IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 9, September 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.423

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 9, September 2023 ||

| DOI:10.15680/IJIRSET.2023.1209075 |

Project Based on an Algorithmic-Based Fault Detection Technique for the 1-D Discrete Cosine Transform

N.Thamilarasi[1], K.R.Badriesh[2]

Assistant Professor, Department of Electronics and Communication Engineering, Jaya Engineering College, Thiruninravur, Tamilnadu, India[1]

PG Student, Department of Electronics and Communication Engineering, Jaya Engineering College, Thiruninravur, Tamilnadu, India[2]

ABSTRACT: The discrete cosine transform (DCT) is a key building block for many applications in communications and signal processing. Likewise, it is also popular in space applications such as those that perform audio or image compression. The problem with space applications is that they usually have to work in a high radiation environment that affects electronic components and distorts their correct functionality. Therefore, it is usual to devise alternative implementation of the designs that can detect the presence of errors and discard the affected samples. In this brief, we explore the use of algorithmic-based fault tolerance (ABFT) techniques, which exploit certain algorithmic properties to detect errors. In particular, an ABFT technique for the Arai DCT is proposed and compared to standard protection schemes based on modular redundancy. Experimental results show that important savings in terms of resource overhead can be obtained with our approach while still maintaining the error detection rate at a reasonable level.

KEYWORDS: Configuration memory, discrete cosine transform (DCT), field-programmable gate array (FPGA), image compression, soft error.

I. INTRODUCTION

Since the beginning of space exploration, radiation has posed serious problems to the electronics onboard satellites and other spacecraft [1]. From temporary failures to permanent damages in instruments, particle strikes have jeopardized missions, limiting func-tionality and efficiency [1], [2]. The sources of this problem are diverse and may be classified into three categories: radiation induced by cosmic rays, radiation trapped in the Earth's magnetic field, and solar radiation. The first one consists of protons and heavy ions and usually tends to present a continuous low intensity [2]. The second type of radiation is primarily formed by high-energy protons and electrons. Finally, solar radiation includes a variety of sources such as protons, ions, neutrons, gamma rays, and so on. In all cases, solutions to mitigate this problem have been difficult to implement, mainly due to the hard constraints on area and power consumption of the space applications. Among all the solutions, a first possibility is to physically shield the electronic components. This is, most of the times, unfeasible due to the heavy weight of the shielding materials. Besides, not all particles can be easily deflected with standard shielding, thus not solving the problem entirely [1]. Another solution is to fabricate circuits with a technology that can support higher doses of radiation, something known as rad-hard components. This infers a couple of problems. Rad-hard technologies require a much more expensive fabrication process, since runs are usually limited and not all the facilities can provide this technology. This usually leads to hefty cost overheads, sometimesof an order of magnitude with respect to the commercial off-the- shelf counterpart designs. Moreover, rad-hard components usually provide lower functionality than standard components, which may limit the scope of the mission [3]. Finally, another approach is to produce components protected by design. In this case, the technology is the same as in the commercial case, but the actual circuit is designed with extra redundancy in order to provide error-handling capabilities. An example of this are error-correction codes (e.g., Hamming) to protect memories [4]. Other examples to detect and correct errors are dual modular redundancy (DMR) and triple modular redundancy (TMR) schemes [5]. These techniques usually incur area and power consumption overheads, thus creating difficulties to space missions in which small satellites are used (e.g., CubeSats). Therefore, in order to reduce these overheads, other

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 9, September 2023 ||

| DOI:10.15680/IJIRSET.2023.1209075 |

approaches not based on massive redundancy are usually followed by hardware designers. One of these approaches is the so-called algorithmic-based fault tolerance (ABFT) [6]. This approach can be applied to circuits with certain algorithmic properties. If one of these properties exists, it can be used to check the correctness of each output with respect to its corresponding input, usually with a lower overhead than DMR. One of the most paradigmatic cases of ABFT is the Parseval's check in signal processing [7]. In this brief, we have focused on the discrete cosine transform (DCT) and have managed to protect it with the previously mentioned ABFT approach. The choice of the DCT as the design to protect corresponds to two reasons. The first one is its ubiquitous utilization in many signal-processing systems [8]. The second one is that it is a basic building block for image-processing hardware onboard satellites [9] and therefore very sensitive to the aforementioned effects of radiation. There are several briefs that study DCT implementations from different points of view. One approach is to try to optimize the mathematical foundation to come up with a simpler circuit (less area and power consumption) [10]. DCT area-efficient implementations have also been researched [11]. Efficiency, specifically in video coding applications, is usually a primary target. A common approach is to use hardware acceleration and parallelism [12]. The research of alternative implementation is also usual, so that the DCT can save computational cycles and resources [13]. Another approach is to use hardware–software co-design to reach a tradeoff between performance and used resources [14]. Reconfigurable architectures for DCT are also a recent target, especially to deal with variable transform size due to video resolution changes. Field-programmable gate arrays (FPGAs) are a popular alternative to support this reconfigurable functionality [15]. For top performance, DCT implementation on multicore architectures is sometimes proposed, which usually increases the power consumption. Efforts to keep consumption within reasonable limits without jeopardizing efficiency are also present in the literature [16]. There are also articles devoted to reduce the power consumption of the DCT architecture. In [17], the voltage scaling versus robustness tradeoff is studied with a soft N-modular redundancy scheme, while in [18], the voltage scaling versus delay error resiliency is analyzed. From the reliability point of view, there are publications that explore properties of fast unitary transforms, a family that comprises DCT and other similar transforms [19]. In this brief, we explore the ABFT properties of the Arai 1-D DCT implementation but with an approach different from [17], [18]. Specifically, the reliability versus resource overhead tradeoff is analyzed, since this is an important issue in space applications. Section II introduces the mathematical properties of the DCT as well as the hardware implementation studied in this brief. In Section III, the proposed protection technique for the DCT is explained. The experimental setup and the results obtained after the fault injection campaigns are presented in Sections IV and V. Finally, the shortcomings of the proposed technique are discussed in Section VI, and Section VII concludes this brief.

II. DISCRETE COSINE TRANSFORM

The amount of memory space required to store raw images can be restrictive in terms of cost and power consumption in space applications. As mentioned in Section I, the DCT is widely used in image- or video-processing systems for data compression. The main reason for this is the energy compaction property of the DCT, which is very close to the Karhunen–Loève transform [20]. For example, in a lossy image compression process (e.g., JPEG) the image is processed in blocks of 8 8 pixels to express them as a sum of cosine functions oscillating at different frequencies. In order to do that in hardware, each region of 8×8 pixels is decomposed into a cascade of 8×1 1-D DCTs that are applied on the eight rows. Then, this intermediate result is transposed, and the 1-D DCTs are applied again but this time on the eight columns. Different variants of the 1-D DCT can be applied in these steps, being the type-II DCT the most commonly used form [21]. This variant is usually referred to as "the DCT" and is defined by

$$F_k = \sum_{n=0}^{N-1} f_n \cos\left[\frac{\pi k}{N}\left(n+\frac{1}{2}\right)\right], k = 0,1,\ldots,N-1 \quad (1)$$

Where f_n are the input pixels of the original image and F are the output pixels after the DCT.

The implementation of (1) in hardware requires intensive computation, which makes it unfeasible for real-time applications. For this reason, several algorithms such as the Leoffler [22] or the Arai [23] have been proposed over the years to reduce the hardware complexity of the DCT computation. In particular, the Arai algorithm for the eight-point DCT is widely popular in hardware due to its low arithmetic complexity. This is because the Arai scheme only needs five multiplications and 29 additions, so its resource usage and power consumption is reduced. The block diagram of this architecture is illustrated in Fig. 1.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 9, September 2023 ||

| DOI:10.15680/IJIRSET.2023.1209075 |

In this figure, the subtraction of pixel values is represented by placing a negation symbol before the addition block. The values of the fixed multipliers are given by

$$m_1 = cos\frac{4\pi}{16} \quad m_3 = cos\frac{2\pi}{16} - cos\frac{6\pi}{16}$$
$$m_2 = cos\frac{6\pi}{16} \quad m_4 = cos\frac{2\pi}{16} + cos\frac{6\pi}{16} \tag{2}$$

In Section III, the proposed error-detection technique for the eight-point Arai 1-D DCT architecture presented previously isexplained in detail.

III. PROPOSED TECHNIQUE FOR THE ARAI DCT

The algorithmic property of the Arai DCT used to create the proposed ABFT technique has been inferred by dividing the block diagram shown in Fig. 1 into five sequential steps. This way, the outcomes of the adders of each step can be expressed using simpleaddition or subtraction equations. These equations are as follows:

Step 1:

$$\begin{aligned} a_{11} &= f_0 + f_7 \quad a_{15} = f_3 - f_4 \\ a_{12} &= f_1 + f_6 \quad a_{16} = f_2 - f_5 \\ a_{13} &= f_2 + f_5 \quad a_{17} = f_1 - f_6 \\ a_{14} &= f_3 + f_4 \quad a_{18} = f_0 - f_7 \end{aligned} \tag{3}$$

Step 2:

$$\begin{aligned} a_{21} &= a_{11} + a_{14} \quad a_{25} = a_{15} + a_{16} \\ a_{22} &= a_{12} + a_{13} \quad a_{26} = a_{16} + a_{17} \\ a_{23} &= a_{12} - a_{13} \quad a_{27} = a_{17} + a_{18} \\ a_{24} &= a_{11} - a_{14} \end{aligned} \tag{4}$$

Step 3:

$$\begin{aligned} a_{31} &= a_{21} + a_{22} \quad a_{31} = (a_{23} + a_{24}).m_1 \\ a_{32} &= a_{21} - a_{22} \quad a_{34} = (a_{25} - a_{27}).m_2 \end{aligned} \tag{5}$$

Step 4:

$$\begin{aligned} a_{41} &= -a_{33}.m_1 + a_{24} \\ a_{42} &= a_{33}.m_1 + a_{24} \\ a_{43} &= a_{25}.m_3 + a_{34}.m_2 \\ a_{44} &= -a_{26}.m_1 + a_{18} \\ a_{45} &= a_{34}.m_2 + a_{27}.m_4 \\ a_{46} &= a_{26}.m_1 + a_{18} \end{aligned} \tag{6}$$

Step 5:

$$\begin{aligned} a_{51} &= -a_{43} + a_{44} \quad a_{51} = -a_{45} + a_{46} \\ a_{52} &= a_{43} + a_{44} \quad a_{54} = a_{45} + a_{46} \end{aligned} \tag{7}$$

Output:

$$F_0 = a_{31} \quad F_1 = a_{32} \quad F_2 = a_{41} \quad F_3 = a_{42}$$
$$F_4 = a_{51} \quad F_5 = a_{52} \quad F_6 = a_{53} \quad F_7 = a_{54} \tag{8}$$

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 9, September 2023 ||

| DOI:10.15680/IJIRSET.2023.1209075 |

where a_{ij} represents the outcome of adder j of step i. In each step, the adders are enumerated from top to bottom. For example, adder a_{25} is the fifth adder of step 2, which is the adder just before the multiplier m_3 .

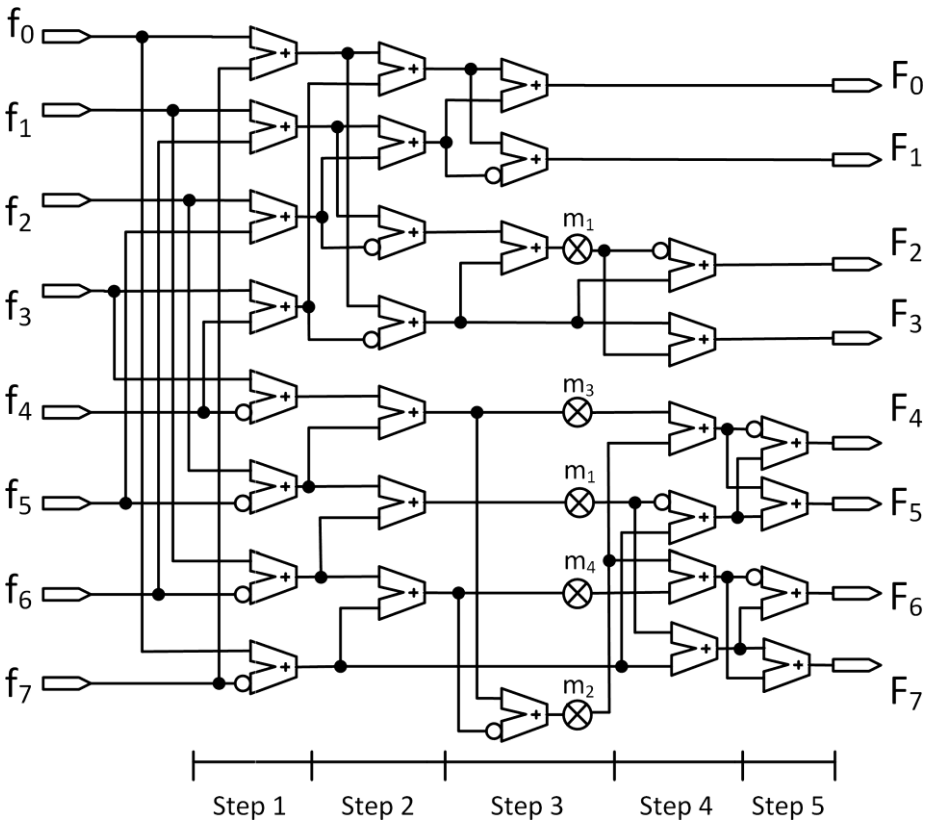

Fig 1. Block diagram of the eight-point Arai 1-D DCT architecture [23].

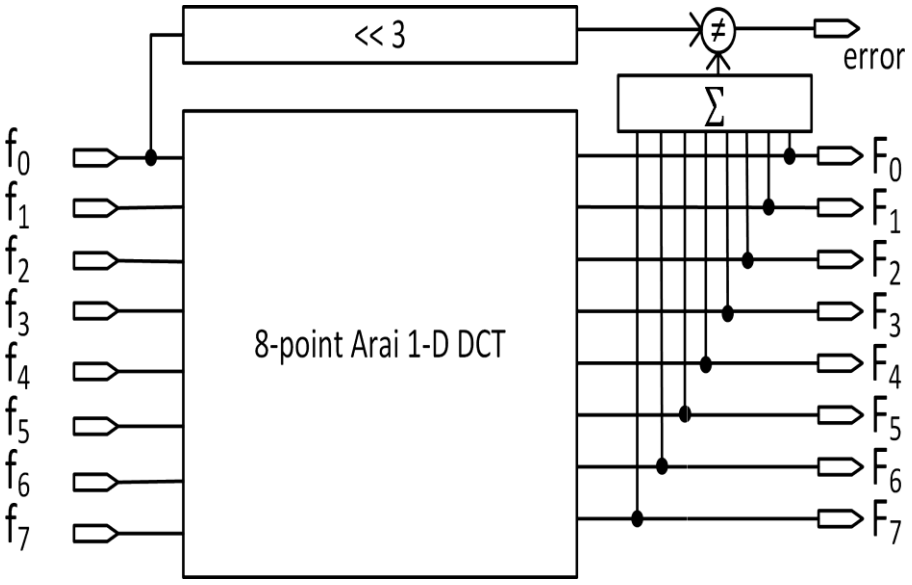

Fig 2. Proposed protection technique for the eight-point Arai 1-D DCT architecture.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 9, September 2023 ||

| DOI:10.15680/IJIRSET.2023.1209075 |

Using these equations, the Fk output values of the Arai DCT can be expressed as a function of the fn input values as follows:

$$F_0 = f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7$$
$$F_1 = f_0 - f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7$$
$$F_2 = f_0 + f_7 - f_3 - f_4 - [m_1.(f_1 + f_6 - f_2 - f_5 + f_0 + f_7 - f_3 - f_4)]$$
$$F_3 = f_0 + f_7 - f_3 - f_4 + [m_1.(f_1 + f_6 - f_2 - f_5 + f_0 + f_7 - f_3 - f_4)]$$
$$F_4 = (f_0 - f_7) - [m_1.(f_2 - f_5 + f_1 - f_6)] - [m_3.(f_3 - f_4 + f_2 - f_5)] + [m_2.(f_3 - f_4 + f_2 - f_5 - f_1 + f_6 - f_0 + f_7)]$$
$$F_5 = (f_0 - f_7) - [m_1.(f_2 - f_5 + f_1 - f_6)] + [m_3.(f_3 - f_4 + f_2 - f_5)] + [m_2.(f_3 - f_4 + f_2 - f_5 - f_1 + f_6 - f_0 + f_7)]$$
$$F_6 = m_1.(f_2 - f_5 + f_1 - f_6) + (f_0 - f_7) - \{m_4.(f_1 + f_6 - f_0 + f_7)] + [m_2.(f_3 - f_4 + f_2 - f_5 - f_1 + f_6 - f_0 + f_7)]\}$$
$$F_7 = m_1.(f_2 - f_5 + f_1 - f_6) + (f_0 - f_7) + \{m_4.(f_1 + f_6 - f_0 + f_7)] + [m_2.(f_3 - f_4 + f_2 - f_5 - f_1 + f_6 - f_0 + f_7)]\}$$

(9)

Finally, if the sum of the F_k outputs is performed, the following can be obtained:

$$\sum_{k=0}^{7} F_k = 8.f_0$$

(10)

This algorithmic property of the Arai DCT has been used to create the ABFT protection scheme illustrated in Fig. 2.

In this figure, the sum of the outputs of the DCT module is compared to the f_0 input multiplied by 8 to enable an error-detection capability in the 8-point Arai 1-D DCT. It should be mentioned that this multiplication by 8 has been performed by concatenating three zeros to the right to mimic a 3-bit left-shifting operation. In Section IV, the proposed error-detection scheme presented in Fig. 2 has been evaluated and compared to the classic DMR implementationand the unprotected version of the Arai DCT.

IV. EXPERIMENTAL SETUP

In order to assess the error-detection capabilities of the proposed protection technique, three fault injection campaigns have been conducted. In particular, the unprotected DCT (see Fig. 1), the DCT protected with a DMR scheme, and the proposed technique (see Fig. 2) have been evaluated. The first implementation provides no protection and minimal FPGA resources. The second one provides the upper bound protection (in terms of error detection) and a significant area overhead. Therefore, both of them will be compared with our proposed technique (third implementation) to evaluate the protection and area tradeoff. All designs have been implemented in a Digilent Nexys 4 double data rate (DDR) Artix-7 FPGA together with the Xilinx soft error mitigation (SEM) intellectual property (IP) Controller [24], which has been used to perform the fault injection campaigns. In these experiments, the FPGA failure model has been followed. This means that the errors have been injected in the configuration memory of the device and not in the user logic. This is a usual approach for FPGAs, since in a real scenario, most errors would happen in the aforementioned configuration memory, given that it has a much higher cross section compared to the user logic storage elements (user flip-flops, registers, and so on). It is worth mentioning that the FPGA failure model produces more complex and unpredictable effects than the regular application-specific integrated circuit (ASIC) failure model. While the latter usually produces errors in the data path that propagate (or are masked) toward the output, the former may also produce transformations in the actual structure of the design and its routing [25].

Since most of the errors that happen in a real scenario are isolated, single-error fault injection campaigns have been conducted. First, an error is injected with the SEM IP in one of the essential bits of the design. These are the configuration bits that may produce an error in the design, depending on whether the bit is critical or not [26]. The circuit is exercised and the output is compared with the previously calculated golden (error-free) output. The behavior of the circuit is logged and the previously injected error is removed. Then, a new error is injected in a different essential bit of the design, and the process is repeated. The campaign ends when all the essential bits of the design are injected and tested, performing an exhaustive fault injection campaign. Since all the essential bits are targeted, each

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 9, September 2023 ||

| DOI:10.15680/IJIRSET.2023.1209075 |

implementation has received a different number of injections. In other words, more injections have been performed in the DMR version than in the unprotected version. This is reasonable, since in a real scenario, a larger circuit would receive a higher number of events than a smaller one. In any case, all versions of the DCT (unprotected, DMR, and proposed) have been fully characterized. The results are presented and discussed in Section V.

V. EXPERIMENTAL RESULTS

In terms of reliability, different outcomes may arise when errors are injected. A first possibility is when an error injected in the configuration memory does not translate into an error at the output of the circuit. This may happen when the affected resource in the FPGA does not interact with the actual data processing, e.g., an unused lookup table (LUT) entry, a logical masking, and so on. A second possibility is when an error injected in the configuration memory does translate into an error at the output, but this error is detected by the protection logic added to the design. This is a positive outcome since the error does not propagate outside the circuit. A third possibility is when an error injected in the configuration memory translates into an error at the output and that error is undetected by the protection logic. This is the worst scenario, since the error propagates as silent data corruption. From the point of view of the reliability, the percentage of undetected errors will be used as the figure of merit to compare the fault-tolerant capabilities of the different implementations. The reliability results are depicted in Table 1.

TABLE 1
RELIABILITY COMPARISON

	Unprotected	**DMR**	**Proposed**
No Errors	15474(70.6%)	33260(69.8%)	17354(57.5%)
Detected	0(0.0%)	12842(27.0%)	9471(31.4%)
Undetected	6458(29.4%)	1541(3.2%)	3365(11.1%)
Total Injections	21932(100%)	47643(100%)	30190(100%)

First, an exhaustive fault injection campaign was conducted on the unprotected version. A total of 21 932 errors were injected in all the essential bits. Of these, 15 474 did not produce any error at the output of the design. This phenomenon is common in FPGAs since the bit flip can be performed in a bit that does not create a malfunction in the design (i.e., noncritical bit). From the rest that actually produced an error at the output (6458), all of them were undetected, which represents a 29.4% of the performed injections. It makes sense that none of the errors that propagated to the output was detected, since the unprotected version does not contain any fault-tolerant technique. The FPGA resources needed by this design were 378 LUTs and 96 flip-flops (see Table 2).

TABLE 2
FPGA OVERHEAD COMPARISON

	Resources		f_{max} **(MHz)**	**Power (mW)**
	LUTs	**Flip-flops**		
Unprotected	378	96	83	7
DMR	732	193	62	14
Proposed	476	120	50	7

Then, another injection campaign was conducted in the DMR version of the circuit. In this case, a total of 47 643 injections were performed. Of this, 33 260 did not produce any error at the output. From the rest, 12 842 errors were detected by the DMR scheme, and 1541 were undetected, which represents 3.2% of the total injections. This is of course lower than the 29.4% of the unprotected version. It is worth noticing that the DMR scheme did not detect 100% of the errors. Errors in the routing can produce a small percentage of errors that go undetected at the output. To achieve this reliability, the FPGA resources needed were 732 LUTs and 193 flip-flops, which represent an overhead of 94% and 101%, respectively, compared to the unprotected case. So, as expected, the area approximately doubles in order to reduce the percentage of undetected errors to 3.2%.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 9, September 2023 ||

| DOI:10.15680/IJIRSET.2023.1209075 |

Finally, an error injection campaign was also conducted in the proposed ABFT-based design in order to evaluate the error detection rate and the resource overhead. A total of 30 190 errors were injected in this design, of which 17 354 did not produce any error at the output. Considering the rest that did produce an error, 9471 were detected and 3365 were undetected, which represents 11.1% of the total number of injections. This means that although the proposed technique is eight percentage points higher than DMR in terms of undetected errors, it manages to reduce around two-thirds of the percentage of undetected errors produced in the unprotected version.

Although it is not the goal of this brief, it is interesting to point out that the behavior of the DMR and the proposed technique when single-event upsets (SEUs) affect the user logic (i.e., flip-flops) is similar. The reason is that the intermediate steps are not registered. Consequently, if only the last stage is registered, all the bit flips in the flip-flops would be detected if the protection technique is applied to the registered outputs. However, an error in the registers after comparison would not be detected in either case.

Next, the FPGA overheads of the different techniques have been analyzed (see Table 2). Our protection technique only needs 476 LUTs and 120 flip-flops, which represent an overhead of 26% and 25%, respectively, compared to the unprotected design. This infers important savings compared to the resources consumed by the DMR scheme (which doubles the area of the unprotected design). Besides, the obtained reliability is still in a reasonable range, much better than the one offered by the unprotected version. The same applies to power consumption. In this case, the proposed technique has a consumption overhead in the same range as the unprotected design, which represents half of the overhead incurred by DMR. This is very important in the case of small space missions (e.g., CubeSats), since the power budget is typically the worst constraint in these types of scenarios. Finally, regarding the frequency, it can be seen that the proposed technique has a delay that is around 10% worse than DMR. If this is a hard constraint in the application, a possible solution would be to pipeline the design in order to increase the throughput.

VI. SHORTCOMINGS OF THE PROPOSED TECHNIQUE

As presented in Section V, the proposed technique has 11.1% of undetected errors, whereas DMR only has 3.2%. The reason is that the ABFT strategy used is not valid in all cases as the equality still holds when the error affects some parts of the circuit. This is due to compensation effects between branches. Let us consider, for example, that there is an error in a43 from (6) in step 4. This error propagates to a51 and a52 in step 5 correspond to F4 and F5 in the output. When the outputs are summed up in the presented algorithm, a43 is canceled as $F4 + F5 = a51 + a52 = -a43 + a44 + a43 + a44 = 2 \cdot a44$. The same occurs with a45, a33, and other signals.

This approach has given a solution with a slightly worse reliability but with significant savings in the area overhead with respect to DMR. It is possible to explore additional techniques to provide a higher protection (e.g., by using a selective DMR), but the overhead would also increase. The bottom line is that there are no solutions better than others, since those that excel in reliability would offer worse overheads and vice versa. The important aspect is to have a variety of protection techniques, with a wide range of the reliability versus overhead tradeoffs. Then, it would be up to the designer to choose the one that fits better the application requirements, meeting the expected reliability constraint while minimizing the incurred overhead.

VII. CONCLUSION

In this brief, a new protection technique for the DCT implemented in an FPGA has been presented. It is based on an algorithmic property, which states that the addition of the DCT outputs is equal to eight times the least significant input. Using this, a checker has been designed to reduce the number of undetected errors of the unprotected DCT. Due to the circuit structure, not all the errors can be detected, since some wrong outputs still meet the previous algorithmic property. However, the percentage of undetected errors is close to that of DMR and much better than the results offered by the unprotected DCT. Besides, the FPGA resource overhead that our technique incurs is very limited compared to DMR (which doubles this overhead). In summary, the reliability versus area ratio in our technique outperforms the DMR solution, since the area of the former is quite limited. The reduction in area and power consumption is especially important in small-satellite applications. The loss of reliability in respect to DMR can be acceptable in applications that can allow a percentage of wrong outputs without compromising the expected functionality (e.g., vision applications). In future studies, the protection of other DCT implementations, such as the Leoffler algorithm, will be explored to keep on demonstrating the convenience of the ABFT approach

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 9, September 2023 ||

| DOI:10.15680/IJIRSET.2023.1209075 |

REFERENCES

[1] J. W. Howard and D. M. Hardage, "*Spacecraft environments interactions: Space radiation and its effects on electronic systems*," NASA, Washington, DC, USA, NASA Tech. Rep. TP-1999-209373, 1999.
[2] A. Holmes-Siedle and L. Adams, Handbook of Radiation Effects. New York, NY, USA: Oxford Univ. Press, 2002, pp. 80–85.
[3] I. Mcloughlin, V. Gupta, G. Sandhu, S. Lim, and T. Bretschneider, "*Fault tolerance through redundant COTS components for satellite processing applications*," in Proc. 4th Int. Conf. Inf., Commun. Signal Process. Joint 4th Pacific Rim Conf. Multimedia, Jul. 2004, pp. 296–299.
[4] C. L. Chen and M. Y. Hsiao, "*Error-correcting codes for semiconductor memory applications: A state-of-the-art review*," IBM J. Res. Develop., vol. 28, no. 2, pp. 124–134, 1984.
[5] Techniques for Radiation Effects Mitigation in ASICs and FPGAs Handbook, Standard ECSS-Q-HB-60-02A, ECSS Secretariat ESA-ESTEC Requirements & Standards Division, Noordwijk, The Netherlands, 2016.
[6] K.-H. Huang and J. A. Abraham, "*Algorithm-based fault tolerance for matrix operations*," IEEE Trans. Comput., vol. C-33, no. 6, pp. 518–528, Jun. 1984.
[7] S.-J. Wang and N. Jha, "*Algorithm-based fault tolerance for FFT networks,*" IEEE Trans. Comput., vol. 43, no. 7, pp. 849–854, Jul. 1994.
[8] S. Bouguezel, M. Ahmad, and M. Swamy, "*A new class of reciprocal–orthogonal parametric transforms*," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 56, no. 4, pp. 795–805, Apr. 2009.
[9] S. Vijay and D. Anchit, "*Low power implementation of DCT for on-board satellite image processing systems*," in Proc. 52nd IEEE Int. Midwest Symp. Circuits Syst., Cancun, Mexico, Aug. 2009, pp. 774–777.
[10] D. F. Chiper and L. T. Cotorobai, "*A unified VLSI architecture for 1D IDCT and IDST based on pseudo-band correlations*," in Proc. 10th Int. Conf. Electron., Comput. Artif. Intell. (ECAI), Iasi, Romania, Jun. 2018, pp. 1–5.
[11] M. Masera, G. Masera, and M. Martina, "*An area-efficient variable-size fixed-point DCT architecture for HEVC encoding*," IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 1, pp. 232–242, Jan. 2020.
[12] S. Nouri, R. Ghaznavi-Youvalari, and J. Nurmi, "*Design and implementation of multi-purpose DCT/DST-specific accelerator on heterogeneous multicore architecture*," in Proc. IEEE Nordic Circuits Syst. Conf. (NORCAS) Int. Symp. System-Chip (SoC), Tallinn, Estonia, Oct. 2018, pp. 1–10.
[13] P. Dahiya and P. Jain, "*Realization of recursive algorithm for one-dimensional discrete cosine transform and its inverse*," in Advances in System Optimization and Control (Lecture Notes in Electrical Engineering), vol. 509, S. Singh, F. Wen, M. Jain, Eds. Singapore: Springer, 2019.
[14] B. Chan Jia Ching, A. A.-H. Ab Rahman, and N. Ahmad, "*Implementation of an 8×8 discrete cosine transform on programmable system-onchip*," J. Phys., Conf. Ser., vol. 1049, Jul. 2018, Art. no. 012084.
[15] M. Zheng, J. Zheng, Z. Chen, L. Wu, X. Yang, and N. Ling, "*A reconfigurable architecture for discrete cosine transform in video coding*," IEEE Trans. Circuits Syst. Video Technol., to be published.
[16] R. Radhika and N. Partheeban, "*Improving power efficiency in the implementation of discrete cosine transform in multicore architecture*," J. Comput. Theor. Nanosci., vol. 15, no. 8, pp. 2597–2603, Aug. 2018.
[17] E. P. Kim and N. R. Shanbhag, "*Soft N-modular redundancy*," IEEE Trans. Comput., vol. 61, no. 3, pp. 323–336, Mar. 2012.
[18] G. Karakonstantis, N. Banerjee, and K. Roy, "*Process-variation resilient and voltage-scalable DCT architecture for robust low-power computing*," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 10, pp. 1461–1470, Oct. 2010.
[19] S.-H. Hu and J. A. Abraham, "*Quality aware error detection in 2-D separable linear transformation*," in Proc. IEEE 25th Asian Test Symp. (ATS), Hiroshima, Japan, Nov. 2016, pp. 257–262.
[20] A. Edirisuriya, A. Madanayake, V. S. Dimitrov, R. J. Cintra, and J. Adikari, "*VLSI architecture for 8-point AI-based Arai DCT having low area-time complexity and power at improved accuracy*," J. Low Power Electron. Appl., vol. 2, no. 2, pp. 127–142, Mar. 2012.
[21] N. Ahmed, T. Natarajan, and K. R. Rao, "*Discrete cosine transform*," IEEE Trans. Comput., vol. C-100, no. 1, pp. 90–93, Jan. 1974.
[22] D. F. G. Coelho, S. Nimmalapalli, V. S. Dimitrov, A. Madanayake, R. J. Cintra, and A. Tisserand, "*Computation of 2D 8×8 DCT based on the Loeffler factorization using algebraic integer encoding*," IEEE Trans. Comput., vol. 67, no. 12, pp. 1692–1702, Dec. 2018.
[23] Y. Arai, T. Agui, and M. Nakajima, "*A fast DCT-SQ scheme for images*," IEICE Trans., vol. E71, no. 11, pp. 1095–1097, 1988.
[24] Soft Error Mitigation Controller Logicore IP Product Guide (PG036), Xilinx, San Jose, CA, USA, Sep. 2015.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 9, September 2023 ||

| DOI:10.15680/IJIRSET.2023.1209075 |

[25] B. Du and L. Sterpone, "*Online monitoring soft errors in reconfigurable FPGA during radiation test*," in Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC), Turin, Italy, May 2017, pp. 1–5.

[26] C. D. Patterson, P. Sundararajan, B. J. Blodget, and S. P. McMillan, "*Method and system for identifying essential configuration bits*," U.S. Patent 7 406 673, Jul. 29, 2008.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.423

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details