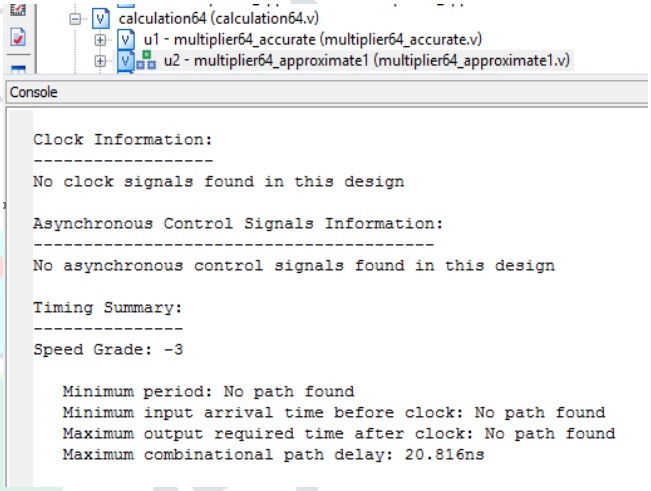

Figure 11: 64 bit approximate compressor DQ4:2C1

3 The below figure 12 shows the delay and area value generated from Xilinx software for 64 bit approximate compressor DQ4:2C2.

multiplier64_approximate2 Project Status			
Project File:	susi.xise	**Parser Errors:**	No Errors
Module Name:	multiplier64_approximate2	**Implementation State:**	Synthesized
Target Device:	xc6slx4-3tqg144	**•Errors:**	No Errors
Product Version:	ISE 14.7	**•Warnings:**	70 Warnings (2 new)
Design Goal:	Balanced	**•Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**•Timing Constraints:**	
Environment:	System Settings	**•Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	3486	2400	145%
Number of fully used LUT-FF pairs	0	3486	0%
Number of bonded IOBs	256	102	250%

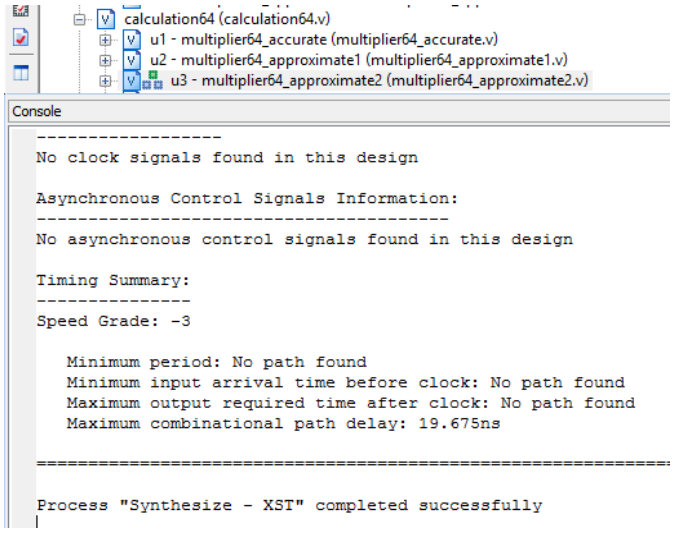

Figure 12: 64 bit approximate compressor DQ4:2C2

4. The below figure 13 shows the delay and area value generated from Xilinx software for 64 bit approximate compressor DQ4:2C3.

multiplier64_approximate3 Project Status			
Project File:	susi.xise	**Parser Errors:**	No Errors
Module Name:	multiplier64_approximate3	**Implementation State:**	Synthesized
Target Device:	xc6slx4-3tqg144	**•Errors:**	No Errors
Product Version:	ISE 14.7	**•Warnings:**	5 Warnings (1 new)
Design Goal:	Balanced	**•Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**•Timing Constraints:**	
Environment:	System Settings	**•Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	4687	2400	195%
Number of fully used LUT-FF pairs	0	4687	0%
Number of bonded IOBs	256	102	250%

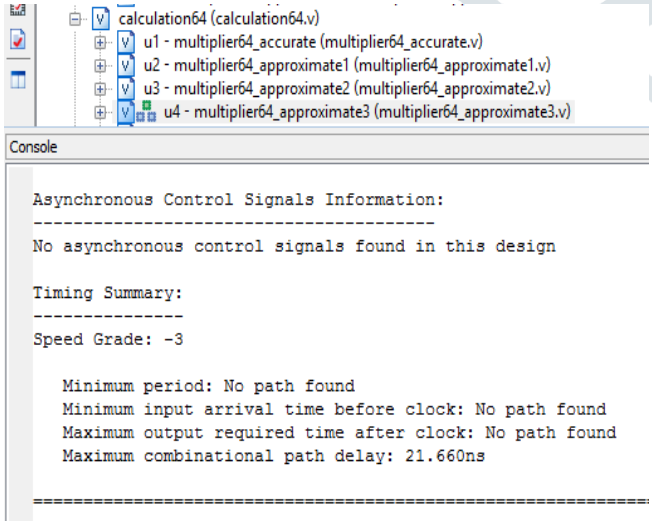

Figure 13: 64 bit approximate compressor DQ4:2C3

5. The below figure 14 shows the delay and area value generated from Xilinx software for 64 bit approximate compressor DQ4:2C4.

multiplier64_approximate4 Project Status			
Project File:	susi.xise	**Parser Errors:**	No Errors
Module Name:	multiplier64_approximate4	**Implementation State:**	Synthesized
Target Device:	xc6slx4-3tqg144	**•Errors:**	No Errors
Product Version:	ISE 14.7	**•Warnings:**	5 Warnings (1 new)
Design Goal:	Balanced	**•Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**•Timing Constraints:**	
Environment:	System Settings	**•Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	6575	2400	273%
Number of fully used LUT-FF pairs	0	6575	0%
Number of bonded IOBs	256	102	250%

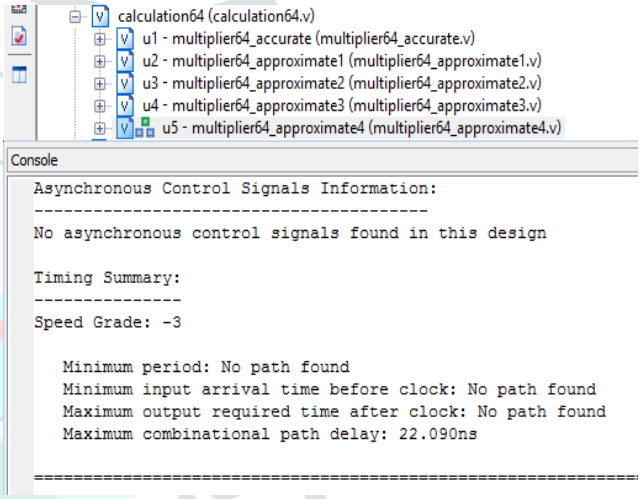

Figure 14: 64 bit approximate compressor DQ4:2C4

6. The below figure 15 shows the 64 bit MRED and NED calculated values generated from Xilinx software

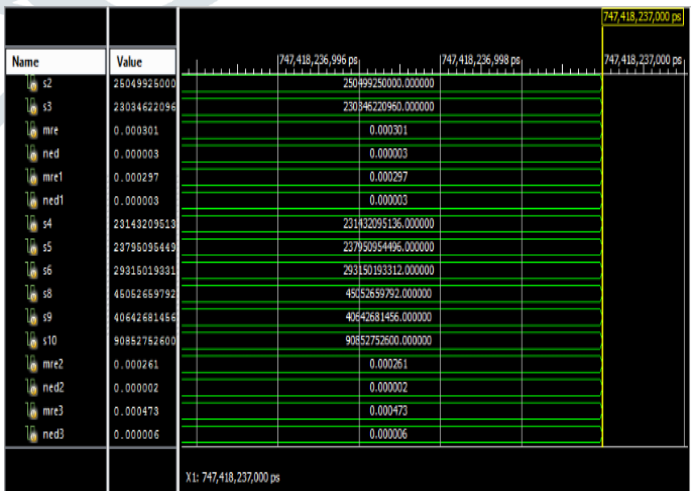

Figure 15: 64 bit MRED and NED calculated values

7. Mat lab results in PSNR

The PSNR block computes the peak signal-to-noise ratio, in decibels, between two images. The MSE represents the

cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. Lower the value of MSE, the lower the error. Below figure 16 shows the PSNR values for the all four approximations.

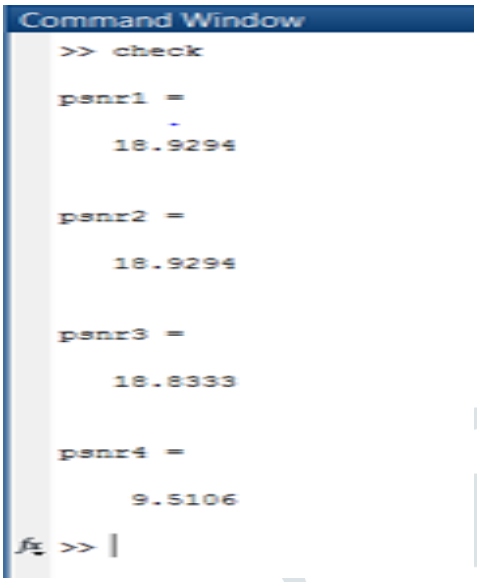

Figure 16: PSNR Values

8. The below figure 17 shows the output for different approximation and exact compressor this are the results from the mat lab which shows that the approximate 2 means the figure 3 has an image clarity as like the exact image with less area, and delay.

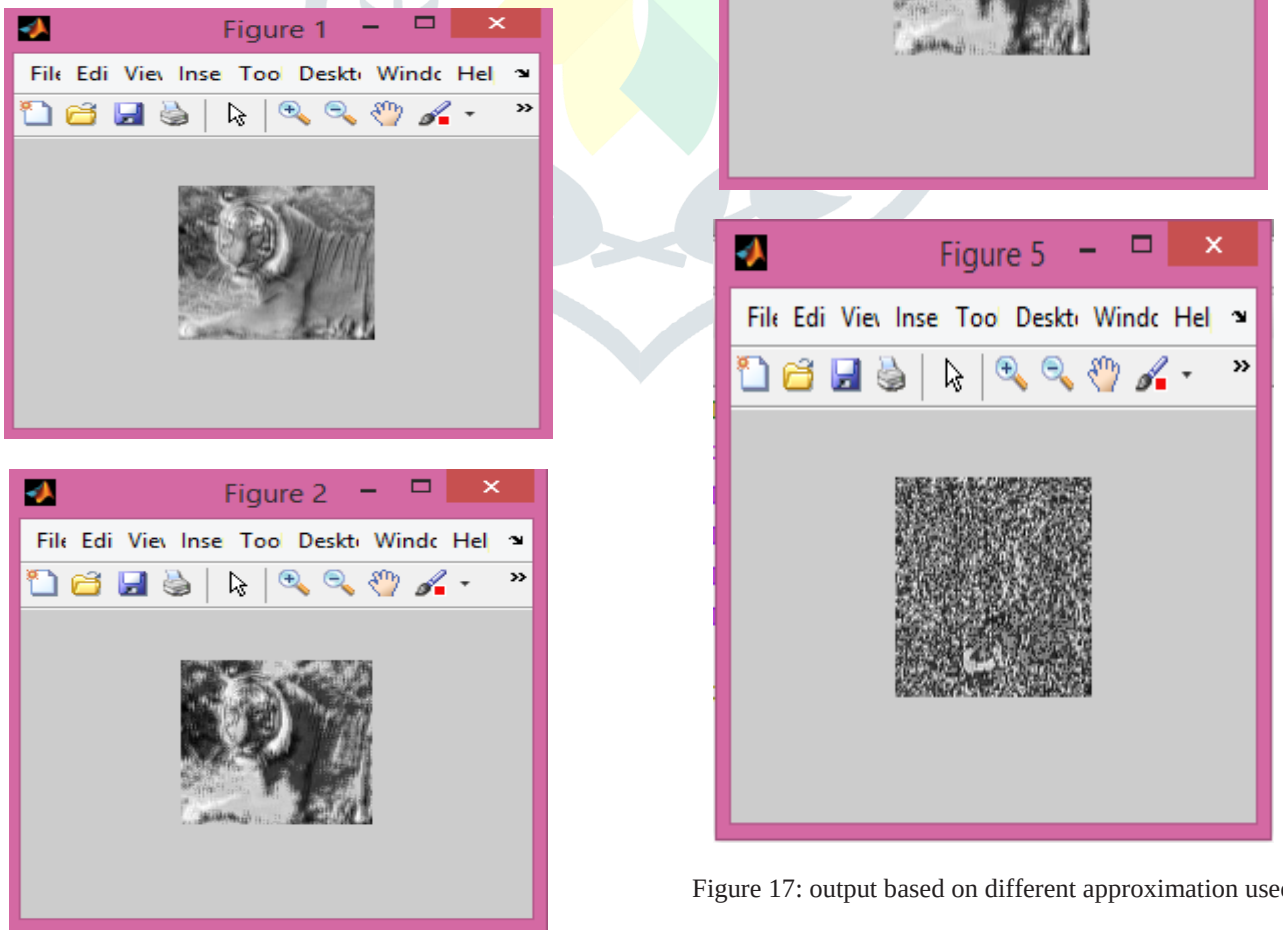

Figure 17: output based on different approximation used

9. Performance analysis

Table 2: 8 bit compressor comparison table

8 bit 4:2 compressor structure	MRED	NED	AREA(no. of slice LUTS)	DELAY In ns
Exact	-	-	100	11.166
DQ4:2C1	0.000311	0.000003	28	8.366
DQ4:2C2	0.000231	0.000002	25	7.140
DQ4:2C3	0.000516	0.000004	45	8.471
DQ4:2C4	0.000737	0.000006	78	9.545

By this comparison table the compressor DQ4:2C2 is more efficient when compared with other structure in area and delay

Table 3: 16 bit compressor comparison table

16 bit 4:2 compressor structure	MRED	NED	AREA(no. of slice LUTS)	DELAY In ns
Exact	-	-	477	14.805
DQ4:2C1	0.000301	0.000003	186	12.072
DQ4:2C2	0.000297	0.000003	175	10.891
DQ4:2C3	0.000261	0.000002	253	13.029
DQ4:2C4	0.000473	0.000006	370	13.110

When compared with area and delay the compressor structure DQ4:2C2 is more efficient.

Table 4: 32 bit compressor comparison table

32 bit 4:2 compressor structure	MRED	NED	AREA(no. of slice LUTS)	DELAY In ns
Exact	-	-	2017	18.997
DQ4:2C1	0.000301	0.000003	859	16.240
DQ4:2C2	0.000297	0.000003	815	15.088
DQ4:2C3	0.000261	0.000002	1122	17.201
DQ4:2C4	0.000473	0.000006	1592	17.312

By area and delay comparison between the DQ4:2C2 compressor is more efficient than any other compressor structure.

Table 5: 64 bit compressor comparison table

64 bit 4:2 compressor structure	MRED	NED	AREA(no. of slice LUTS)	DELAY In ns
Exact	-	-	8286	23.784
DQ4:2C1	0.000301	0.000003	3662	20.816
DQ4:2C2	0.000297	0.000003	3486	19.675
DQ4:2C3	0.000261	0.000002	4687	21.660
DQ4:2C4	0.000473	0.000006	6575	22.090

DQ4:2C2 compressor is more efficient when compared with the other structure in the above table. By comparing all above performance table the compressor structure DQ4:2C2 in 8, 16, 32, 64 bit are more efficient when area and delay are considered so the structure DQ4:2C2 is much more used when area and delay are main parameters.

VII. CONCLUSION AND FUTURE SCOPE

Conclusion

For 8-bit, 16-bit, 32-bit and 64-bit multipliers the delay, area, power, energy, and EDP of the Dadda multipliers using the proposed approximate compressors are improved compared with those of the Dadda multipliers employing the exact compressor. The improvements increase as the bit length increases.by using this method; we can conclude that performance (speed) of the system can be increased by using reduced area and delay approximate compressor in the system. By using approximate compressor in image processing techniques the clarity of the image can be improved and tried to reach the clarity of the exact multiplier image, with reduced area and delay in this paper the DQ4:2C2 means the approximation 2 designs is more efficient when compared with other compressors.

Future scope

The image can be replaced by videos and the clarity of videos can be checked by using approximate multipliers. And that are tired to match the clarity of the exact multipliers.

VIII. REFERENCE

[1] Omid akbari, Mehdi kamal, ali afzali-kusha, and massoud pedram, "Dual-quality 4:2 compressor for utilizing in dynamic accuracy configurable multipliers," *IEEE Transaction on Very Large Scale Integration(VLSI) Systems*, vol. 25, no. 4,pp. 1352 - 1361, April 2017.

[2] Suganthi venkatachalam and seok_bum ko, "Design of power and area efficient approximate multipliers," *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 5,pp. 1782 - 1786, May 2017.

[3] Sumod Abraham, Sukhmeet Kaur and Shivani Singh, "Study of Various High Speed Multipliers," *IEEE – International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2015.

[4] Radheshyam Gupta,Rajdeep Dhar and K.L. Baishnab,Jishan Mehedi, "Design of High Performance 8 bit Vedic Multiplier using Compressor," *IEEE – International Conference on Advances in Engineering and Technology (ICAET 2014)*, May 2014.

[5] Lucas G. de Castro, Henrique S. Ogawa, and Bruno de C. Albertini, "Automated Generation of HDL Implementations of Dadda and Wallace Tree Multipliers," *IEEE – VII Brazilian Symposium on Computing Systems Engineering (SBESC)*, November 2017.

[6] Addanki Purna Ramesh, "Implementation of Dadda and Array Multiplier Architectures Using Tanner Tool," *International Journal of Computer Science & Engineering Technology (IJCSET)*, Vol. 2, no. 2,pp. 2229-3345, 2011.

[7] D. Baran, M. Aktan, and V. G. Oklobdzija, "Multiplier structures for low power applications in deep-CMOS," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2011, pp. 1061–1064.

[8] 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 Integration. (VLSI) Syst.*, vol. 18, no. 9,pp. 1301–1309, Sep. 2010.

[9] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "RAP-CLA:A reconfigurable approximate carry look-ahead adder," *IEEE Trans. Circuits Syst. II, Express Briefs*, doi: 10.1109/TCSII.2016.2633307.
[10] A. Sampson *et al.*, "EnerJ: Approximate data types for safe and general low-power computation," in *Proc. 32nd ACM SIGPLAN Conf. Program.Lang. Design Implement. (PLDI)*, 2011, pp. 164–174.
[11] A. Raha, H. Jayakumar, and V. Raghunathan, "Input-based dynamic reconfiguration of approximate arithmetic units for video encoding," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 3,pp. 846–857, May 2015.
[12] J. Joven*et al.*, "QoS-driven reconfigurable parallel computing for NoC-based clustered MPSoCs," *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1613–1624, Aug. 2013.
[13] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, "On reconfiguration oriented approximate adder design and its application," in *Proc.IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, Nov. 2013,pp. 48–54.
[14] S. Hashemi, R. I. Bahar, and S. Reda, "DRUM: A dynamic range unbiased multiplier for approximate applications," in *Proc. IEEE/ACM Int.Conf. Comput.-Aided Design (ICCAD)*, Austin, TX, USA, Nov. 2015,pp. 418–425.
[15] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 984–994, Apr. 2015.
[16] C. H. Lin and I. C. Lin, "High accuracy approximate multiplier with error correction," in *Proc. IEEE 31st Int. Conf. Comput. Design (ICCD)*,Oct. 2013, pp. 33–38.
[17] C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," in *Proc. Conf. Design, Autom. Test Eur. (DATE)*, 2014, Art. no. 95.
[18] R. Zendegani, M. Kamal, M. Bahadori, A. Afzali-Kusha, and M. Pedram, "RoBA multiplier: A rounding-based approximate multiplier for high-speed yet energy-efficient digital signal processing,"*IEEE Transation Very Large Scale Integration (VLSI) Systems*, doi 10.1109/TVLSI. 2016. 2587696.
[19] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.