electronics

MDPI

Article

# FAC: A Fault-Tolerant Design Approach Based on Approximate Computing

**Padmanabhan Balasubramanian * and Douglas L. Maskell**

School of Computer Science and Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore; asdouglas@ntu.edu.sg

* Correspondence: balasubramanian@ntu.edu.sg; Tel.: +65-6790-4745

**Abstract:** This article introduces a new fault-tolerant design approach based on approximate computing, called FAC, for designing redundant circuits and systems. Traditionally, triple modular redundancy (TMR) has been used to ensure complete tolerance to any single fault or a faulty processing unit, where the processing unit may be a circuit or a system. However, TMR incurs more than 200% overhead in terms of area and power compared to a single processing unit. Alternative redundancy approaches have been proposed in the literature to mitigate these overheads associated with TMR, but they provide only partial or moderate fault tolerance. Among the alternatives, majority voting-based reduced precision redundancy (MVRPR) may be useful for error-resilient applications such as digital signal processing. While MVRPR guarantees only moderate fault tolerance, the proposed FAC is well-suited for error-resilient applications and ensures 100% tolerance to any single fault or a faulty processing unit, like TMR. In this work, we evaluate the performance of TMR, MVRPR, and FAC for a digital image processing application. The image processing results obtained demonstrate the effectiveness of FAC. Moreover, when the processing unit is implemented using a 28-nm CMOS technology, FAC achieves significant improvements over TMR, including a 15.3% reduction in delay, a 19.5% reduction in area, and a 24.7% reduction in power. Compared to MVRPR, FAC exhibits notable enhancements, with an 18% reduction in delay, a 5.4% reduction in area, and an 11.2% reduction in power. When considering the power-delay product, which reflects energy efficiency, FAC demonstrates a 36.2% reduction compared to TMR and a 27.2% reduction compared to MVRPR. When considering the power-delay-area product, which represents design efficiency, FAC achieves a 48.7% reduction compared to TMR and a 31.1% reduction compared to MVRPR.

**Keywords:** fault tolerance; triple modular redundancy; approximate computing; arithmetic circuits; digital logic design; low power; high-speed; CMOS

check for updates

**Citation:** Balasubramanian, P.; Maskell, D.L. FAC: A Fault-Tolerant Design Approach Based on Approximate Computing. *Electronics* **2023**, *12*, 3819. https://doi.org/10.3390/electronics12183819

Academic Editor: Marcin Witczak

Received: 24 July 2023
Revised: 1 September 2023
Accepted: 7 September 2023
Published: 9 September 2023

## 1. Introduction

Due to the reduction in the size of transistors, processing units comprising them like electronic circuits and systems are more susceptible to faults or failures during regular operation [1,2] or aging [3]. This susceptibility is amplified when processing units are subjected to challenging conditions, such as space, where the occurrence of high-energy radiation [4] is highly likely. Radiation can have various sources and causes when it comes to harsh environments. Various sources of radiation can lead to different types of radiation effects that impact the performance, reliability, and functionality of electronic components. Some common causes of radiation in harsh environments include, but are not limited to, the following:

- Ionizing radiation: Consists of particles such as alpha particles and beta particles or electromagnetic waves such as gamma rays and X-rays with enough energy to ionize atoms and molecules, creating charged particles and potentially damaging electronic components.

- Protons and neutrons: These can be found in space environments, especially near the Earth's radiation belts and in deep space, which can create displacement damage in materials and induce charge buildup in sensitive regions of electronic components.
- Solar flares: These are sudden bursts of energy and particles released by the Sun during magnetic disturbances. They can lead to enhanced radiation levels in space environments, impacting electronic systems in satellites and spacecraft.
- Cosmic rays: These are high-energy particles originating from outside the solar system which include protons, alpha particles, and heavier ions that can interact with the Earth's atmosphere and contribute to radiation in high-altitude and space environments.
- Nuclear environments: In nuclear facilities and during nuclear testing, electronic circuits or systems can be exposed to intense radiation fields from nuclear reactions, leading to various radiation effects.

The types of radiation effects on electronic circuits and systems include single-event effects, total ionizing dose effects, displacement damage, and more. These effects can cause temporary or permanent changes in electronic behavior such as single-event upsets, latch-ups, gate ruptures, and increased leakage currents. Therefore, designing electronic components and systems to withstand these radiation effects involves using radiation-hardened materials, implementing shielding, using redundancy, employing error correction codes, etc., to enable robustness in the presence of radiation.

Studying and testing the effects of radiation on electronic components can be done through simulations and real test experiments. The simulation methods include:

- Monte Carlo simulations: These use statistical techniques to model the behavior of radiation particles as they interact with materials. This helps in understanding how radiation affects electronic components and can predict potential failures.
- Device-level simulations: Specific electronic components, such as transistors and diodes, can be modeled using device-level simulation tools like TCAD (Technology Computer-Aided Design). These simulations help analyze how radiation-induced charge buildup affects the operation of these devices.
- Circuit-level simulations: Tools like SPICE are used to simulate the behavior of entire electronic circuits under radiation conditions. This allows for studying the impact of radiation on circuit performance, timing, and functionality.
- System-level simulations: For complex systems, digital twin simulations can be created to replicate the entire system's behavior under radiation exposure. This involves simulating the interactions between various components and subsystems to predict system-level effects.

Real test experiments include the following:

- Ionizing radiation sources: Radiation sources such as X-rays, gamma rays, and particle accelerators are used to subject electronic components and systems to controlled levels of radiation. These sources replicate the types of radiation encountered in harsh physical environments.
- Radiation chambers: Specialized chambers can be used to expose electronic components and systems to controlled radiation levels. These chambers allow researchers to precisely control the radiation dose and study the effects on components and systems.
- Field testing: In some cases, electronic systems are deployed in radiation-prone environments, such as satellites or spacecraft, and their behavior is monitored in real time. This provides valuable data on the actual impact of radiation on the system's performance.
- Post-irradiation analysis: After exposure to radiation, components and systems are analyzed to identify changes in performance, behavior, and failure modes. Techniques like scanning electron microscopy and other material analysis methods are used to identify radiation-induced damage.
- Single-event effects (SEE) testing: SEEs are rapid, transient effects caused by a single radiation particle striking a sensitive node in an electronic circuit. Testing involves

exposing components to radiation and observing how these particles affect circuit behavior, potentially leading to errors or failures.

- Radiation-hardened components: Some electronic components are designed to be more resilient to radiation, and their effectiveness is tested through exposure to radiation sources to ensure they meet the required performance levels.

Given the above, processing units such as circuits or systems utilized in safety-critical applications require protection against radiation. This work focuses on redundancy as a fault tolerance design strategy to address faults that may arise in processing units within a specified limit due to the impact of radiation. In the rest of this article, Section 2 surveys relevant literature on accurate and approximate redundancy techniques. Section 3 describes the proposed redundancy approach utilizing approximate computing called FAC. An abridged version of this work was presented in IEEE TENSYMP 2023 [5]. This article is an extended version that contains 2× extra image processing results. In Section 4, we assess the performance of TMR, MVRPR, and FAC for a digital image processing application. Section 5 presents the design metrics of single, TMR, MVRPR, and FAC implementations of a sample processing unit. Compared to [5], we present two extra figures of merit for evaluating the redundant designs in this article. In Section 6, we draw some conclusions based on the findings and insights discussed in the preceding sections.

## 2. Survey of Related Literature

N-Modular Redundancy (NMR) is a well-known approach that uses N identical processing units, and the outputs of the N processing units are combined using majority voters to generate the final output. In NMR, for a set of N identical processing units (where N is an odd number, typically N = 3 or more), it is necessary for the majority, specifically (N + 1)/2 processing units, to function correctly to ensure the proper functioning of the NMR scheme, assuming the majority voter itself operates correctly. However, the majority voter may be hardened like a processing unit by duplicating it to ensure a robust operation. Within the NMR scheme, faults of $(N - 1)/2$ processing units can be tolerated without affecting the final output.

Triple Modular Redundancy (TMR) represents the fundamental version of NMR and enjoys widespread popularity and use. In a practical study presented in [6], various Virtex FPGA devices were exposed to radiation from protons and heavy ions. The study revealed that single-bit upsets accounted for 96% to 99% of all upsets, with multiple-bit upsets making up the remaining percentage. Considering the dominant prevalence of single-bit upsets in high-energy radiation environments like space, TMR offers an effective solution. TMR involves the use of three identical processing units, and their outputs are combined through majority voting to produce the primary output. Consequently, TMR can successfully tolerate any single fault or any faulty processing unit. However, implementing TMR requires two additional identical processing units and a majority voting logic compared to a single processing unit. As a result, a TMR implementation incurs additional overheads in terms of area and power, exceeding 200% compared to a single implementation. Moreover, a TMR implementation may experience a slightly increased delay compared to a single implementation due to the presence of a majority voter in the critical data path.

To mitigate the area and power overheads associated with TMR, researchers have proposed compromise approaches [7–10] that aim to minimize design metrics such as area, power, and delay while compromising on fault tolerance to some extent. One such approach, known as Selective insertion of TMR (STMR), was introduced in [7]. STMR suggests applying TMR only to the critical components of a processing unit while leaving the less critical parts as a single implementation. By adopting STMR instead of the conventional full TMR, it becomes possible to reduce both the area and power requirements of the redundant implementation. However, there are a couple of challenges associated with STMR. Firstly, determining which parts of a processing unit are critical and which are not may not be straightforward for all practical applications. Moreover, this differentiation may not remain valid throughout the entire lifespan of the processing unit. Secondly, if the unprotected, less critical parts of a processing unit are affected, there is no guarantee that the outputs of the processing unit will remain unaffected or intact.

In [8], the concept of Approximate TMR (ATMR) was introduced. ATMR involves using one accurate processing unit and two different approximate processing units with reduced logic. The outputs of the accurate and approximate processing units are majority-voted to generate the primary outputs. Unlike traditional (full) TMR which utilizes three accurate processing units, ATMR offers reduced area and power dissipation due to its combination of one accurate and two approximate units. However, the implementation of ATMR comes with certain challenges. Firstly, if either the accurate processing unit or one of the approximate processing units produces a faulty output, the corresponding output of ATMR could become erroneous. Secondly, if the accurate processing unit itself becomes faulty, and its outputs do not match the outputs of the approximate units, ATMR could experience failure. These scenarios highlight the fact that ATMR is not fully resilient to a single fault or a faulty processing unit, which goes against the fundamental property of TMR. This is because the primary strength of TMR lies in its ability to reliably mask any single fault or a faulty processing unit.

Furthermore, an alternative approach called Fully Approximate TMR (FATMR) was introduced in [8,9] to address the design overheads associated with traditional TMR. FATMR employs three distinct approximate versions of the original accurate processing unit, and their outputs are subjected to majority voting using accurate majority voters. In FATMR, the outputs of any two approximate processing units align, meaning that if one of the approximate units produces a faulty output, the corresponding output of FATMR would be erroneous. Moreover, if any of the approximate processing units were to become faulty, it could jeopardize the FATMR implementation, leading to inaccurate outputs. Consequently, FATMR tends to exhibit a higher degree of unreliability compared to ATMR. Both ATMR and FATMR are unsuitable for safety-critical applications due to the inherent uncertainty in their output, even in the presence of a single fault or a faulty processing unit. Therefore, ATMR and FATMR are excluded from further discussion in this article. Additionally, to the best of our knowledge, there is no practical demonstration of the usefulness of ATMR and FATMR in real-world applications.

In [10], a novel technique called Majority Voting-based Reduced Precision Redundancy (MVRPR) was introduced, specifically targeting naturally error-resilient applications. One such application is digital signal processing, which encompasses tasks like digital image, video, and audio processing. These applications inherently possess a degree of error tolerance since minor distortions in images or videos or subtle background noise in audio might not be discernible to the human eye or ear due to the limitations of human perception. Considering that digital image, video, and audio processing are utilized in space systems, a reduced-precision approach for these tasks can be deemed acceptable, provided the resulting quality remains adequate. By reducing the precision of the digital system, it becomes possible to lower its design metrics and enhance its energy efficiency, making MVRPR relevant and advantageous in this context.

In [10], the authors focused on describing the design of an MVRPR adder. The key feature of an MVRPR adder is its division into two equal-sized parts: a significant part and a less significant part. The categorization of these parts as significant and less significant is determined by the importance assigned to the sum bits generated by each respective part. The sum bits from both parts are concatenated to obtain the final sum output. In the MVRPR implementation, the significant part of the adder benefits from TMR protection. This means that the significant part is triplicated, and its corresponding outputs are subjected to majority voting, ensuring high reliability. On the other hand, the less significant part remains as a single, unprotected implementation. As a result, any potential single-bit upset(s) that affect the less significant part could affect the primary output. Because of the absence of protection for the less significant part in MVRPR, its fault tolerance capability is only moderate when compared to TMR as TMR offers a 100% fault tolerance by triplicating all parts of the processing unit.

## 3. Proposed Redundancy Approach—FAC

In this article, we introduce a novel Fault-tolerant design approach based on Approximate Computing, abbreviated as FAC. Before delving into the details of FAC, we provide a brief overview of approximate computing. Approximate computing presents a promising alternative to traditional accurate computing, especially for applications that naturally tolerate errors. By accepting a certain level of compromise on the computation accuracy, approximate computing offers advantages such as reduced area, lower power dissipation, higher processing speed, and improved energy efficiency [11,12]. The benefits of approximate computing have been successfully demonstrated in various practical applications, particularly those that inherently exhibit error resilience, such as multimedia tasks encompassing digital signal processing, computer graphics, computer vision, neuromorphic computing, and the implementation of hardware for AI, machine learning, and neural networks [13]. Consequently, leveraging the potential of approximate computing becomes an appealing prospect for designing fault-tolerant processing units, especially in resource-constrained environments like space applications, where area efficiency, low power dissipation, high processing speed, and energy efficiency are critical factors.

Earlier works [7–9] have suggested approximate implementations of redundancy; however, as discussed in Section 2, these approaches suffer from significant drawbacks and are unlikely to be suitable for practical applications. In contrast, the proposed FAC exhibits good potential for utilization in safety-critical applications, such as digital imaging or video systems employed in space missions. FAC is designed generically and can be applied to address any form of NMR. Nonetheless, for this article, we focus our discussion on a 3-tuple version of FAC to facilitate a direct comparison with TMR and MVRPR. To elucidate the distinctions between TMR, MVRPR, and FAC, we provide an illustrative overview of their general architectures in Figure 1a–c, respectively.

In TMR, three identical processing units are employed, denoted by Figure 1a, and the processing units are all accurate. The majority voters utilized in TMR are also accurate. The outputs of each processing unit are represented by A and B, with output A assumed to hold more significance than output B. This assumption is reasonable, particularly for arithmetic circuits, where the output bits vary in significance from most to least significant. The outputs A and B from the processing units are subjected to voting using (accurate) majority voters 1 and 2, respectively. A (3-input) majority voter synthesizes the Boolean function F = XY + YZ + XZ, where F represents the output, and X, Y, and Z are the inputs. Various majority voter designs relevant to TMR can be found in [14], with the majority voter typically assumed to be perfect. However, if it cannot be assumed to be perfect, redundancy can be applied to the majority voter, like the processing units. The primary outputs of the TMR implementation, denoted as V1 and V2, are the outputs of majority voters 1 and 2, respectively. By triplicating the processing units and using the majority voters, a TMR implementation effectively conceals any single fault or a faulty processing unit while assuming the majority voters are perfect.

Referring to Figure 1b, as described in MVRPR [10], the processing unit undergoes triplication for its significant part while the less significant portion remains as a single implementation. As mentioned earlier, A is considered more significant than B. Consequently, in the MVRPR implementation, A is assumed to be generated by the triplicated significant parts of the processing unit, while B is output by the non-triplicated, less significant part. Thus, only the A outputs of processing units are subjected to voting, using majority voter 1, with its output labeled as V1. In the context of MVRPR, both the significant and less significant parts may remain interconnected. When the adder functions as a processing unit, the triplicated significant parts and the less significant parts are connected via an intermediate carry signal (Q). This carry signal is output by the less significant part and serves as a carry input to the triplicated significant parts. The output B of MVRPR shown in Figure 1b is logically equivalent to the output V2 of TMR shown in Figure 1a. Therefore, V1 and B represent the primary outputs of the MVRPR implementation. However, it is essential to note that in MVRPR, the less significant part of the processing unit is not protected. Hence, if this part is affected (i.e., if B is affected), it may impact the output of the MVRPR implementation. As a result, MVRPR offers only moderate fault tolerance to a single fault or a faulty processing unit, unlike TMR.

Referring to Figure 1c, our proposed redundancy approach (FAC) involves partitioning a processing unit into two parts based on the significance of their outputs to the primary output, like MVRPR. However, unlike the approach presented in [10], where a processing unit is partitioned into two equal parts, FAC involves dividing the processing unit into two tailored parts based on the application's specific requirements. This implies that the two parts could be of equal or unequal size, depending upon the application. Unlike MVRPR, FAC triplicates both the significant and less significant parts. Further, in FAC, the less significant part of the processing unit is approximated instead of being retained accurately. The constituents of processing units 1, 2, and 3 are depicted within red, blue, and brown boxes in dashed lines in Figure 1c. Nevertheless, all three processing units are identical. It should be noted that each processing unit's less significant (approximate) part in FAC is identical and may or may not be connected to its corresponding significant part. The decision regarding this connection depends on the manner of logical approximation applied to the less significant part of the processing unit. The connections between the less significant and significant parts of each processing unit in FAC are represented by dotted black lines in Figure 1c, with the intermediate output being denoted as T. FAC possesses fault tolerance like TMR, as it can mask any single fault in the significant or less significant part of any processing unit or a faulty processing unit. However, because the triplicated less significant parts are approximated, the practical applicability of FAC hinges on two factors: (i) The manner of logical approximation applied to the less significant parts of the processing unit, (ii) The extent of approximation in these less significant parts.

In Figure 1c, the outputs A from processing units 1, 2, and 3 are subjected to a majority voting process using majority voter 1, resulting in output V1. This voting mechanism is also employed in TMR and MVRPR. As mentioned earlier, the triplicated less significant portions of processing units 1, 2, and 3 in FAC are identical but approximate. Consequently, the output B* from processing units 1, 2, and 3 may or may not be equivalent to the output B from the accurate processing unit, depending on the inputs provided. For instance, if we consider that a 2-input EXOR gate is accurately used as the less significant part of a processing unit in MVRPR, with binary inputs X and Y, the EXOR gate will yield 1 if $X \neq Y$ and 0 if $X = Y$. If, due to approximation, the EXOR gate is replaced with a 2-input OR gate in FAC, the OR gate will output 1 when $X = Y = 1$, and $X \neq Y$, and it will output 0 only when $X = Y = 0$. Thus, for the conditions where $X = Y = 0$ and $X \neq Y$, both EXOR and OR gates will produce the same output; however, when $X = Y = 1$, the outputs of the two gates will differ. Consequently, B* may or may not be equal to B based on the inputs provided. The outputs B* from the less significant parts of processing units 1, 2, and 3 are subjected to voting using majority voter 2, resulting in the output V2*. It should be noted that V2* may or may not be equal to V2, and V1 and V2* represent the primary outputs of an FAC implementation.

Arithmetic circuits, including adders, multipliers, dividers, and data paths with functions like the discrete Cosine transform, finite/infinite impulse response filter, and sum of absolute difference, exhibit varying degrees of significance in their output bits. This characteristic allows us to partition these processing units into significant and less significant parts, presenting an opportunity for implementing them using FAC. Depending on the target application, the less significant part of a processing unit can be approximated to a suitable degree. Similarly, logic functions can be redundantly implemented according to FAC, and again, the level of logic approximation for the less significant part should be determined based on the practical application [15].

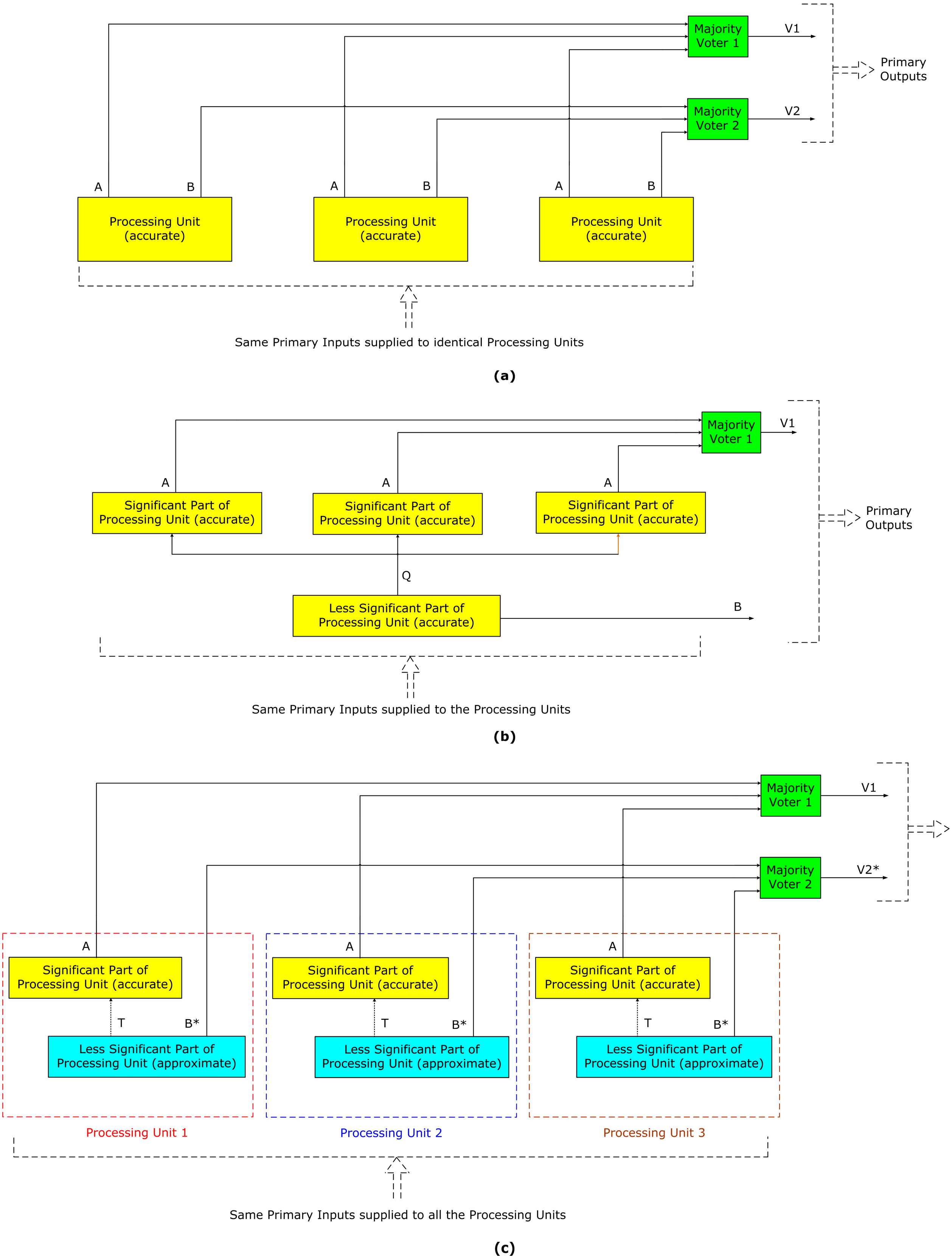

Majority Voter 1
V1
Majority Voter 2
V2
Primary Outputs
A
B
Processing Unit (accurate)
Same Primary Inputs supplied to identical Processing Units
(a)
Majority Voter 1
V1
Primary Outputs
A
Significant Part of Processing Unit (accurate)
Q
Less Significant Part of Processing Unit (accurate)
B
Same Primary Inputs supplied to the Processing Units
(b)
Majority Voter 1
V1
Majority Voter 2
V2*
Primary Outputs
A
Significant Part of Processing Unit (accurate)
T
B*
Less Significant Part of Processing Unit (approximate)
Processing Unit 1
Processing Unit 2
Processing Unit 3
Same Primary Inputs supplied to all the Processing Units
(c)

**Figure 1.** General block-level illustrations of: (**a**) TMR architecture, (**b**) MVRPR architecture, and (**c**) Proposed FAC architecture. B* of (**c**) may or may not be equal to B of (**a**,**b**), depending upon the inputs supplied.

To summarize the three redundant architectures representatively illustrated in Figure 1, TMR consists of three accurate and identical processing units along with two accurate majority voters. In contrast, MVRPR involves triplicating the significant part of the processing unit while retaining the less significant part as a single unit. MVRPR does not use a majority voter for the less significant part. Consequently, MVRPR can achieve reduced area and power dissipation compared to TMR. However, both MVRPR and TMR exhibit similar delays when the less significant part of MVRPR is connected to the significant parts of the processing units. On the other hand, FAC takes advantage of approximating the less significant part of the processing unit, leading to reduced area and power dissipation compared to TMR. If the combined approximated logic of the triplicated less significant parts in FAC is smaller than the accurate less significant part in MVRPR, FAC can achieve area and power reduction compared to MVRPR, which has been observed in the application studied in this work and will be discussed in the next section. Additionally, in a FAC implementation, if the less significant and significant parts of processing units can be disconnected, FAC could reduce the delay compared to both TMR and MVRPR. Thus, FAC offers the advantage of providing 100% protection against single faults or a faulty processing unit, like TMR, while also achieving improved optimization in design metrics for implementation by incorporating acceptable approximations within the processing units. This allows FAC to possibly achieve the best of both worlds in terms of fault tolerance and design efficiency, particularly for error-tolerant applications.

## 4. Digital Image Processing Application and Results

To compare the performance of TMR, MVRPR, and the proposed FAC, a digital image processing scenario involving fast Fourier transform (FFT) and inverse fast Fourier transform (IFFT), as in [16], was considered. A selection of 8-bit grayscale images with a spatial resolution of 512 × 512 from [17] was randomly chosen for evaluation. Each image was converted into a matrix format and subjected to FFT computation, followed by image reconstruction using IFFT. The FFT and IFFT computations were carried out in integer precision with scaling to ensure no data loss or overflow occurred during the process. During the FFT and IFFT computations, multiplication was performed with precision, while addition was performed precisely using an accurate adder and imprecisely using an inaccurate adder, separately. The architecture of the inaccurate adder [18] used in the FAC approach is depicted in Figure 2, featuring a violet section representing the accurate part and a pink section representing the approximate part. The accurate and approximate adder parts are marked in Figure 2 for easy comparison with Figure 1c. The accurate part is considered significant, while the approximate part is regarded as less significant. In Figure 2, the adder size is N bits, the size of the approximate adder part is L bits, and the size of the accurate adder part is (N–L) bits. Thus, the accurate part adds (N–L) input bits along with a carry input from the approximate part and produces (N–L + 1) sum bits. In the approximate part, sum bits $SUM_{L-1}$ up to $SUM_{L-4}$ have reduced logic while the remainder of the sum bits $SUM_{L-5}$ up to $SUM_0$ are assigned a constant binary 1. For synthesis, $SUM_{L-5}$ up to $SUM_0$ are individually connected to tie-to-high standard library cells. The value of L is determined based on the maximum error tolerable for a given application. The adder inputs are represented by $A_{L-1}$ up to $A_0$ and $B_{L-1}$ up to $B_0$, while the adder output is denoted by $SUM_N$ up to $SUM_0$. Subscripts (N–1) and 0 indicate the most significant bit and the least significant bit for the adder inputs, respectively. Similarly, subscripts N and 0 signify the most significant bit and the least significant bit for the adder's sum outputs, respectively.

TMR and MVRPR utilize the accurate adder, while FAC employs an inaccurate adder, as illustrated in Figure 2. To determine the maximum allowable approximation for the inaccurate adder, ensuring acceptable image quality after processing, extensive experimentation with numerous images and error analysis was conducted. The fundamental principle in approximate computing involves integrating the highest degree of approximation that maintains an acceptable level of output quality. Typically, this level is determined through trial and error specific to a given application. From a hardware standpoint, employing less approximation than an application can accommodate (called 'under-approximation') would yield satisfactory output quality but curtail the potential savings in design metrics achievable when compared to using precise hardware. Conversely, adopting a level of approximation that exceeds what an application can tolerate (called 'over-approximation') would result in subpar and unsatisfactory output quality (despite yielding exaggerated savings in design metrics), which is undesirable. Hence, the ideal approach is to identify the 'optimum approximation' that an application can embed while ensuring a practically acceptable output quality. This strategy allows for the maximization of design metric savings compared to using precise hardware, without compromising the output quality beyond the acceptable threshold for the given application. In [19], it was illustrated how the quality of processed digital images varies for the three example scenarios of under-approximation, optimum approximation, and over-approximation. For a 32-bit addition, the maximum acceptable approximation while ensuring an acceptable output quality (here, image quality) was found to be 10 sum bits for the approximate adder part (L = 10) and 22 sum bits for the accurate adder part (N–L = 22), based on trial and error.

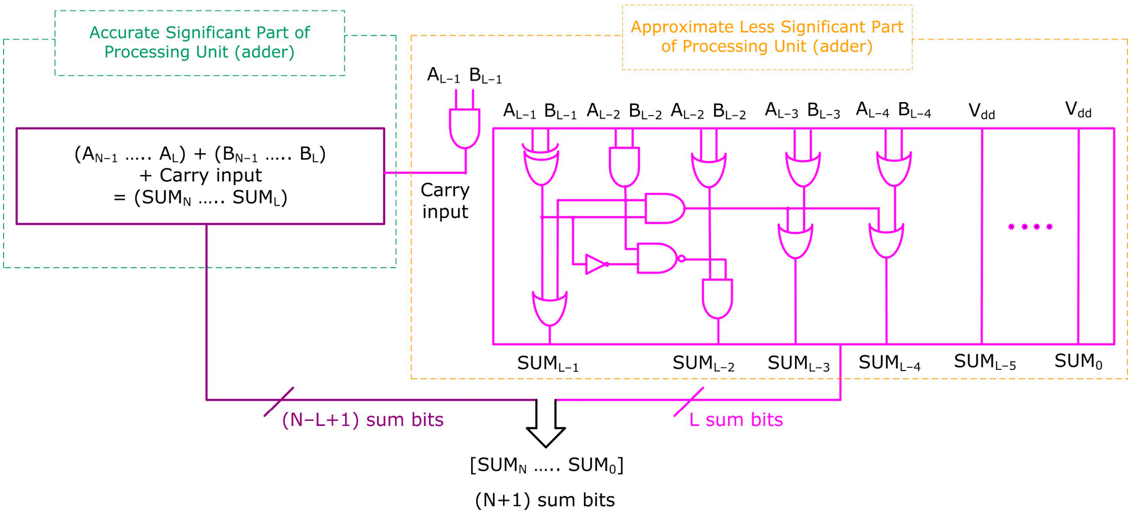
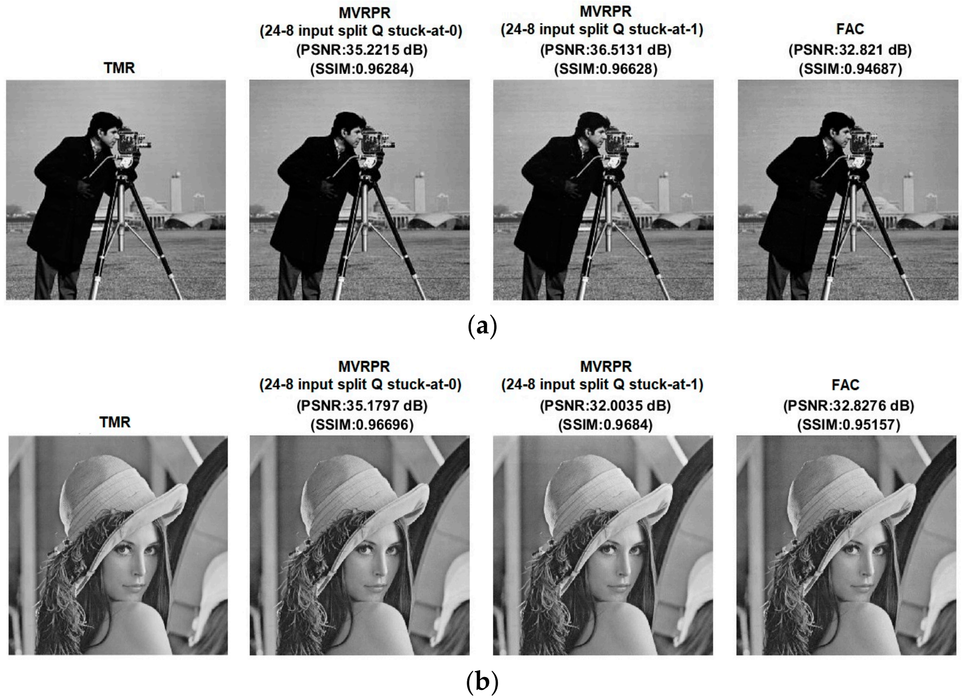

Accurate Significant Part of Processing Unit (adder)
Approximate Less Significant Part of Processing Unit (adder)
$A_{L-1}$ $B_{L-1}$
$(A_{N-1} ..... A_L) + (B_{N-1} ..... B_L)$ + Carry input = $(SUM_N ..... SUM_L)$
Carry input
$A_{L-1}$ $B_{L-1}$ $A_{L-2}$ $B_{L-2}$ $A_{L-2}$ $B_{L-2}$ $A_{L-3}$ $B_{L-3}$ $A_{L-4}$ $B_{L-4}$ $V_{dd}$ $V_{dd}$
$SUM_{L-1}$ $SUM_{L-2}$ $SUM_{L-3}$ $SUM_{L-4}$ $SUM_{L-5}$ $SUM_0$
(N–L+1) sum bits
L sum bits
$[SUM_N ..... SUM_0]$
(N+1) sum bits

**Figure 2.** The architecture of the N-bit inaccurate adder used for FAC implementation (in this work).

In the MVRPR adder, the carry overflow from the less significant adder part serves as the carry input to the triplicated significant adder parts. It has been hypothesized in [10] that if the intermediate carry signal (represented by Q in Figure 1b) input to the significant adder parts experiences a single-bit upset, the impact on the sum output of the significant adder part would be limited to a maximum difference of 1. However, the effect on the overall sum output was not analyzed in [10]. Additionally, the impact of a single-bit upset of Q on the addition of small- or medium-sized numbers was not examined in [10]. Furthermore, the effect of single-bit upset(s) on the sum bit(s) of the less significant adder part and their consequences on a practical application was not investigated. Moreover, in [10], the MVRPR adder and the TMR adder were only synthesized, and their design metrics were estimated and compared. The practical utility of the MVRPR adder was not demonstrated for any specific application. In [10], the partitioning of a processing unit (specifically, the adder) into two halves was suggested, which might not be suitable even for inherently error-tolerant applications, as noted in [20]. Our observation is that, according to MVRPR, a processing unit (such as the adder) should be divided into two parts of appropriate sizes based on the application's requirements. For the digital image processing application, it was found through trial and error that an unequal partitioning of an adder could be advantageous for an MVRPR implementation. In the case of image processing, splitting an MVRPR adder into a significant part with 24 bits and a less significant part with 8 bits may prove beneficial, as observed in [20].

The results of digital image processing corresponding to TMR, MVRPR, and FAC are depicted in Figure 3 for a selection of random digital images considered from [17]. Regarding MVRPR, the impact of a single-bit upset on the intermediate carry signal Q (output by the 8-bit less significant part and provided as input to the 24-bit triplicated significant parts) during digital image processing was analyzed and those findings are presented in Figure 3 for comparison. To assess the quality of the processed images, the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index Measure (SSIM) were calculated. PSNR serves as a general figure of merit for digital signal processing [21], while SSIM [22] specifically measures digital image processing quality. The ideal values are PSNR = ∞ and SSIM = 1 (decimal).

In Figure 3, the images processed using the accurate adder representative of TMR exhibit ideal values of PSNR and SSIM. For MVRPR, two sets of results are presented in Figure 3—one assuming the intermediate carry signal Q is stuck-at-0, and the other assuming Q is stuck-at-1 due to a single-bit upset. When Q alone experiences a single-bit upset in an MVRPR adder used for digital image processing, it is found to have minimal impact on the processed image quality, and the resultant images are considered acceptable. However, there is a possibility that the sum bits of the less significant part of an MVRPR adder might be affected by single-bit upset(s) concurrently with or independently from Q when Q experiences a single-bit upset as well. The effect of these scenarios on image processing was not analyzed in this study and would require further investigation; however, this investigation concerning [10] is beyond the scope of this research. In Figure 3a–f, the FAC implementation, despite using an approximate adder for the less significant part, is found to consistently produce high-quality images comparable to accurately processed ones. Furthermore, FAC ensures 100% tolerance to all single-bit upsets, like TMR, but unlike MVRPR which provides only moderate fault tolerance.



(**a**)

(**b**)

**Figure 3.** *Cont.*

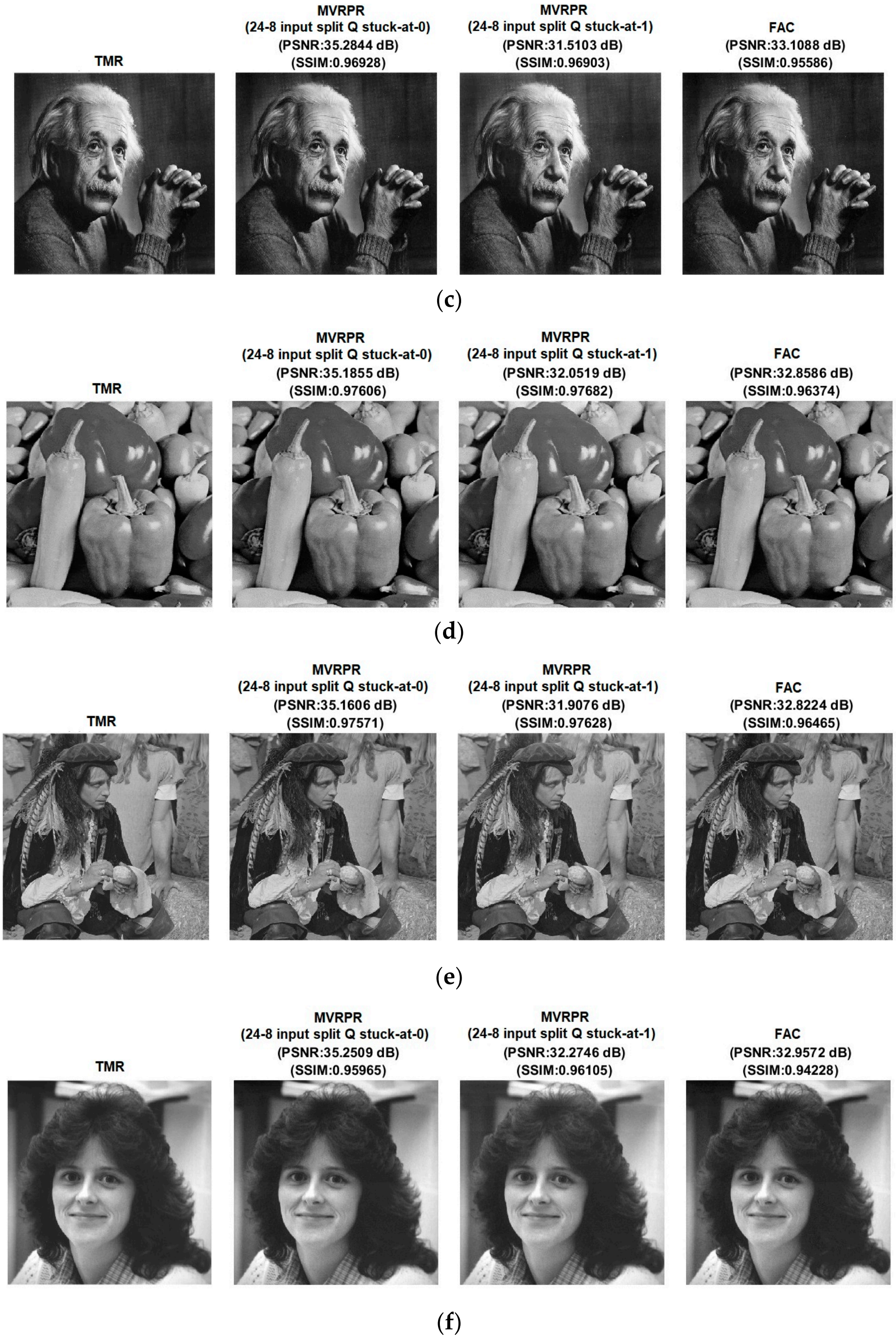


**Figure 3.** Results of digital image processing corresponding to TMR, MVRPR, and the proposed FAC based on experimentation with many images shown in (**a**–**f**). PSNR and SSIM values are also shown for the images. For MVRPR, a 32-bit accurate adder was optimally split into a 24-bit significant part and an 8-bit less significant part with the carry signal between these parts (Q in Figure 1b) assumed to be subject to single-bit upset as discussed in [10]. FAC is seen to consistently yield images with PSNR > 30 dB and SSIM close to unity.

## 5. Design Metrics

To physically implement the adders (processing units) used for digital image processing, we provided structural descriptions of each adder in Verilog hardware description language. The adders considered include: (i) An accurate 32-bit carry-lookahead adder (CLA) [23], representing a single adder, (ii) A 32-bit TMR adder, (iii) A 32-bit MVRPR adder, comprising a 24-bit significant part and an 8-bit less significant part (since the 24-8 input partition was found to be optimum as noted in the previous section), and (iv) A 32-bit FAC adder with a 22-bit significant part and a 10-bit less significant part (since the 22-10 input partition was found to be optimum as noted in the previous section). Both the TMR and MVRPR adders utilized the accurate CLA structure from [23], and the significant (accurate) part of the FAC adder was realized based on the same CLA structure. All adders were physically synthesized using a 28-nm CMOS standard digital cell library [24], with a typical low-leakage library specification employing a 1.05 V supply voltage and a 25 °C operating junction temperature. During simulation and synthesis, default wire load and a fanout-of-4 drive strength were assigned to all sum bits. Synopsys EDA tools were employed for synthesis, simulation, and the estimation of design metrics. Specifically, the Design Compiler was used for synthesis and to estimate the total area of the adders, including cells and interconnect area. To evaluate the performance of the adders, a test bench comprising over one thousand random inputs was supplied at a latency of 2 ns (500 MHz) to simulate their functionality using VCS. The switching activity was recorded during simulation, which was then utilized to estimate the total power dissipation using PrimePower. Further, PrimeTime was used to estimate the critical path delay for each adder. The design metrics of the adders, including area, power dissipation, and critical path delay, are given in Table 1.

**Table 1.** Design metrics of non-redundant and redundant adders, implemented using a 28-nm CMOS technology.

| Implementation | Area (µm$^2$) | Delay (ns) | Power (µW) |
|---|---|---|---|
| Single adder (CLA) | 527.45 | 1.13 | 91.7 |
| TMR adder | 1752.43 | 1.24 | 291.8 |
| MVRPR adder | 1490.61 | 1.28 | 247.6 |
| FAC adder | 1410.06 | 1.05 | 219.8 |

The single adder (i.e., accurate CLA) exhibits the lowest area, and power dissipation among all the adders considered, but it lacks fault tolerance. In comparison, the TMR adder experiences increased delay due to the additional majority voter delay in its critical data path. The TMR adder occupies 2.3× more area and dissipates 2.2× more power compared to the single adder since it includes two extra CLAs and a majority voting logic. In contrast, the MVRPR adder only triplicates its 24-bit significant part, while leaving its 8-bit less significant part unprotected. Consequently, the MVRPR adder has reduced area and power dissipation compared to the TMR adder, but it offers only moderate fault tolerance since its 8-bit less significant part is not safeguarded. The delay of the MVRPR adder is slightly higher than that of the TMR adder, primarily due to the loading effect experienced on its intermediate carry signal (i.e., Q shown in Figure 1b), which serves as the carry input to the three significant adder parts. The proposed FAC adder features a 22-bit accurate significant part, and its critical path is governed solely by this significant part. This advantage arises because, as shown in Figure 2, the accurate and approximate FAC adder parts are connected by a small carry input logic (represented by T in Figure 1c), defined as the logical conjunction of input bits $A_{L-1}$ and $B_{L-1}$. As a result, the FAC adder exhibits a reduced critical path delay even compared to the single adder (accurate CLA), and TMR and MVRPR adders. Additionally, the approximate 10-bit sum logic of the FAC adder results in a smaller silicon footprint, leading to reduced power dissipation in comparison to both TMR and MVRPR adders. The proposed FAC adder demonstrates significant improvements when compared to the TMR adder, including a 15.3% reduction in delay, a 19.5% decrease in area, and a 24.7% reduction in power dissipation. Compared to the MVRPR adder, the FAC adder exhibits an 18% reduced delay, a 5.4% smaller area, and an

11.2% reduction in power. Furthermore, the FAC adder outperforms the single adder with a 7.6% decrease in delay.

Two commonly used metrics for assessing the energy efficiency and design effectiveness of a digital logic design are the power-delay product (PDP) and the power-delay-area product (PDAP). Minimizing power, delay, and area is desirable, so it follows that minimizing PDP and PDAP is also desirable. We calculated PDP and PDAP values for both non-redundant and redundant adders from Table 1 and normalized the data. This involved dividing the actual PDP and PDAP values of all adders by the highest PDP and PDAP, respectively, corresponding to any adder. The normalized figures of merit (PDP in blue bars and PDAP in orange bars) are shown in Figure 4 below. Notably, the single (non-redundant) adder exhibits the lowest PDP and PDAP values but lacks fault tolerance. Among the redundant adders, the FAC adder demonstrates lower PDP and PDAP compared to the TMR and MVRPR adders. Specifically, in comparison to the TMR adder, the FAC adder achieves a 36.2% reduction in PDP and a 48.7% reduction in PDAP. Compared to the MVRPR adder, the FAC adder achieves a 23.8% reduction in PDP and a 31.1% reduction in PDAP. Hence, from Figures 3 and 4, and Table 1, it is inferred that FAC is better than MVRPR and is preferable to TMR for inherently error-tolerant applications.

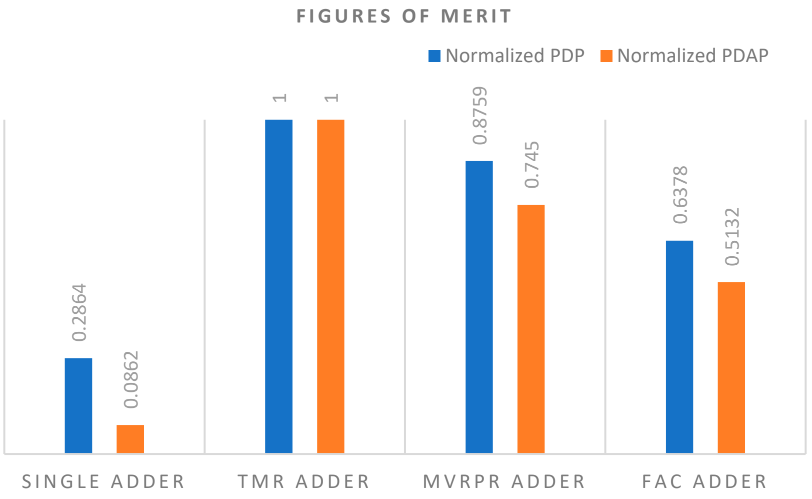


**Figure 4.** Normalized figures of merit (PDP and PDAP) of non-redundant and redundant adders.

## 6. Conclusions

In this article, a novel fault-tolerant design approach called FAC was introduced, which demonstrates comparable fault tolerance to NMR. In this work, specifically, a 3-tuple version of FAC was examined, allowing for a direct comparison with TMR and MVRPR for a digital image processing application. The image processing results obtained demonstrate the usefulness of FAC for inherently error-tolerant applications. Unlike MVRPR, FAC ensures 100% tolerance to any single fault or a faulty processing unit, like TMR. For the example implementation considered, FAC was found to achieve reductions in all design metrics compared to TMR and MVRPR, without compromising the fault tolerance. To cope with multiple faults (i.e., with more than one corresponding output bit affected or more than one processing unit failing), according to NMR, a higher-order version such as quintuple modular redundancy, septuple modular redundancy, etc., may have to be used; the corresponding equivalent according to our proposed architecture would be a 5-tuple version of FAC, a 7-tuple version of FAC, etc.

**Author Contributions:** Conceptualization, P.B.; methodology, P.B. and D.L.M.; validation, P.B.; formal analysis, P.B.; investigation, P.B. and D.L.M.; data curation, P.B.; writing—original draft preparation, P.B.; visualization, P.B.; supervision, D.L.M.; project administration, P.B. and D.L.M.; funding acquisition, D.L.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partially funded by the Singapore Ministry of Education (MOE), Academic Research Fund under grant numbers Tier-1 RG48/21 and Tier-1 RG127/22.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** All data are available within the manuscript.

**Acknowledgments:** The authors thank Raunaq Nayar for his help with image processing.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## References

1. Miskov-Zivanov, N.; Marculescu, D. Multiple transient faults in combinational and sequential circuits: A systematic approach. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2010**, *29*, 1614–1627. [CrossRef]
2. Baumann, R.C. Radiation-induced soft errors in advanced semiconductor technologies. *IEEE Trans. Device Mater. Reliab.* **2005**, *5*, 305–316. [CrossRef]
3. Rossi, D.; Omana, M.; Metra, C.; Paccagnella, A. Impact of aging phenomena on soft error susceptibility. In Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, Vancouver, BC, Canada, 3–5 October 2011.
4. Mahatme, N.N.; Bhuva, B.; Gaspard, N.; Assis, T.; Xu, Y.; Marcoux, P.; Vilchis, M.; Narasimham, B.; Shih, A.; Wen, S.-J.; et al. Terrestrial SER characterization for nanoscale technologies: A comparative study. In Proceedings of the IEEE International Reliability Physics Symposium, Monterey, CA, USA, 19–23 April 2015.
5. Balasubramanian, P.; Maskell, D.L. A fault-tolerant design strategy utilizing approximate computing. In Proceedings of the IEEE Region 10 Symposium (TENSYMP), Canberra, Australia, 6–8 September 2023.
6. Quinn, H.; Graham, P.; Krone, J.; Caffrey, M.; Rezgui, S. Radiation-induced multi-bit upsets in SRAM-based FPGAs. *IEEE Trans. Nucl. Sci.* **2005**, *52*, 2455–2461. [CrossRef]
7. Ruano, O.; Maestro, J.A.; Reviriego, P. A methodology for automatic insertion of selective TMR in digital circuits affected by SEUs. *IEEE Trans. Nucl. Sci.* **2009**, *56*, 2091–2102. [CrossRef]
8. Gomes, I.A.C.; Martins, M.G.A.; Reis, A.I.; Kastensmidt, F.L. Exploring the use of approximate TMR to mask transient faults in logic with low area overhead. *Microelectron. Reliab.* **2015**, *55*, 2072–2076. [CrossRef]
9. Arifeen, T.; Hassan, A.S.; Moradian, H.; Lee, J.A. Input vulnerability-aware approximate triple modular redundancy: Higher fault coverage, improved search space, and reduced area overhead. *Electron. Lett.* **2019**, *54*, 934–936. [CrossRef]
10. Ullah, A.; Reviriego, P.; Pontarelli, S.; Maestro, J.A. Majority voting-based reduced precision redundancy adders. *IEEE Trans. Device Mater. Reliab.* **2018**, *18*, 122–124. [CrossRef]
11. Han, J.; Orshansky, M. Approximate computing: An emerging paradigm for energy-efficient design. In Proceedings of the 18th IEEE European Test Symposium, Avignon, France, 27–31 May 2013.
12. Venkataramani, S.; Chakradhar, S.T.; Roy, K.; Raghunathan, A. Approximate computing and the quest for computing efficiency. In Proceedings of the 52nd ACM/EDAC/IEEE Design Automation Conference, San Francisco, CA, USA, 8–12 June 2015.
13. Mittal, S. A survey of techniques for approximate computing. *ACM Comput. Surv.* **2016**, *48*, 1–33. [CrossRef]
14. Balasubramanian, P.; Mastorakis, N.E. Power, delay and area comparisons of majority voters relevant to TMR architectures. In Proceedings of the 10th International Conference on Circuits, Systems, Signal and Telecommunications, Barcelona, Spain, 13–15 February 2016.
15. Ma, J.; Hashemi, S.; Reda, S. Approximate logic synthesis using Boolean matrix factorization. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2022**, *41*, 15–28. [CrossRef]
16. Zhu, N.; Goh, W.L.; Zhang, W.; Yeo, K.S.; Kong, Z.H. Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing. *IEEE Trans. VLSI Syst.* **2010**, *18*, 1225–1229.
17. Available online: http://imageprocessingplace.com/root_files_V3/image_databases.htm (accessed on 26 March 2023).
18. Balasubramanian, P.; Nayar, R.; Maskell, D.L. An approximate adder with reduced error and optimized design metrics. In Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems, Penang, Malaysia, 22–26 November 2021.
19. Balasubramanian, P.; Nayar, R.; Maskell, D.L.; Mastorakis, N.E. An approximate adder with a near-normal error distribution: Design, error analysis and practical application. *IEEE Access* **2021**, *9*, 4518–4530. [CrossRef]
20. Balasubramanian, P. Analysis of redundancy techniques for electronics design—-case study of digital image processing. *Technologies* **2023**, *11*, 80. [CrossRef]
21. Bovik, A. *Handbook of Image and Video Processing*, 2nd ed.; Academic Press: Cambridge, MA, USA, 2005.
22. Zhou, W.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612.

23. Balasubramanian, P.; Mastorakis, N.E. High-speed and energy-efficient carry look-ahead adder. *J. Low Power Electron. Appl.* **2022**, *12*, 46. [CrossRef]
24. Synopsys SAED_EDK32/28_CORE Databook. Revision 1.0.0. January 2012. Available online: https://www.synopsys.com/academic-research/university.html (accessed on 26 April 2023).

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.