

Hardware-Centric Fault Tolerance: Bridging ARP4754A Safety Objectives with DO-254 ASIC Implementation

Eren ALTUN^a

^aASIC and FPGA Expert in Aerospace, Toulouse, FRANCE

KEY WORDS

DO-254,
ARP4754A,
TMR (Triple Modular Redundancy),
Single Event Upsets (SEU),
Safety-Critical ASIC,
Deterministic Hardware,
ARP4761,
Design Assurance Level (DAL),
Deep submicron silicon processes,
ate-level hardware implementation

ABSTRACT

As autonomous aerospace platforms transition toward highly integrated and intelligent systems, the gap between high-level safety objectives and gate-level hardware implementation becomes increasingly critical. This paper explores a hardware-centric approach to fault tolerance, specifically focusing on the alignment of ARP4754A system-level safety requirements with the rigorous design assurance flows of DO-254 for custom ASIC development. Central to this research is the mitigation of Single Event Effects (SEE) in deep submicron silicon processes a major challenge for mission-critical AI applications. We analyze the implementation of Triple Modular Redundancy (TMR) and majority voter logic as primary mechanisms for ensuring deterministic behavior under radiative stress. Furthermore, the study details a methodology for transforming ARP4761 safety assessments into verifiable RTL-level constraints, ensuring that partitioning and redundancy schemes effectively eliminate Single Points of Failure (SPoF). By integrating fault-injection testing and formal verification into the early stages of the silicon lifecycle, we demonstrate a path toward DAL-A compliant hardware that maintains peak performance without compromising safety. This research provides a strategic roadmap for architects seeking to deploy robust, self-diagnostic, and certifiable silicon for the next generation of flight-critical edge intelligence.

1 Introduction

The rapid evolution of autonomous flight technologies ranging from Urban Air mobility (UAM) to unmanned tactical platforms has created an unprecedented demand for onboard compute power. As Artificial Intelligence (AI) and complex decision-making algorithms move to the edge, the industry is shifting away from general-purpose processors toward custom ASICs (Application Specific Integrated Circuit) to meet stringent Power-Performance-Area (PPA) requirements.

However, this technological leap introduces a fundamental challenge: the Safety Gap. Modern silicon, manufactured in deep submicron processes, is increasingly susceptible to transient fault such as SEU (Single Event Upsets) caused by atmospheric radiation. In a mission-critical environment, a single flipped bit in a neural engine or control register can lead to catastrophic system failure. To bridge this

gap, engineers must align the high-level system safety objectives defined in ARP4754A with rigorous gate-level hardware design assurance mandated by RTCA/DO-254.

This white paper examines a hardware-centric methodology for fault tolerance. We explore how architectural decisions specifically TMR (Triple Modular Redundancy) and deterministic logic partitioning can be derived directly from ARP4761 safety assessment. By embedding reliability into the physical RTL architecture rather than relying on software-level mitigations, we can achieve the deterministic behaviour required for DAL-A certification.

The following sections will detail the transformation of system-level “safety goals” into silicon-level of “design-requirements”, providing technical roadmap for developing next-generation of trustworthy aerospace hardware.

2 From System Safety to Silicon Requirements (The ARP-to-DO Link)

2.1 Section 1 : Requirements Decomposition – From ARP4754A to Silicon

The development of a certifiable ASIC does not start with RTL code, but with a thorough understanding of aircraft-level risks. End-to-end traceability from system safety analysis down to logic gates is the cornerstone of avionics trust.

From Aircraft Risk to RTL Constraints (From ARP4754A to Do-254)

Everything starts with the ARP4761 safety analysis (FHA / PSSA), which identifies failure conditions. Let us consider the example of an AI-assisted engine control function:

Failure Condition: “Undetected transmission of an erroneous thrust command.”

Severity Classification: *Catastrophic (DAL A)*.

Within the ARP4754A framework, this safety objective is allocated to the ASIC as hardware requirements.

As the hardware architect, I derive concrete DO-254 design constraints from these requirements: a failure probability objective of 10^{-9} per flight hour mandates the implementation of real-time error detection mechanisms directly at the RTL level.

Consequently, every critical data path is designed not only for performance, but also includes dedicated monitoring logic able to drive the system into a fail-safe state within a limited number of clock cycles.

Analysis of Independence and Common Causes (CCA)

One of the key challenges of system-on-chip (SoC) integration is to ensure that multiple critical functions cannot fail simultaneously due to a single initiating event.

In order to meet the ARP4754A independence requirement, a rigorous Common Cause Analysis (CCA) is performed at the silicon level:

Physical Segregation (Floorplanning): Keep-out zones are imposed at the die level to physically separate redundant cores, thereby mitigating the effects of localized manufacturing defects or severe thermal gradients.

Data and Control Flow Isolation: Hardware firewalls are implemented to ensure that logic faults or bus congestion in non-critical functions cannot propagate to resources shared with DAL A functions.

Domain Diversity: Clock and reset distribution networks are strictly segregated. This ensures that failures in power or clock distribution do not result in a common-mode failure capable of incapacitating all redundant logic.

The “Safety Manual” : Ensuring Integration Safety

A certifiable ASIC shall not be treated as a “black box,” but as an integral part of a larger ecosystem. The objective of the design process is the delivery of a comprehensive Safety Manual. This document is critical for the system integrator at LRU level, as it specifies:

Safety Assumptions: The set of operating conditions and external monitoring functions (e.g., external watchdogs, voltage supervision) that shall be ensured by the system integrator to support final certification.

Diagnostic Mechanisms: An exhaustive inventory of status registers and hardware alerts that shall be monitored by the software to properly handle failures detected by the ASIC.

Failure Coverage Analysis: Documented evidence demonstrating the device’s capability to detect its own failures, allowing the system integrator to complete the system safety case.

2.2 Case Study: Optimized Triple Modular Redundancy (TMR) Architecture

In AI applications for autonomous flight, responsiveness and reliability are inseparable. This section illustrates why a silicon-implemented Triple Modular Redundancy (TMR) architecture is the only approach able to satisfy DAL A requirements.

Hardware vs Software Redundancy : Advantages of Deterministic Design

The selection between software-managed redundancy and hardwired (hardware) redundancy significantly affects the system timing budget.

- **Decision Latency:** In software implementations, comparing outputs from three cores involves interrupts, bus arbitration, and complex synchronization, resulting in non-deterministic latency. In the ASIC architecture, the hardware voter executes this operation within a single clock cycle (on the order of a few nanoseconds), without imposing any CPU overhead.
- **Metrics Comparison (Typical Estimates):**

Metric	Software TMR	Hardware TMR (ASIC)	Gain
Vote Latency	~50 μ s to 500 μ s	< 10ns	x 5000
Reliability (MTBF)	Susceptible to OS and compiler bugs	Proven by construction at RTL level	High
Power consumption	High (three active CPUs)	Optimized (Dedicated logic)	~40%

Key Insight: In a Sense & Avoid detection and avoidance algorithm, hardware latency provides near-instantaneous response, while software-based implementations may introduce critical delays when transient faults occur.

Architecture of a Robust Voter: Avoiding Single Points of Failure

The voter circuitry represents the most critical element of the architecture, as its failure would negate all redundancy. To prevent the voter from becoming a Single Point of Failure (SPoF), a triplicated voter architecture is implemented.

Triple Voting: Three independent voters are implemented rather than a single instance. Each voter output independently drives a downstream redundant path.

Majority Logic: The voting decision is performed using a purely combinational Boolean expression:

$$Out = (A \cdot B) + (B \cdot C) + (A \cdot C)$$

Divergence Monitoring: Dedicated comparators generate an interrupt as soon as a discrepancy is detected between channels, enabling early identification of a faulty element before system-level integrity is affected.

Strategy for Managing Single Event Effects (SEE)

In radiation-prone environments, energetic particles may induce Single Event Upsets (SEUs) by flipping the state of memory elements. The protection strategy is implemented at two complementary levels:

- **Memory Scrubbing and Error Correction Coding (ECC):** ECC is applied to caches and large AI weight storage elements. A continuous memory scrubbing mechanism detects and corrects bit errors, thereby preventing the accumulation of faults leading to multi-bit upsets.
- **Radiation-Hardened-by-Design (RHBD) Registers:** Critical finite state machines (FSMs) and counters are implemented using hardened flip-flops, such as Dual Interlocked Cell (DICE) structures. These cell topologies provide intrinsic resistance to ion-induced bit flips.

Additional SEE mitigation strategies could be envisaged, including AI-level algorithmic redundancy, adaptive scrubbing of critical registers, temporal output signature verification, or internal AI-driven fault detection mechanisms. These topics are outside the scope of the present white paper and will be discussed in a future, dedicated document.

3 Evidence through Verification and Validation (V&V)

Under DO-254, verification entails more than showing that the system functions correctly; it is intended to demonstrate safe behavior under fault conditions. For a DALA ASIC, this demonstration relies on two fundamental pillars: robustness against physical disturbances and end-to-end traceability of requirements.

Fault Injection Techniques: Simulating Unpredictable Failure Modes

Fault injection is a key verification technique for certifiable ASICs. Its purpose is to deliberately introduce errors or abnormal conditions into the design to validate that the system behaves safely under unexpected scenarios, without waiting for in-flight testing.

Validation of TMR and Error-Correction Mechanisms:

To validate the effectiveness of our Triple Modular Redundancy (TMR) architecture and correction mechanisms, fault injection is performed at the RTL level to simulate the impact of cosmic particles (protons, heavy ions) on the silicon.

SEU Simulation (Single Event Upsets):

Using formal verification and simulation tools, bit-flips are randomly injected into critical registers and data paths while the AI performs its computations.

Success Criteria (Pass/Fail):

- **Self-Correction:** The system must continue to provide correct outputs without interruption, thanks to the TMR voter.
- **Detection:** Divergence alerts must be raised in the status registers to notify the system that maintenance or intervention is required.
- **Non-Propagation :** Faults must never corrupt adjacent safety-critical domains.

Hardware and RTL-Level Faults:

At the RTL or gate level, faults can also simulate stuck-at faults or transient glitches, ensuring that fail-safe mechanisms are triggered as intended.

Software-Level Faults:

For designs interacting with embedded software, faults can be injected in input streams, memory

interfaces, or bus transactions to verify that error detection, correction, and recovery mechanisms function correctly without violating safety objectives.

SEE Emulation:

Single Event Effects (SEEs), such as SEUs and Single Event Transients (SETs), are emulated in the testbench to confirm that ECC, RHBD registers, and TMR voters effectively mitigate transient and permanent errors.

Coverage and Traceability:

Every injected fault is traced back to the original system safety requirements. This ensures that all critical failure modes identified in the FHA/PSSA have been exercised and validated.

Result:

This fault injection campaign provides both statistical and deterministic evidence that the residual error rate complies with ARP4761 safety objectives (10^{-9} per flight hour).

Traçabilité Totale : De l'Exigence à la Porte Logique

No logic element may exist unintentionally in a certifiable ASIC. We implement a rigorous bidirectional traceability framework, linking each design element to its originating safety requirement.

- **Top-Down Traceability:** Requirements derived from ARP4754A are translated into hardware specifications and further into identifiable RTL code blocks. For instance, a safety requirement for “temperature monitoring” is traced down to the analog sensor and its digital control logic.
- **Bottom-Up Traceability:** An analysis ensures the absence of “unnecessary functions.” Each Verilog/VHDL module must be associated with a documented requirement, preventing hidden gates or extraneous functionality that could cause unexpected in-flight behavior.
- **Verification Matrix:** A comprehensive matrix confirms that all requirements have been verified through simulation, formal methods, or fault injection, ensuring complete coverage.

Conclusion: Certified Silicon as the Foundation for the Future of Autonomous Flight

The integration of artificial intelligence and advanced decision-making systems into aerospace platforms represents a pivotal milestone. Yet, as this white paper has illustrated, raw computational performance must never compromise safety. The central challenge of the current decade is not only achieving computational power but ensuring that it is fully certifiable.

This study has shown that aligning high-level system objectives (ARP4754A) with physical implementation (DO-254) is now essential. By pursuing a hardware-centric design methodology, we address the inherent limitations of conventional software approaches and provide practical, verifiable solutions:

Determinism by Design: Catastrophic failure conditions are explicitly mapped into RTL logic structures, removing the unpredictability inherent in generic architectures.

Physical Resilience: A Triple Modular Redundancy (TMR) architecture, combined with robust voter mechanisms, meets the most demanding failure probability targets (DAL A) while providing active mitigation against atmospheric radiation-induced effects (SEE/SEU).

Assured Independence: Rigorous Common Cause Analysis (CCA) and strict physical partitioning at the silicon level guarantee that the high-density integration of functions on a single SoC does not compromise flight-critical integrity.

In conclusion, achieving full autonomy demands a fundamental paradigm shift. The custom ASIC has evolved beyond a simple electronic component to become a cornerstone of avionics safety. By providing an optimized design, a comprehensive Safety Manual, and end-to-end traceability, we equip system integrators with a clear and secure path to FAA/EASA certification.

Innovation in aerospace begins at the transistor level. Through precise control of every logic gate and clock cycle, we establish the foundation for autonomous flight, embedding trust directly into the silicon.

References:

- [1] **SAE ARP4754A** : Guidelines for Development of Civil Aircraft and Systems.
- [2] **RTCA DO-254 / EUROCAE ED-80** : Design Assurance Guidance for Airborne Electronic Hardware.
- [3] **SAE ARP4761** : Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment.
- [4] **NASA Handbook (NASA-HDBK-8739.23)** : Complex Electronics Handbook for Assurance Representatives.
- [5] **Velazco, R., Fouillat, P., & Reis, R.**, Radiation Effects on Embedded Systems, Springer, 2007.