IJREAT International Journal of Research in Engineering& Advanced Technology, Volume 8, Issue 4, Aug - Sep, 2020
ISSN: 2320 – 8791 (Impact Factor: 2.317)
www.ijreat.org

SELF-HEALING FIR FILTER USING ERROR SUSTAIN ADDER AND SYNTHESIZABLE MULTIPLIER

[1]Swaminathan.K, [2]Vijay.R, [3]Vennila.C

[1]Research Scholar,Dept. of ECE,University college of engineering Pattukkottai,Tamilnadu,India
[2]Assistant Professor, Dept. of EEE, Saranathan College Of Engineering,Trichy,Tamilnadu,India
[3]Professor, Dept. of ECE, Saranathan College Of Engineering,Trichy,Tamilnadu,India

Abstract. In this paper, a 4 tap FIR filter with a self-Healing methodology is presented. The self-healing methodology is carried out by two separate FIR filter re-designed by a synthesizable multiplier and Wallace multiplier as a separate module. Additionally, an Error sustain adder is used in both modules for adding the product of input data stream and filter coefficient of present and previous states. The input stream of data is given in a parallel manner for avoiding the time delay between FIR modules and filter output from the two modules is compared with conventional FIR output for occurrence of error in the filter output. Wallace multiplier is implemented by 4:2 and 3:2 compressors as its basic logic element in the partial product reduction. The works aims to achieve the self-Healing methodology in the front end level of VLSI design technique itself.

Index Terms—Front end design, Hardware description language (HDL), Synthesizable multiplier, Error sustain adder, Wallace multiplier,**4:2compressor,**Verilog

I. INTRODUCTION

FIR filters are stable and can be designed with linear phase leading to give error-free output[1].However, the realization often involves a complex structure to meet the requirement.It based on selecting a symmetric linear phase (impulse response) sequence of specified length and involves iterative techniques. For speech processing and datatransmission requires frequency response to be flat and phase response in linear with frequency in pass band. They can be realized using FIR filter. So, the FIR filter output should be free from noise. This leads to go for “Self –Healing” technique to be implemented in FIR filter[2-6].

Generally Self-healing used for recovering a error free output from a module having an internal defect. As FIR realization has adders and multipliers, the chance of getting the erroneous output. Here a synthesizable multiplier, Wallace multiplier with compressor, error sustain adder in FIR filter implementation. By comparing the output from individual FIR filter, corresponding error free output is given. The entire design is carried out using hardware description language (HDL) broadly in Verilog.

It is a popular standard language used from abstract level to concrete chip level [7, 8]. The use of function may give only one value [9]. Hence, use of procedure gives more solution for synthesizable multiplier.

Microprocessor,Arithmetic architectures and signalling processor uses digital signal for their complexity analysis,performsnce calculation[10].Digital signal processor involves convolution,filterin operation.It is accomplished by multupliers,adders and shifters as its processing module.This multipliers have genrarion of partial products,propagation of carry which increases circuit complexity and power consumptionMany techniques were proposed for reducing the partial product.The use of compressor is a way to reduce the partial terms.It is designed with half and full adders which count the number of one’s in the input data.Different range of compressor like 3:2,4:2,5:2 compresser were used for the optimization.

For producing an error free output accuracy of the computing device must be 100% in comparison actual design of the device.But when the optimization is implemented in the excat calculation it is not possible to achieve the 100% accuracy.This implies that the performance of the approximation is reflects on the accuracy achieved.Also exact calculation is not required for every applications.Applications like image processing,multimedia,filtering etc., may tolerate error in thier output.

Inexact computing technique become more familiar because of its low complexity,less area occupied which gives tolerable error rate in the output.Parametrs like Error Rate(ER),Error distance(ED) are used to calculate the accuracy of the approximated output.Error Rate gives number errorneous output to the total number of exact output.Error Distance is the arithmetic distance between Exact and approximated output.The entire design is carried out using hardware description language (HDL) broadly in Verilog.

Approximation was carried out by XOR/XNOR gate operation in the calculation of the adder output[15] in the sum and carry output.Instead of using XOR/XNOR gate operation, multiplixer were used for the approximation of the output[16].A 2X2 bit multiplier is optimized by altering the gates used in the sum and cary blocks in a full adder block of a multiplier.With this,multiplier produces “7d” when the inputs are “11”,”11”.But the exact output is “1001”.This results in the probability of error to be 0.0625.Several optimization were proposed in the reduction of partial product [17]-[20].

A particular row and column of partial product may be skipped when the value of corresponding multiplier and multiplicand bit is “0” respectively [17]-[18].This is called as “Row and Column Bypassing”. In [19], carry save adders are skipped in both horizontal and vertical directions based on the zeros in partial products. The partial product was divided into two parts as MSB and LSB. The MSB part will be calculated by accurate method, whereas LSB will be calculated on approximated techniques [20].

Here, Section II describes self-healing methodology, Section III gives hardware implementation, Section IVresult and simulation, Section V gives Conclusion and Section VI describes references.

II. SELF HEALING METHODOLOGY

In FIR filter, we have adders, multipliers for performing addition, product of filter co-efficient and input data respectively. The structure of 4-tap FIR filter includes adders, multipliers and D flip flop as shown in the Fig 1a.

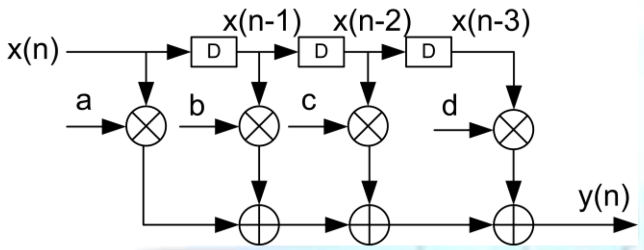

Fig 1a. 4 - Tap Filter

The conventional adders in the filter are replaced by error sustain adders. Wallace multiplier, Synthesizable multiplier are implemented as separate module. The module get the filter inputs simultaneously process the inputs for processing the output without any noise. The proposed methodology is given in the Fig 1b.

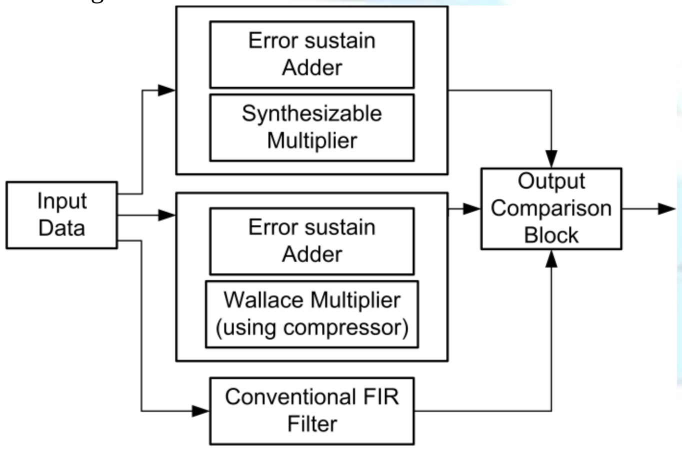

Fig 1b. Proposed Methodology

The various FIR modules produces their corresponding outputs as shown in Fig1b. The outputs are compared by finding the difference between outputs with conventional FIR outputs. If the outputs are same, the difference is zero. The conventional FIR filter output is treated as final output. If the differ, on comparison of output with conventional FIR output. The difference between the outputs are found out. The output pair which has the minimum value is given as final output by finding average value between them.

A. *Error sustain Adder*

The conventional adders speed is limited by the propagation of delay from least significant bit (LSB) to most significant bit (MSB). To overcome this trade-off, we can go for Error Sustain Adder. Some basic terminology used in the adder are given as below [8]:

1) *Total Error* (T_E)*:* Here O_C denotes correct output, O_E denotes erroneous output obtained at the adder

2) *Accuracy* (A_C)*:* It gives the error free result of adder for inputs in terms of Accuracy. It is calculated as given below equation (1)

$$A_{C=}\ [1-(T_E/O_C)]*100 \qquad (1)$$

3) *Acceptable Accuracy*(A_A): It is the acceptable accuracy greater than threshold value to meet the requirement of the accuracy

4) *Acceptable Probability* (P_A) *:* It ranged from 0 to 1must be higher than acceptable accuracy ($A_C > A_A$)

Generally delays are propagated from LSB to MSB for conventional adders. For neglecting this delay, we go for given below addition procedures in Fig 1c.

Process direction ← | → Process direction

(MSB) (LSB)

1	0	0	0	1	1	0	0	1	0	1	0	1	0	0	0
0	1	0	0	0	0	1	0	0	1	0	0	1	1	0	1
1	1	0	0	1	1	1	0	1	1	1	0	1	1	1	1

↓

All bit set to 1

Fig 1c. Addition procedures

In the above addition, 16 bit input operands are divided equally into two parts each of 8 bits. So, it is divided as 8 bits and the addition operation is carried over in opposite direction. The procedure for approximate addition as follows:

i) For Accurate part, the normal addition between inputs operands are taken, applied from LSB to MSB. This done for maintaining correctness

ii) For inaccurate part, addition is carried over from LSB to MSB as per the condition given below

a) When both the bits of input data are "0" or different normal addition will be takes place for calculating the output

b) When both the bits are "1" then addition will be stopped, all the output bits are assigned as "1" onwards.

For the above example, the actual output is 52972. But the output through approximation is "52970".The total error (T_E) and Accuracy (A_C) can be computed as follows in equation (2) and (3)

$$\text{Total Error } (T_E) = 52972 - 52970 \quad (2)$$

$$\text{Accuracy } (A_C) = [1 - (2/\ 52970)]*100 = 99.9\ \% \quad (3)$$

Hence the accuracy is above 98%, the overall delay can be greatly reduced by eliminating carry propagation from LSB toMSB

B. Synthesizable Multiplier in FIR

In the conventional multiplier, the partial addition, carry propagation will take place. This will make the time delay in the output generation. This delay may be avoided using synthesizable multiplier.

This process will be initiated with a temporary register of (2n+1) with a size of "n" bit multiplier and multiplicand. It is carried out by (n-1) shifting operations. The multiplication is done on the basis of "LSB" bit position as per the condition stated below:

i) Copy the multiplier to temporary register from bit position 0 to (n/2 -1) position

ii) Check the LSB position, do any of the step below

(a) Simply right shift the temporary register and increment the counter value

(b) Done thepartial addition with multiplicand, right shift the temporary register

(iii) If the counter value reaches (n-1) value, stop the process. The content of the temporary register gives the final product value.

C. Wallace Multiplier Using Compressor

An efficient Wallace multiplier can be designed using 4:2 compressor. The number of half adders, full adders and compressor are chosen on number of partial product terms available for the addition. The final product will be generated in the third level by adding the terms in previous stage. The compressor can be designed by using two full adders [21].

D.FIR Filter

In FIR filter, the order will be decided on the calculation of number of filter coefficients required for the implantation of filter. If the order of filter be "N" then "N+1" coefficient are required [22]. The Sampling period (T_S) of FIR filter will be given as equation (4) and (5)

$$T_S = T_M + (N-1) \quad (4)$$

The sampling frequency (F_S) as,

$$F_S = 1/[T_M + (N-1)] \quad (5)$$

III.HARDWARE IMPLEMENTATION

A. Error Sustain Adder

The error sustain adder[23]of 16 bits are subdivided into accurate and inaccurate part of each 8 bits. The accurate part can beimplemented by conventional adders like Ripple carry adder (RCA),Carry select adder[24],Carry look ahead adder[25],Carry skip adder[26] etc.,

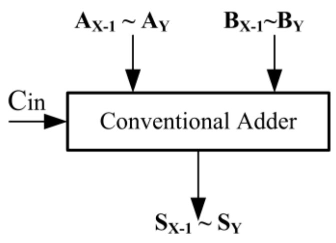

Fig 2. Accurate Part

The inaccurate part plays a vital role in determining accuracy, speed of adders used in FIR. It has two blocks control block and carry free addition block as shown in Fig 3.

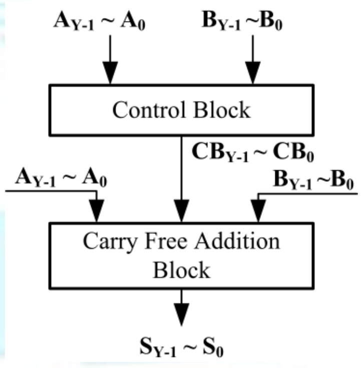

Fig 3.Inaccurate part

The control block has 8 modules for generation of control signal. The block diagram for carry free addition and control block are shownin Fig 4(a), 4(b), 4(c).

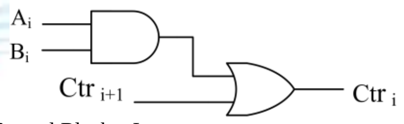

Fig 4.(a) Control Block – I

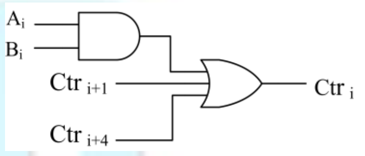

Fig 4.(b) Control Block – II

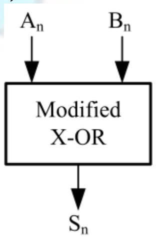

Fig 4.(c) Modified X-OR gate

In the aboveFigure, two types of control block are shown. The inputs are subdivided accurate and inaccurate part. For the inaccurate part, inputs are again subdivided into two parts having 4 bits. The forth bit position must have the control block of type II as shown in the Fig. 4(b). The carry free

addition is carried over with the 8 modified XOR blocks. The inputs to carry free addition block is control signal generated from control signal block.
The schematic of modified XOR gate isshown in Fig. 5 which is different from normal XOR logic. Here the two PMOS transistors P1, P2 and anNMOS transistor N1 are used for deciding the mode of operation of the circuit. The control signal CTL comingfrom respective bit position of CSGC cell as shown in Fig. 4 is given as input respective modified XOR gate.The control signal CTL given as input to transistor P1 and its inverted output from inverter is given to transistorN1 and transistor P2 as shown in Fig.5. The circuit has two modes of operation when control signal is eitherlow or high. When CTL=0 the transistors P1 and N1 both are ON and P2 if OFF. In this mode circuit operates innormal mode of operation as XOR logic does giving the SUM output. When CTL=1 the two transistors P1 andN1 are turned OFF and P2 transistor is turned ON pulling the SUM output to VDD hence the output SUM is setto "1"

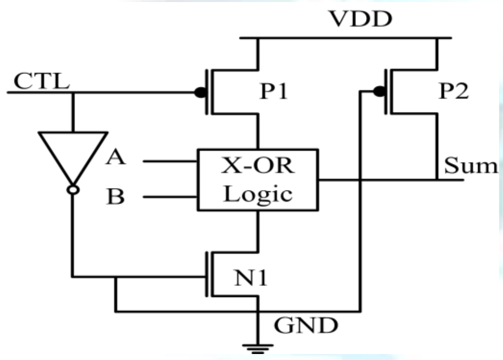

Fig 5. Schematic diagram for Modified XOR gate

B. *Wallace Multiplier Using Compressor*

The conventional Wallace multiplier is designed by using full adder,half adders. The speed of the computation is improved by implementing compressor in the design. The Wallace multiplier using compressor of 4:2 type is shown in Fig 6a.

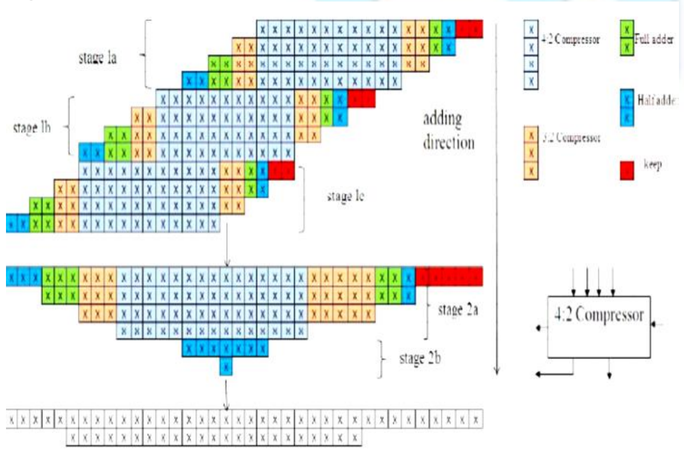

Fig 6a.Wallace tree using compressor

From the above, we may see that Wallace tree structure will be optimized by using 3:2 and 4:2 compressor. For the faster computation, the compressor are approximated with its corresponding pass rate values. We can choose any of the design of compressor on the basis of their pass rate values. The general structure of 3:2and 4:2 compressor is shown in the Fig 6b and 6c respectively.

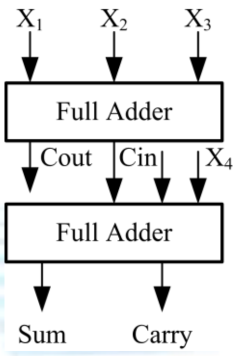

Fig 6b.Schematic diagram of 3:2 compressors

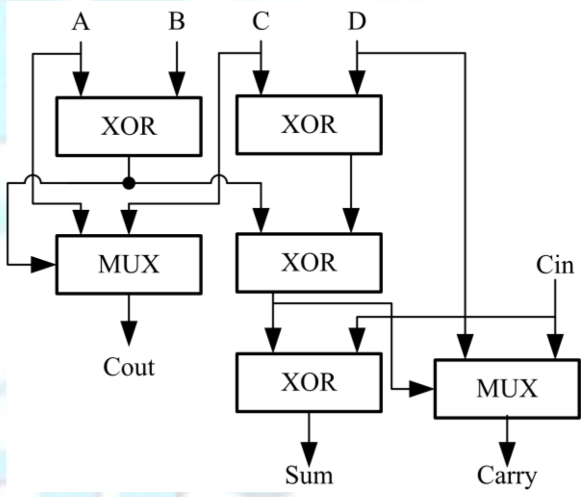

Fig 6c.Schematic diagram of 4:2 compressor

Using the approximated compressor in the Wallace multiplier computational speed is comparably increased. This will make the filter to produce the output in faster manner. Hence we will see the compressor design and then redesigning of Wallace multiplier with 4:2 and 3:2 compressor

C. *Design of Approximated 3:2 Compressor*

Here designing of approximated 3:2 compressor is given. Each design will be consider in the pass values and the number of corrected outputs.

I. Design 1

In this section,design of 3:2 compressor is discussed .It has 3primariy inputs(x,y,z) and output(Y_1,Y_2).This design compress the 3 inputs into 2 outputs with a single approximate equation. As the approximated equations are common for both the outputs,the reuse of same logical functions.This will reduce the number of logic gates,area and power consumption

$$Y_1 = [((x'\oplus z')+(y'\oplus z'((\oplus((x'\oplus z').(y\bullet z))]+x \quad (5)$$

$$Y_2 = [((x'\oplus z')+(y'\oplus z'((\oplus((x'\oplus z').(y.z))] \bullet x \quad (6)$$

From the above equation (5) and (6),we may see that the identical part in the approximated euqtions.This design gives accurate outputs for 7 input out of 8 inputs

II.Design 2

In this design,the approximated equations are further more optimized to have low circuit level complexity.The outputs areproduced with minimal number of gates when compare to Design1.the output are given as

$$Y_1 = [((x'\oplus z')+(y'\oplus z')] \quad (7)$$

$$Y_2 = z' \quad (8)$$

As the outputs uses less numger of logical structures they have minimal occupatioal area,less complexity.This design gives a excat equation for 6 inputs out of 8 outputs

III. Design 3

The approximated outputs are produced by using a simple OR operation in the inputs which gives a simple structure for the output equations.The outputs are

$$Y_1 = y'+z' \quad (9) \qquad Y_2 = x'+z' \quad (10)$$

We may see equation (9) and (10) uses a single logic structure for giving the outputs.This design produces a pass rate of 62.5% and 75% for output Y_1 and Y_2 respectively.

IV. Design 4

Here also,The approximated outputs are produced by using a same equations which gives a simple structure for the output equations.The outputs are in equation (11) and (12)

$$Y_1 = y+z \quad (11) \qquad Y_2 = y+z \quad (12)$$

The equation (11) and (12) uses a single OR logic structure for giving the outputs.This design produces a pass rate of 75% for output Y_1 and Y_2 respectively.Table 1 and 2 shows the outputs of both actual and approximated output. The erroneous output is differentiated by * symbol. From the table, we can count the corrected output for each approximated design for each design given.

Input [2:0]	Actual output O_1	Actual output O_2	Design 1		Design 2	
			Y_1	Y_2	Y_1	Y_2
0	0	0	0	0	0	0
1	1	0	1	0	1	0
2	1	0	1	0	1	0
3	0	1	0	0*	1*	0*
4	1	1	1	1	1	1
5	1	1	1	1	1	1
6	0	1	1*	1	1*	1
7	1	0	1	1	1	1*

Table 1. Truth Table of Design 1 and Design 2

Input [2:0]	Actual output O_1	Actual output O_2	Design 3		Design 4	
			Y_1	Y_2	Y_1	Y_2
0	0	0	0	0	0	0
1	1	0	1	1*	1	1*
2	1	0	1	0	1	1*
3	0	1	1*	1	1*	1
4	1	1	0*	0	0*	0
5	1	1	1	1	1	1
6	0	1	1*	0*	1*	1
7	1	0	1	1	1	1*

Table 2. Truth Table of Design 3 and Design 4

D. Design of Approximated 4:2 Compressor

In this section, a 4:2 compressor is optimized with 4 different logical structures with minimum logical gate functions.

I. Design 1

This design uses a identical logical functions for producing both the approximated output of the compressor. It has 4 inputs (X1, X2, X3, and X4) and outputs (Y_1, Y2) are given in equation (13),(14),(15)

$$Y_1 = (X1\oplus X2\oplus X3\oplus X4) + (X4+Cin) \quad (13)$$

$$Y2 = (X1\oplus X2\oplus X3\oplus X4) \bullet (X4+Cin) \quad (14)$$

$$Cin= (X1'\oplus X2'\oplus X3'\oplus X4') \oplus [(X1'\oplus X2'\oplus X3'\oplus X4') \bullet Cin] \quad (15)$$

Form the above equations, we may observe that the outputs have same identical logic equations which decreases the circuit complexity and re-usage of logic structure. This design produces a accurate outputs for 24 inputs out of 32 input combinations

II. Design 2

Here also a same logical structure is used for giving the approximated equations for giving their output in equation (16),(17),(18)

$$Y_1=(X1\oplus X2\oplus X3\oplus X4) + (X4+Cin) \quad (16)$$

$$Y2 = (X1\oplus X2\oplus X3) \bullet Cin \quad (17)$$

$$Cin= (X1'\oplus X2'\oplus X3'\oplus X4') \oplus [(X1'\oplus X2'\oplus X3') \bullet Cin] \quad (18)$$

In the equation (16) and (17) have the identical optimized logical structure which increases the re-usage and minimal circuit complexity. This proposed design gives accurate results for 29 inputs for 32 input combinations.

III. Design 3

In a conventional 4:2 compressor, approximated outputs are designed by XOR, Multiplexer. But the structure is proposed by 3:2 compressor and XOR gates. The 3:2 compressor given in the design1 of 3:2 compressor will be used for this structure are given in equations (19),(20),(21)

Y_1=[(X1'⊕X3')+(X2'⊕X3')]⊕[(X2•X3)•(X1'⊕X3')] ⊕[X4'⊕ Cin] (19)

Y2 = (X1'⊕X3') +(X2'⊕X3')]⊕[(X2•X3) • (X1'⊕X3')] •[X4'⊕ Cin] (20)

Cin= (X1'⊕X3') ⊕ [(X4'⊕ X2'] (21)

IV. Design 4

The 4:2 compressor may be optimized by using simple XOR and OR operations which gives outputs are in equations (22),(23),(24)

Y_1=(X1'⊕ X3') + (X4'⊕ X2') (22)

Y2 = (X1+Cin) (23)

Cin= (X1'⊕ X3') + (X1'⊕ X2')(24)

For the equations (22)-(24), we get 24accurate outputs for 32 inputs

E.Design of approximated 15:4 and 16:4 compressor

Here a approximated 15:4 and 16:4 compressor may be designed using 4:2 compressor and 3:2 compressor. In 16x16 multiplier, we can use this 15:4 compressor in its partial product optimization. The 16:4 and 15:4 compressor structure with 4:2 and 3:2 compressor are shown below

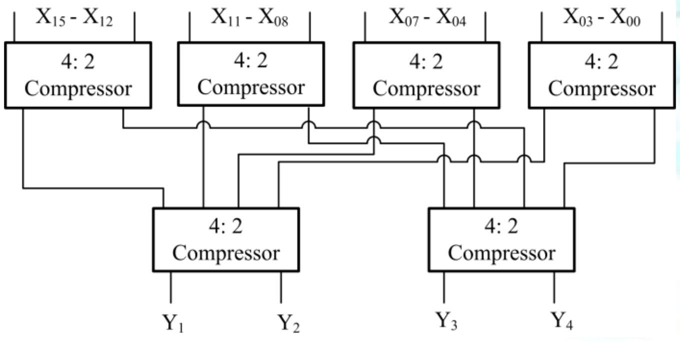

Fig 7. 16:4 compressors using 4:2 compressor

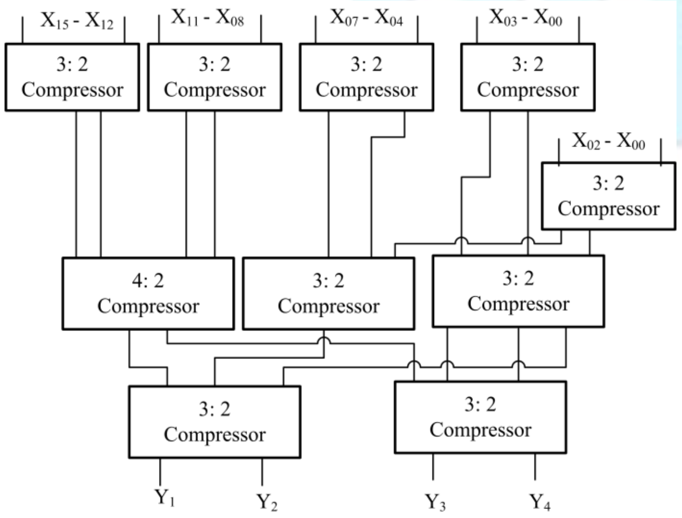

Fig 8. 15:4 compressor using 4:2and 3:2 compressors

The multiplier structure with the approximated 15:4 compressor is shown in the Fig 9. This type of multiplier may also use in the FIR filter in multiplier part of the filter

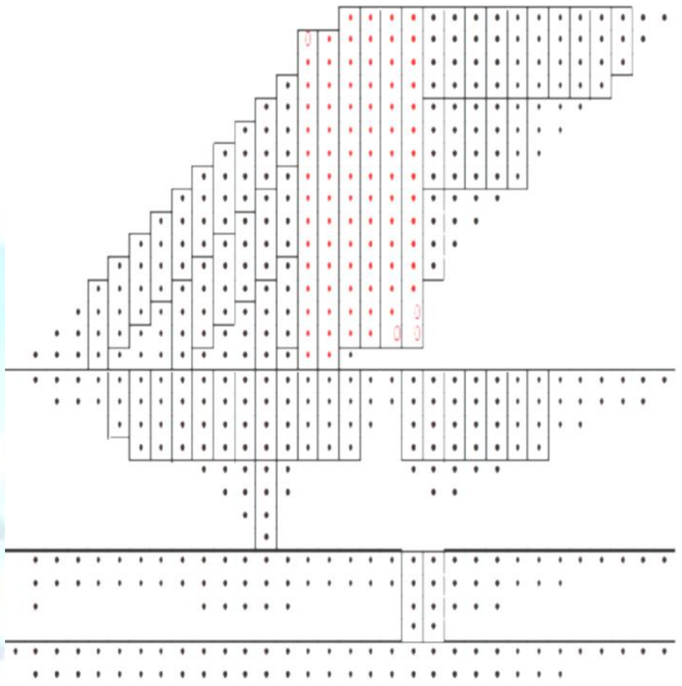

Fig 9. 16x16 multiplier using 4:2and 3:2 compressors

In the above structure, use of 4:2 and 3:2 compressor in the rectangular boxes. The 13th , 14th, 15th shows the usage of 15:4 approximated compressor with zero are added for the incomplete position in the input part

IV. RESULTAND SIMULATION

A .Result of 3:2 Compressors

The approximated 3:2 and 4:2 compressor of different structure were summarized and compared with the conventional logic structure are shown in Table 3 and 4.

3:2 compressor	CPU Time	Pass rate (%)	Gate Delay (ns)
Exact Design	0.63	100	1.92
Design 1	0.45	87.5	1.81
Design 2	0.45	62.5	1.63
Design 3	0.32	69.1	1.42
Design 4	0.32	75	1.42

Table 3. Comparison of 3:2 Compressor

4:2 compressor	CPU Time	Pass rate (%)	Gate Delay (ns)
Exact Design	0.68	100	0.97
Design 1	0.50	62.5	0.82
Design 2	0.52	90.6	0.79
Design 3	0.60	78.1	0.78
Design 4	0.52	62.5	0.79

Table 4. Comparison of 4:2 Compressor

B. Result of 15:4 Compressor

The 15:4 compressor using 4:2 compressor and 3:2 compressor proposed in this paper are compared in the Table 5 and 6.

15:2 compressor (approximate 4:2 compressor of)	CPU Time	LUT	IOB's used	Gate Delay (ns)
Design 1	0.25	10	20	1.82
Design 2	0.23	09	20	1.80
Design 3	0.23	10	19	1.72
Design 4	0.24	10	18	1.72

Table 5. Comparison of 15:4 Compressor (Using 4:2 compressor)

15:2 compressor (approximate 3:2 compressor of)	CPU Time	LUT	IOB's used	Gate Delay (ns)
Design 1	0.23	10	20	0.77
Design 2	0.23	09	20	0.75
Design 3	0.20	10	19	0.74
Design 4	0.20	10	18	0.75

Table 6. Comparison of 15:4 Compressor (Using 3:2 compressor)

A graphical comparison of approximated 3:2 and 4:2 compressor with exact design in terms of pass rate, CPU time, Gate Delay are shown in Fig 10 and Fig 11 respectively.

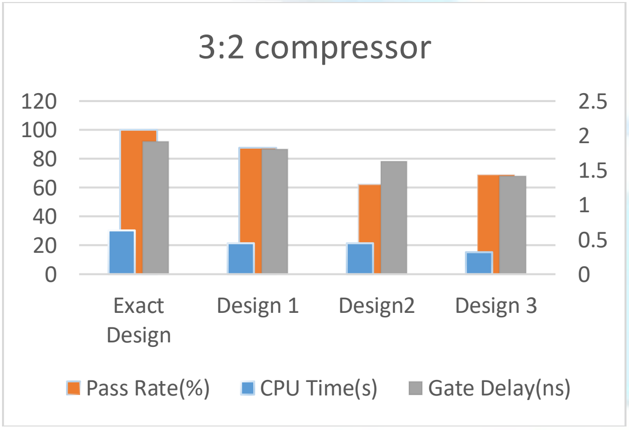

Fig 10. Graphical Representation of 3:2 Compressor

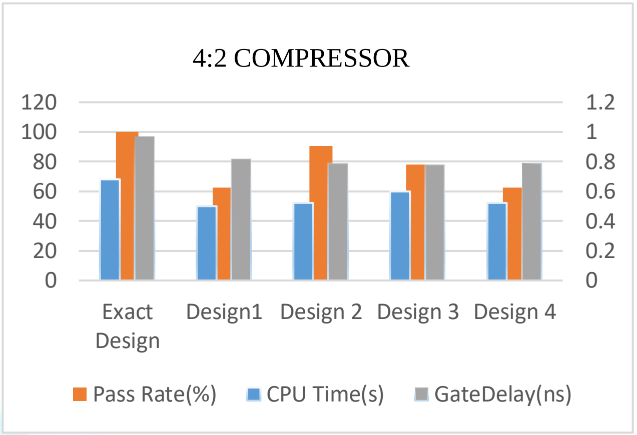

Fig 11. Graphical Representation of 4:2 Compressor

The above graphical representation shows the various parameters results. Depending on the requirements of the architecture parameter corresponding designs will be consider for the structural design for both the compressors. The self –healing 4 –tap FIR filter output is given in the Fig12.

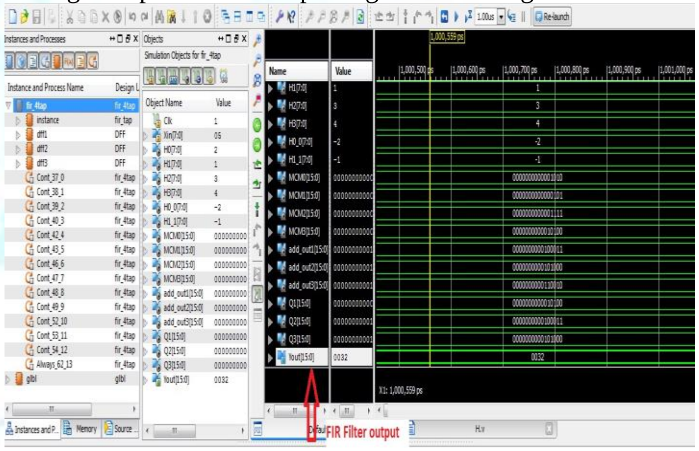

Fig 12. Output of FIR Filter

The comparison output for producing error free output is shown in the Fig 8 with the comparison of filter outputs r1[15:0] and r2[15:0] the error free output of FIR filter is decided are given in Fig13.

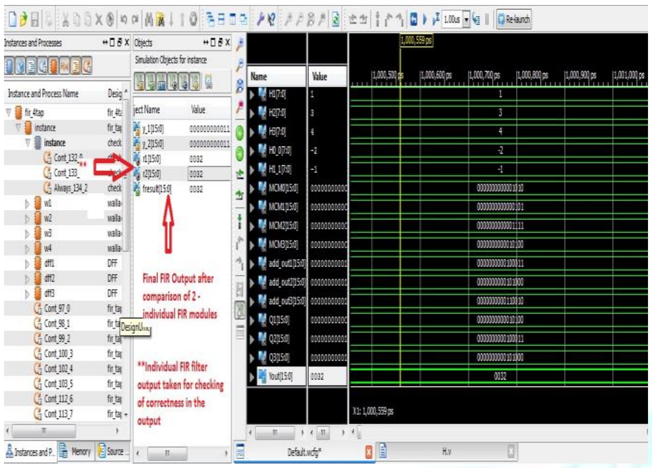
Final FIR Output after comparison of 2 - individual FIR modules
**Individual FIR filter output taken for checking of correctness in the output

Fig 13. Output of FIR filter with self - healing process

V .CONCLUSION

The self – healing 4 –tap FIR filter were designed with error sustain technique and compressor was presented. Also Wallace multiplier are redesigned with approximated compressor are used for the optimization in the logic structureAs per the self-healing methodology, the FIR filter outputs were compared with conventional FIR structure for getting the error free output. The process was carried over on the cost of higher area usage. But the FIR can be able to correctthe output from the error occurred in the output data .The accuracy of 98% is obtained from Error sustain adder. This filter can be extended for N- tap size.

VI. REFERENCES

[1]Anurag Aggarwal, Astha Satija, Tushar Nagpal, " FIR filter designing using Xilinx system generator," International Journal of ComputerApplications, vol. 68, no.11,Aprl. 2013.. 15–64.Aprl. 2013
[2] M.Breuer, "Intelligible test techniques to support error tolerance," in Proc. Asian Test Symp., Nov. 2004, pp. 386–393.
[3] H. Chung and A. Ortega, "Analysis and testing for error tolerant motion estimation," in Proc. Defect and Fault Tolerance in VLSISyst. Symp., 2005, pp. 514–522.
[4] Baker, Louis. VHDL programming with advanced topics. John Wiley & Sons, Inc., 2018.
[5] Botros,N.(2006). HDL programming VHDL and Verilog.Dreamtech press.International Journal of Computer Science Issues, vol.8, 2012.
[6] Fedra, Zbynek, and Jaromir Kolouch. "VHDL procedure forcombinational divider." Telecommunications and Signal Processing (TSP), 2011 34th International Conference on. IEEE, 2011
[7] S.Karthick, S.Karthika, S.Valarmathy, "Design and analysis of low power compressors," International Journal of advanced research inelectrical, electronics and instrumentation engineering, vol.1, Dec. 2012
[8] Gowrishankar V., Manoranjitham D., Jagadeesh P., " Efficient FIR filter design using modified carry select adder & Wallace treemultiplier," International Journal of Science, Engineering and Technology Research, vol.2, March 2013.
[9] S.Karthick, S.Karthika, S.Valarmathy, "Design and analysis of low power compressors," International Journal of advanced research inelectronics and instrumentation engineering, vol.1, Dec. 2012.
[10] N. Ravi1, A.Satish, .T.Jayachandra Prasad, T.Subba Rao, "A new design for array multiplier with trade off in power and area," International Journal of Computer Science Issues, vol.8, May 2011.