Grenze International Journal of Engineering and Technology, Special issue

Realization of Modified Efficient Coding Scheme for Fault Tolerant Parallel Filters

Elizabeth Mathew[1], AbhayaVikraman[2] and Sandhya Venugopal[3]
[1]PG Scholar, Dept. Of ECE (VLSI & ES), ChristKnowledgeCity, Mannoor, India
elizabethattupuram@gmail.com
[2]Assistant Professor, Dept. Of ECE, ChristKnowledgeCity, Mannoor, India
abhayavikraman205@gmail.com
[3]Assistant Professor, Dept. Of ECE, ChristKnowledgeCity, Mannoor, India
sandhyamahendran@gmail.com

***Abstract*—Any life critical missions we implement need to be trustworthy at all the stages of its operation. It has become the greatest need for any system and it was found that it can be achieved by proper fault detection techniques. In order to process signals digital filters are most often integrated in modern digital signal processing systems. According to Moore's law, number of integrated circuits used in signal processing as well as communication systems are becoming more circuitous and their productions are doubling, per year. This triggered the need for fault tolerant executions. This paper replaces the conventional Error Correction Code based protection scheme to an efficient coding scheme for making the Digital FIR filters fault tolerant with less number of redundant filters and to improve the performance of the filters. Fault tolerance requires hardware redundancy and all the existing systems uses redundant modules for that. For any number of filters number of multipliers and redundant modules will be least in the proposed technique.**

***Index Terms*— Efficient Coding Scheme; FIR filter; Hard errors; Redundancy; Soft Errors; Single Event Upsets [SEU]; Triple Modular Redundancy.**

I. INTRODUCTION

Parallel digital filters are frequently used in almost all sectors for digital conversation. In medical area it's used in ECG machines, EEG machines, MRI scanners, and so forth. In army area it's used in RADAR communications, as the quantity of instruments per chip and procedure efficiency has been bettering exponentially, the complexity of electronic circuits has additionally multiplied. In space, clinical, safety and car purposes which need relevant reliability, presence of electronic circuits are abundant. The Moore's law states that the density of transistors in a dense integrated circuit roughly pairs every two years. This expanded intricacy makes the circuit more vulnerable to errors. Errors are mainly classified into soft errors and hard errors. Soft errors are those errors which act for a temporary period of time and which are affecting the data transmitted. Hard error permanently crashes the system. Manufacturing versions and these errors are the key reliability challenges. Undesired outputs are most commonly as a result of error in the circuit. It is the temporary state or transition which inverses the long-established state of the procedure. It's mostly caused by a glitch ensuing at a circuit node because of excessive energy particle striking at that node. These factors make the system more prone to get affected, in some cases it may lead to catastrophic failures too. A system

Grenze ID: 01.GIJET.3.2.27

collapse when a real running system diverges from its specific behavior. The reason of flop or system crash is called an error. An error can be invalid system state, one that is not acceptable by the system behavior requirements. The error itself is the outcome of failing in the system or fault Thus the concept of fault tolerance has emerged. Fault tolerance is the capability of system to handle with core errors. The idea of fault tolerance is to boost reliance of system. Fault is the rudimentary cause of a system failure. That means an error is only as specified and nothing more than the sign of fault. A fault could not constantly results in an error, but the same fault may outcome in numerous errors. Similarly a single error may raise a numerous failures. This paper discusses on ways to protect the system from single errors.

If any excessive energy alpha element hits a circuit node the energy will get relocated to the circuit node, which results in spurious transition and might change the logical value of a circuit node by means of creating a momentarily error that can impact the approach operation. Silicon on Insulator (SOI) which is a particular manufacturing process was once used to preserve the circuits. Most important breakthrough for security of the circuits was the ideology to add redundancy.

II. Literature Review

Triple Modular Redundancy (TMR) is the most accustomed technique which makes the scheme three folds the system and adds voting logic to correct errors. Fundamentally, a TMR system includes of three indistinguishable devices and voting logic. The voting logic is the mainstream voter which takes one of the high priority output values. It is the best example of a technology which uses the notion of redundancy. Redundant modules are additional systems which are used to protect the desired system. Inline to the defect in system, individual device may have an error inside and generate a different output. [4]This irregularity will be trapped and fixed by voting logic .Thus the chosen output continually correct value under the supposition of single error. Thus, the voted output is always a correct value under the assumption of single error. When the TMR notion is applied to a workstation, all output signal of the CPU are voted therefore no error should exist at output of voters. Any error that occurs exemplify that one of the CPUs has an error inside. If that error is not corrected by some way it might result in more errors and finally become unrecoverable. As it triples the area and power of the circuit, it is not satisfactory in some applications.

Another method [5] is to use the algorithmic properties of the circuit to detect or correct errors which can reduce the overhead required to guard the circuit. This is referred to as Algorithm-Based Fault Tolerance (ABFT). Algorithm based computation schemes like FFT and QR factorization are protected using this scheme. QR factorization is a computation scheme where Q is an orthogonal matrix and R is a tridiagonal matrix. In that scheme Parseval theorem has been taken as the basic principle for the error detection and correction. For the Fast Fourier Transform the sum of square of response of inputs was checked for equality with sum of square of output which was stated as the Parseval’s check. This was also termed as sum of square check or SOS check.

In [5] Nicolaidis stated that mitigation of soft errors were the main concern and there various technologies were proposed like Self-checking design, Error masking design, Error trapping design which was based on Muller C element. All these schemes discusses on error detection only. Later Shim came with the concept of Algorithmic soft error tolerance. Three distinct techniques were proposed [7] and compared their protection efficiency. The three schemes were classified based on the arrangement of estimators of the scheme. Estimators were added to protect the system from errors. The Fine Grain Soft Error Tolerance and Sub word Detection processing schemes were explored in [8] which exploits the concept of logic masking. In [10] the filters were designed distinctly in different structural form one is transpose form and other is cascaded form to check for any errors.

Over the years as the technology has been scaled from about 1300nm to about 10 nm technology, large number of transistors are integrated on a single chip. Complex signal processing applications and biomedical applications uses parallel digital FIR filters for the processing of large number of signals. In this paper a most efficient protection scheme with minimal hardware utilization is explored. For many applications signal processing circuits are well suited. These circuits mainly include parallel digital FIR filters. FIR filters are most often chosen over IIR filters as they have regular structures and are stable at linear phase. Over the years many ABFT techniques [3] have been proposed to protect the basic blocks that are commonly used in those circuits. Parseval theorem [4] was one of the most basic algorithms used to check the error rate in a system.

In bridging concurrent and noncurrent error detection multipliers are replaced by constant shifting [9] this idea was explored to make a new scheme for protection .An arbitrary matrix designed [2] to protect the parallel filters with less number of redundant modules was also used in this scheme.

III. DIGITAL FILTERS

Digital filters are employed using a particular specific dedicated digital hardware or digital personal computer. Commonly speaking, digital filters have become the focus of attention over the last 40 years. The interest in digital filters started with the introduction of the digital computer, especially the reasonably priced PC and special purpose signal processing boards. People who led the way in the work mainly the analysis part were Kaiser, Gold and Radar. A digital filter is merely the realization of an equation in computer software. There are no resistor, capacitor, and inductor{R, L, and C} modules as such. However, digital filters can also be built directly into special purpose computers in hardware form. But the execution is still in software.

Computers and computer solicitations use digital signals so they most often use parallel digital filters for their processing. A digital filter can be realized with an FIR filter or an IIR filter operation. FIR filters are most commonly used because it has high stability. This work addresses an efficient fault tolerance technique to protect configurations of parallel FIR filters.

FIR filter can be implemented using the following equation.

$$y[\mathrm{n}] = \sum_{l=0}^{N}(x[n-l].h[l]) \qquad (1)$$

Where x[n] is the input signal, y[n] is the output signal and h[l] is the impulse response of the filter. N is the order of the filter. Order represents the number of delay elements used in the filter. Linearity property is the most important property of digital filters which is exploited in this scheme. It states that sum of any combination of the outputs can be obtained by adding corresponding inputs and filtering the inputs with the same impulse response of the filter. It is explained in equation (2).

$$Y1[\mathrm{n}] + Y2[n] = \sum_{l=0}^{\infty}((\mathrm{X1}[\mathrm{n}-\mathrm{l}] + \mathrm{X2}[\mathrm{n}-\mathrm{l}]).\mathrm{h}[\mathrm{l}]) \quad (2)$$

The FIR filter can be implemented with direct form or tanspose form realization. The proposed technique addresses transposed form implementation and the simple observations shown in (1).

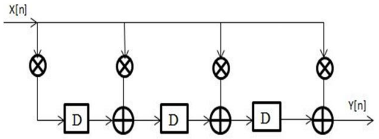


Fig 1 : Transpose form implementation of FIR filter

IV. CONVENTIONAL TECHNIQUE

The protection of parallel filters was done using Error Correction Codes basically hamming codes. Here Gao [1] has proposed error correction for parallel FIR digital filters using Hamming code in which single parallel FIR filter is taken as a bit in ECC technique.

A. Four Parallel Filter Protection

TABLE I.

Information Bits	Parity Bits
4	3
8	4
12	5
27	6

Hamming codes are primarily used to locate whether any transmitted bit is in error and to correct it, so that error free bits are expected at the receiver. To protect information bits to be transferred from errors Hamming codes transmit some number of parity bits along with the information bits. The number of parity bits to be added is based on the Hamming rule.

$$r + p + 1 \geq 2^{p} \quad (3)$$

So according to these remarks to protect four information bits from errors three parity bits have to be supplemented. It is explained in Table 1. With the same concept of hamming codes erroneous outputs and faulty filter can be corrected.

TABLE II: FAULT DETECTION

S1	S2	S3	Faulty Filter	Action
0	0	0	None	None
1	1	1	F1	Correct F1
1	1	0	F2	Correct F2
1	0	1	F3	Correct F3
0	1	1	F4	Correct F4
1	0	0	F5	Correct F5
0	1	0	F6	Correct F6
0	0	1	F7	Correct F7

Here in the conventional scheme inputs are represented as Xin1, Xin2, Xin3, and Xin4 these are processed through filters with same response arranged in parallel. Considering equation1 the responses of the given inputs are Yout1, Yout2, Yout3, and Yout4. For example

$$\text{Yout1}=\sum_{l=0}^{N}(x1 \cdot h[l]) \quad (4)$$

Inputs to the redundant filters were veiled according to the hamming codes and its reaction is given as

$$\text{Q1 [n]} = \sum_{l=0}^{N}(x1 + x2 + x3)h[l]) \quad (5)$$

Where Q1[n] is the output of the first redundant filter and similarly Q2[n] and Q3[n]. These equations of the output of the redundant filters are taken to check whether there is any error in the output or to find any faulty filter. That is expressed as

$$\text{Q1 [n]} = \text{Y1+Y2+Y3} \quad (6)$$

Equations 5 and 6 are then equated and checked for equality. Similarly Q2 [n] and Q3 [n] are also equated to corresponding sum of outputs. Thus there are three sets of equations which were expressed as

$$\text{Q2 [n]} = \text{Y1+Y2+Y4} \quad (7)$$

$$\text{Q3 [n]} = \text{Y1+Y3+Y4} \quad (8)$$

Now these equations 6, 7 and 8 were checked for equality and if all the three were not satisfying the equations then first filter is faulty so as to produce an undesired output. And if 6 and 7 are not satisfied then filter 2 is faulty. Filter 3 is faulty if 6 and 8 are not gratified. After finding the faulty filter we correct it by reconstructing the outputs. The reconstructed outputs are:

$$Y_{c1}\text{ [n]} = \text{Q1 [n]} - \text{Y2} - \text{Y3}. \quad (9)$$

$$Y_{C2}\text{[n]} = \text{Q2 [n]} - \text{Y1} - \text{Y4}. \quad (10)$$

$$Y_{C3}\text{[n]} = \text{Q3 [n]} - \text{Y1} - \text{Y4}. \quad (11)$$

$$Y_{C4}\text{[n]} = \text{Q2 [n]} - \text{Y1} - \text{Y2}. \quad (12)$$

B. Eight Parallel Filter Protection

This scheme takes eight parallel filters to be protected from faults and faulty outputs based on hamming codes. Here based on Table III to protect eight filters four redundant filters have to be added. In this scheme (12, 8) Hamming codes is considered. As already mentioned Hamming codes can be constructed in linear algebra terms through matrices because hamming codes are linear codes. Hamming code considered in this eight parallel filter protection scheme, code generator matrix for computing parity bits are given by:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

In this case, the four redundant modules are added similar to parity check bits added to protect information bits. The inputs to the redundant modules are X9, X10, X11, and X12 are computed as a function of the data bits X1, X2, X3, X4, X5, X6, X7 and X8. '⨁' operation is similar to addition and it exactly represent modulo 2 addition. From the code generator matrix parity bits computed are coded based on equations given below:

$$X\,9 = X1 \oplus X2 \oplus X4 \oplus X5 \oplus X7 \qquad (13)$$
$$X\,10 = X1 \oplus X3 \oplus X4 \oplus X6 \oplus X7 \qquad (14)$$
$$\mathrm{X}\ 11 = X2 \oplus X3 \oplus X4 \oplus X8 \qquad (15)$$
$$X\,12 = X2 \oplus X3 \oplus X4 \oplus X8 \qquad (16)$$

The redundant filters added are considered as Check filters and their outputs are given by:

$$Z1[\mathrm{n}] = \sum_{l=0}^{\infty}((X9[n-l]).h[l]) \qquad (17)$$
$$Z2[\mathrm{n}] = \sum_{l=0}^{\infty}((X10[n-l]).h[l]) \qquad (18)$$
$$Z3[\mathrm{n}] = \sum_{l=0}^{\infty}((X11[n-l]).h[l]) \qquad (19)$$
$$Z4[\mathrm{n}] = \sum_{l=0}^{\infty}((X12[n-l]).h[l]) \qquad (20)$$

These equations can be rewritten as a response of sum of inputs and to explain it Z1 is considered:

$$Z1[\mathrm{n}] = \sum_{l=0}^{\infty}((\mathrm{X1[n-l] + X2[n-l] + X4[n-l] + X5[n-l] + X7[n-l]}).\mathrm{h[l]}) \quad (21)$$

Similarly Z2, Z3 and Z4 can also be written as the sum of response of inputs. Based on linearity property explained in (2). Equation (21) can be formulated as

$$\mathrm{Y1[n]+Y2[n]+Y4[n]+Y5[n]+Y7[n]=}$$
$$\sum_{\mathrm{l=0}}^{\infty}((\mathrm{X1[n\text{-}l]+X2[n\text{-}l]+X4[n\text{-}l]+X5[n\text{-}l]+X7[n\text{-}l]).h[l]})\quad (22)$$

Consequently it can be equated to crisscross for any faults occurred. By examining whether

$$Z1[n] = \mathrm{Y1[n]+Y2[n]+Y4[n]+Y5[n]+Y7[n]} \qquad (23)$$

Correspondingly comparing linearity property and the response of check filters

$$Z2[n] = \mathrm{Y1[n]+Y3[n]+Y4[n]+Y6[n]+Y7[n]} \qquad (24)$$
$$Z3[n] = \mathrm{Y2[n]+Y3[n]+Y4[n]+Y8[n]} \qquad (25)$$
$$Z4[n] = \mathrm{Y5[n]+Y6[n]+Y7[n]+Y8[n]} \qquad (26)$$

The comparisons are considered as S1, S2, S3, S4 checks and if that comparison fulfills the condition a zero value is allotted to that check. If it does not fulfill value one is ascribed. It means that if all the checks S1, S2, S3 are zeros then there would be no fault in any of the filters. Reconstructing Table II to locate which filter is faulty is given by Table III.

TABLE III. FAULT LOCATION OF FILTERS OUTPUTS BASED ON HAMMING CODE

S1	S2	S3	S4	Faulty Output	Action
0	0	0	0	No Error	None
1	1	0	0	Y1	Correct Y1
1	0	1	0	Y2	Correct Y2
0	1	1	0	Y3	Correct Y3
1	1	1	0	Y4	Correct Y4
1	0	0	1	Y5	Correct Y5
0	1	0	1	Y6	Correct Y6
1	1	0	1	Y7	Correct Y7
0	0	1	1	Y8	Correct Y8

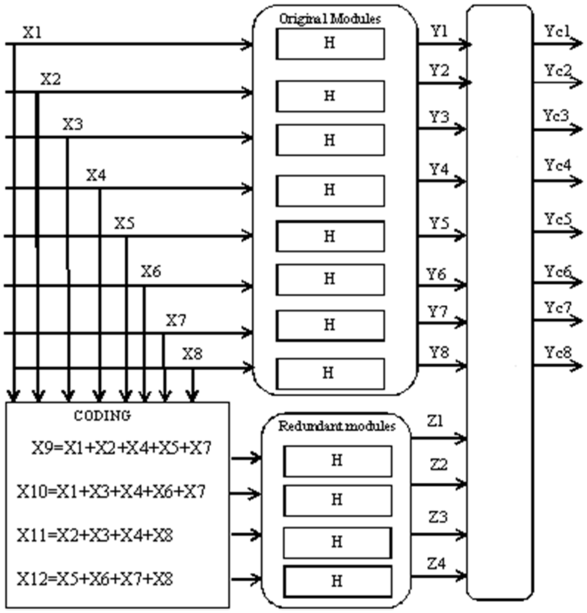


Fig. 2: Proposed scheme for protecting eight parallel filters

Fault correction is achieved by restructuring the inaccurate outputs using the rest of the data and check outputs. For example, when an error on Y1 is detected, it can be corrected by making

$$Yc1[n] = Z1[n] - (Y2[n] + Y4[n] + Y5[n] + Y7[n]) \qquad (27)$$

Similarly when other outputs are faulty they are reconstructed using the equations

$$Yc2[n] = Z1[n] - (Y1[n] + Y4[n] + Y5[n] + Y7[n]) \qquad (28)$$
$$Yc3[n] = Z2[n] - (Y1[n] + Y4[n] + Y6[n] + Y7[n]) \qquad (29)$$
$$Yc4[n] = Z3[n] - (Y2[n] + Y3[n] + Y8[n]) \qquad (30)$$
$$Yc5[n] = Z4[n] - (Y6[n] + Y7[n] + Y8[n]) \qquad (31)$$
$$Yc6[n] = Z4[n] - (Y5[n] + Y7[n] + Y8[n]) \qquad (32)$$
$$Yc7[n] = Z4[n] - (Y5[n] + Y6[n] + Y8[n]) \qquad (33)$$
$$Yc8[n] = Z4[n] - (Y2[n] + Y3[n] + Y4[n]) \qquad (34)$$

To ensure that single errors in the encoding logic will not affect the data outputs, one option is to avoid logic sharing by computing each of the Zi independently. With the two configurations of filters we have studied how filters arranged parallel can be protected from faults by using error correction codes.

V. Arbitrary Matrix based Scheme

In order to reduce the area of the system an arbitrary matrix was designed An Arbitrary matrix is designed and coded based on a scheme proposed by Zhen Gao [2]. This reduces the protection overhead and makes the number of redundant filters independent of the sum of parallel filters. The error can be corrected by taking the final outputs from a set that does not include a single filter. This scheme can also be referred to as Efficient Coding Scheme.

C. Four Parallel Filter Protection

The error correction and detection logic can be simplified assuming that there is only a single error. Here Error correction is based on the arbitrary matrix. For the protection of four parallel filters a 4×6 coding matrix is utilized.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \\ a_{51} & a_{52} & a_{53} & a_{54} \\ a_{61} & a_{62} & a_{63} & a_{64} \end{bmatrix}$$

Formula of the input signal to the ith filter is of the form $v_i[n] = a_{i1}x_1[n] + a_{i2}x_2[n] + a_{i3}x_3[n] + \cdots$ (13)for (i = 1, 2, …, or 6). Therefore inputs to redundant filters are given by:

$$X5=x1+x2+x3+x4 \qquad (35)$$
$$X6=x1+2x2+3x3+4x4 \qquad (36)$$

And the outputs of Redundant Filters rendering to linearity property is

$$Z5=Z1+Z2+Z3+Z4 \qquad (37)$$
$$Z6=Z1+2Z2+3Z3+4Z4 \qquad (38)$$

Fault detection was based on taking the filters as a set of five filters including the redundant filters and omitting one of the filters and then checking for any faults.

$$\begin{matrix} y1 \\ y2 \\ y3 \\ y4 \end{matrix}_{1235} = [A_{1235}]^{-1} \begin{matrix} Z1 \\ Z2 \\ Z3 \\ Z4 \end{matrix} \qquad (39)$$

$$\begin{matrix} y1 \\ y2 \\ y3 \\ y4 \end{matrix}_{1236} = [A_{1236}]^{-1} \begin{matrix} Z1 \\ Z2 \\ Z3 \\ Z4 \end{matrix} \qquad (40)$$

$$\begin{matrix} y1 \\ y2 \\ y3 \\ y4 \end{matrix}_{2345} = [A_{2345}]^{-1} \begin{matrix} Z1 \\ Z2 \\ Z3 \\ Z4 \end{matrix} \qquad (41)$$

$$\begin{matrix} y1 \\ y2 \\ y3 \\ y4 \end{matrix}_{2346} = [A_{2346}]^{-1} \begin{matrix} Z1 \\ Z2 \\ Z3 \\ Z4 \end{matrix} \qquad (42)$$

So if the comparisons 39 ≠ 40 and 41 = 42 means among the filters 12356 there is an error or fault and 23456 are correct, this indicate that the 1stFilter is faulty. Similarly if the filters 1235≠1236 and 1345=1346 then 2nd Filter is faulty.1235 ≠ 1236 and 1245=1246 therefore 3rd Filter is faulty 1245 ≠ 1246 and 1235=1236 4th Filter is faulty. In order to find the fault detection table we need to calculate $[A_{1235}]^{-1}$,$[A_{1236}]^{-1}$, $[A_{2345}]^{-1}$,
$[A_{2346}]^{-1}$,$[A_{1345}]^{-1}$, $[A_{1346}]^{-1}$ $[A_{1245}]^{-1}$ $[A_{1246}]^{-1}$.

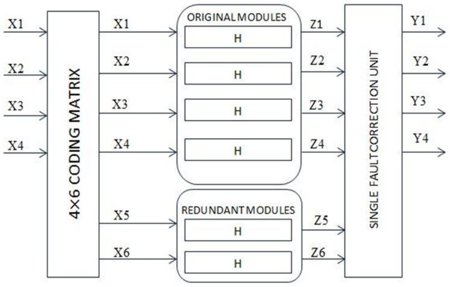


Fig 3: Efficient Coding Scheme for 4 parallel filters

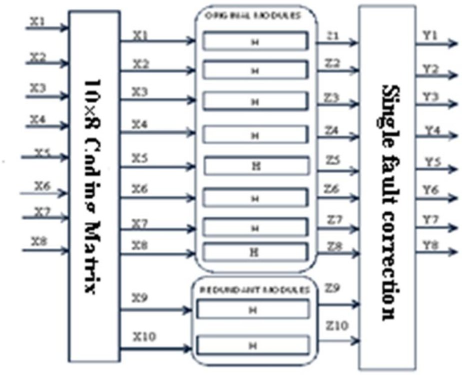


Fig 4 : Efficient Coding Scheme for 8 parallel filters

And the scheme was based on following steps:

- First find all sub matrixes from the arbitrary matrix.
- Find inverse of those sub matrixes.
- Then calculate Check matrix by multiplying the inverse sub matrix with unit matrix and taking the difference of two corresponding equations.
- Check matrix is then multiplied by the inverse of Z matrix.
- Thus calculated the error probability variable ‘e’. Thus e1, e2, e3, e4 are calculated.
- Errors are located based on the error probability variable.

e1	e2	e3	e4	Faulty Output	Action
0	0	0	0	No Error	None
0	#	#	#	Y1	Correct Y1
#	0	#	#	Y2	Correct Y2
#	#	0	#	Y3	Correct Y3
#	#	#	0	Y4	Correct Y4
#	#	#	#	Y5	Correct Y5
#	#	#	#	Y6	Correct Y6

In a practical implementation, the first four rows of the matrix would be an identity matrix so that the inputs to the original. Table III: Fault Detection Using Efficient Coding Scheme filters are the incoming signals. First 4× 6 coding matrix has been used to make the four parallel systems fault tolerant. Later with the same concept and using 8× 10 coding matrix 8 parallel filters are protected.

D. Eight Parallel Filter Protection Scheme.

Similarly for protecting eight parallel filters using efficient coding scheme only two redundant filters are used. Here a 10×8 Coding matrix is used for coding inputs to the redundant filters. For the matrix first eight rows would be taken as an 8×8 identity matrix. The remaining ninth row has all Figs: Its elements as 1. And the tenth row have 1,2,3,4,5,6,7,8 as its elements.

VI. Proposed Scheme

The multipliers which are used to multiply the constant numbers with the inputs to the original filters were replaced with shifters. Multiplication replaced by constant shifting was considered. This can be referred to as modified Efficient Coding Scheme. This scheme can be referred to as the modified efficient Coding Scheme. When the proposed scheme was implemented using Xilinx Vivado IDE the device utilization features obtained were noted down in Table IV.

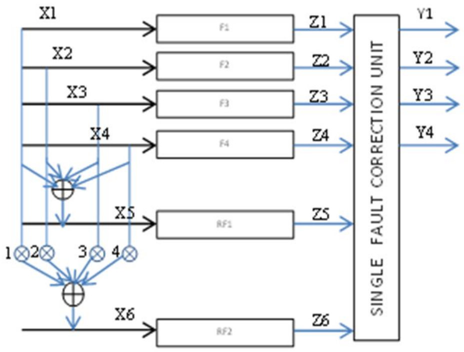


Fig 5: Exact Realization of Arbitrary Matrix based Scheme

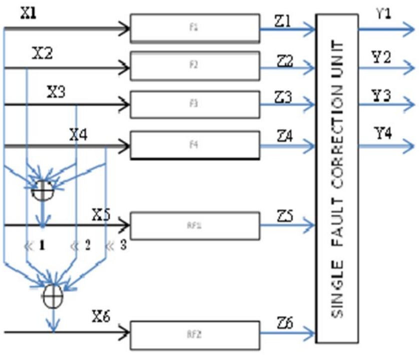


Fig 6: Proposed Scheme

TABLE IV: DEVICE UTILIZATION OF VARIOUS PROTECTION SCHEME

	System to be Protected	Input Bit Width	Protection Schemes	Number of Slice Registers	Number of 4 input Slice LUT's	Number of LUT FF pairs	Number of Bonded IOBs
3 ORDER FILTERS	FOUR PARALLEL FIR FILTERS	4 BIT INPUT	ECC scheme	224	454	206	49
			ECS scheme	152	430	144	49
			Modified ECS scheme	152	405	143	49
		8 BIT INPUT	ECC scheme	375	910	309	97
			ECS scheme	264	856	241	97
			Modified ECS scheme	264	756	241	97
	EIGHT PARALLEL FIR FILTERS	4 BIT INPUT	ECC scheme	432	863	392	97
			ECS scheme	310	1056	288	97
			Modified ECS scheme	318	1088	295	97
		8 BIT INPUT	ECC scheme	791	1936	681	193
			ECS scheme	534	2083	485	193
			Modified ECS scheme	518	1974	469	193

VII. Analysis and Simulation Results

The designs proposed have been realized and charted to a device. The schemes based on the ECC scheme and the matrix based scheme has been implemented on Xilinx spartan6 FPGA and evaluated both in terms of overhead and error coverage. Very High Speed Integrated Circuit Hardware Description Language is used to code for the implementation. It is used to implement the FIR filter structure either using three of its abstraction models. Here chosen model was structural implementation. For the redundant Filter, the bit widths are extended to 10 and 18 bit, respectively. Since both the inputs and outputs to the filters are sequential, the linearity property check is done. The input bit width was varied as 4 bit, 8 bit and 16 bit and also the order of the filter has also been varied from 3 to 5.

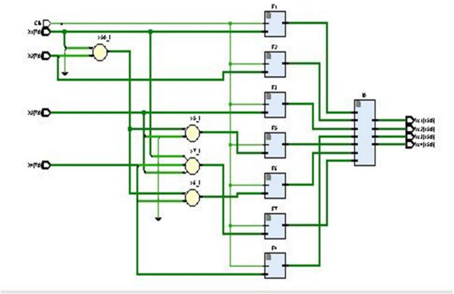


Fig 7: Schematic Representation Of four Parallel Filter Protection Scheme Using Error Correction Codes

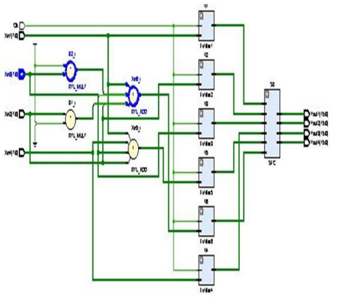


Fig 8: Schematic Representation of Four Parallel Filter Protection Scheme based on the Proposed Scheme using Xilinx Vivado

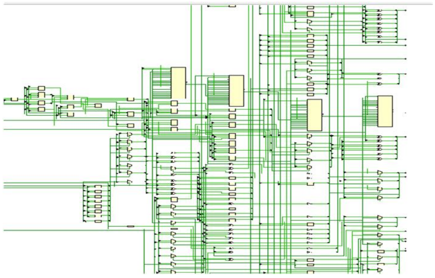
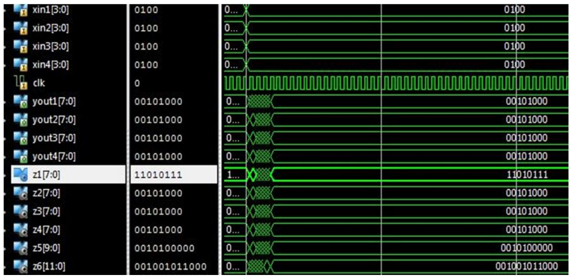


Fig 9: Pin Plan Ahead Representation of Four Parallel Filter Protection Scheme Using modified Coding Scheme



```
xin1[3:0]     0100           0...   0100
xin2[3:0]     0100           0...   0100
xin3[3:0]     0100           0...   0100
xin4[3:0]     0100           0...   0100
clk           0
yout1[7:0]    00101000       0...   00101000
yout2[7:0]    00101000       0...   00101000
yout3[7:0]    00101000       0...   00101000
yout4[7:0]    00101000       0...   00101000
z1[7:0]       11010111       1...   11010111
z2[7:0]       00101000       0...   00101000
z3[7:0]       00101000       0...   00101000
z4[7:0]       00101000       0...   00101000
z5[9:0]       0010100000     0...   0010100000
z6[11:0]      001001011000   0...   001001011000
```

Fig 10: Behavioral Simulation of “four parallel filters” protection scheme using MODIFIED efficient coding scheme with fault injected at Z1 using Xilinx Vivado tool

The implementation of the FIR filter core shown in Fig. 7. using the Xilinx Vivado tool. The inputs are 8-bit wide and the outputs are 16-bit wide and these bit widths have been varied for better analysis too. In each simulation run, one error is inserted to imitate the behavior of a tender error that occurs in segregation. Schematics are generated using Xilinx Vivado tool.

VIII. CONCLUSIONS

A novel technique to execute fault tolerant parallel FIR digital filters has been proposed in this paper. The first part of the paper discusses on various faults occurring in a system and the special techniques which are used to mitigate them. The signal processing and computing technologies became inimitable tools of modern society. The boundless benefits they bring for the welfare of the mankind, the greater the potential for harm when they fail to perform their functions or perform them incorrectly. As digital FIR filters are an inevitable component in signal processing applications, it should be protected from faults. Single Event Effects have become an intensifying limitation of the consistency of electronic components.

The second part focuses on the error correction code based scheme of protection, efficient coding scheme based protection as well as the modified efficient coding scheme based protection. Different protection schemes are modeled using Xilinx Vivado tool. The variations in the output are discussed.

The anticipated scheme manipulates the linearity of filters to implement an error correction method. Here inputs of two redundant filters which are linear combinations of the original filter inputs are used to detect and locate the errors. The previously proposed technique was based on the use of Error Correction Codes (ECCs). This method considers each filter as a bit in the ECC. The proposed scheme beats the ECC technique (similar fault-tolerant capability with lower cost). Therefore, the proposed scheme can be useful to implement fault tolerant parallel filters.

REFERENCES

[1] Zhen Gao, Pedro Reviriego, Wen Pan, Zhan Xu, Ming Zhao,Jing Wang, and Juan Antonio Maestro "Fault Tolerant Parallel Filters Based on Error Correction Codes" IEEE Transactions on Very Large Scale Integration (VLSI) systems, February 23, 2015.

[2] Zhen Gao, Pedro Reviriego, Wen Pan, Zhan Xu, Ming Zhao,Jing Wang, and Juan Antonio Maestro "Efficient Coding Scheme For Fault Tolerant Parallel Filters" IEEE Transactions on Circuits and Systems II, July 23, 2015.

[3] Zhen Gao, Pedro Reviriego, Wen Pan, Zhan Xu, Ming Zhao,Jing Wang, and Juan Antonio Maestro "Fault Tolerant Parallel FFTs Using Error Correction Codes and Parseval Checks" IEEE Transactions on Very Large Scale Integration (VLSI) systems, February 26, 2015.

[4] Jakob Lechner, "Designing Robust GALS Circuits with Triple Modular Redundancy", Ninth European Dependable Computing Conference, 2012.

[5] A. Reddy and P. Banarjee "Algorithm-based fault detection for signal processing applications," IEEE Trans. Comput., vol. 39, no. 10,pp. 1304–1308, Oct. 1990.

[6] B. Shim and N. Shanbhag, "Energy-efficient soft error-tolerant digital signal processing," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 14, no. 4, pp. 336–348, Apr. 2006.

[7] T. Hitana and A. K. Deb, "Bridging concurrent and non-concurrent error detection in FIR FFT's," in Proc. Norchip Conf., 2004, pp. 75–78.

[8] Y.-H. Huang, "High-efficiency soft-error-tolerant digital signal processing using fine-grain subword-detection processing," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 2, pp. 291–304, Feb. 2010.

[9] S. Pontarelli, G. C. Cardarilli, M. Re, and A. Salsano, "Totally fault tolerant RNS based FIR FFT's," in Proc. IEEE IOLTS, Jul. 2008, pp. 192–194.

[10] P. Reviriego, C. J. Bleakley, and J. A. Maestro, "Strutural DMR: A Technique for Implementation of Soft-Error-Tolerant FIR Filters," IEEE Trans. on Circuits and Systems-II: Express Briefs, vol. 58, no. 8, pp. 512-516, Aug. 2011.

[11] P. Reviriego, S. Pontarelli, C. Bleakley and J.A. Maestro, "Area Efficient Concurrent Error Detection and Correction for Parallel Filters", IET Electronic Letters, vol. 48, no 20, pp. 1258-1260, Sept. 2012.

[12] Z. Gao, P. Reviriego, W. Pan, Z. Xu, M. Zhao and J. Wang, J.A. Maestro, "Fault Tolerant Parallel Filters based on Error Correction Codes", IEEE Trans. on Very Large Scale Integration Systems, vol.23, no.2, pp.384-387, Feb. 2015.

[13] R. W. Hamming, "Error correcting and error detecting codes," Bell Sys. Tech. J., vol. 29, pp. 147–160, Apr. 1950.

[14] A. Chatterjee, and M. A. d'Abreu, "The Design of Fault-Tolerant Linear Digital State Variable Systems: Theory and Techniques", IEEE Transactions on Computers, vol. 42, no. 7, July 1993.

[15] C. N. Hadjicostis, "Coding Approaches to Fault Tolerance in Dynamic Systems", PhD Thesis, MIT, 1999.

[16] P.P Vaidyanathan. "Multirate Systems and Filter Banks", Prentice Hall, 1993.

[17] A. Sibille, C. Oestges and A. Zanella "MIMO: From Theory to Implementation", Academic Press, 2010.

[18] N. Kanekawa, E. H. Ibe, T. Suga and Y. Uematsu, "Dependability in Electronic Systems: Mitigation of Hardware Failures, Soft Errors, and Electro-Magnetic Disturbances", Springer Verlag, 2010.