# Are software upsets really a cosmic mystery?


**Last year, an Airbus A320 experienced an in-flight upset. It is suspected that a recent software update made the aircraft susceptible to disruption caused by cosmic radiation. Dewi Daniels investigates the issue, describing the software/hardware interactions and explores what regulations and standards exist to help mitigate these types of event.**

On the 30th October 2025, a JetBlue Airways Airbus A320 experienced an in-flight upset while cruising at 35,000 feet [1]. Four flight attendants and 18 passengers sustained minor injuries.

The incident was notified to the National Transportation Safety Board (NTSB), who opened a formal investigation. The NTSB investigation is still ongoing [2].

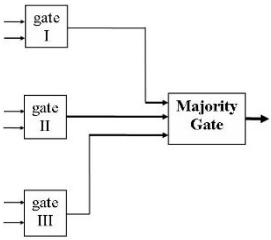
Airbus determined that a recent software release to the Elevator Aileron Computer (ELAC) was susceptible to Single Event Upsets (SEUs). An SEU occurs when a single, high-energy particle (like a cosmic ray or neutron) flips a bit of data (changing a 0 to a 1 or vice-versa) in a memory cell or a Central Processing Unit (CPU) register. The ELAC had been developed by Thales [2, 3].

Airbus issued an Alert Operators Transmission (AOT) A27N022-25 [3] on the 28th November 2025 requiring that all Airbus A319, A320 and A321 aircraft fitted with ELAC B standard L104 be reverted to ELAC B standard L103+.

The European Union Aviation Safety Agency (EASA) issued Emergency Airworthiness Directive (AD) No. 2025-0268-E [4] and the Federal Aviation Administration (FAA) issued Emergency AD 2025-24-51 [5] on the same day.

It may seem surprising that a software update would make an aircraft susceptible to SEUs. After all, an SEU is a hardware event. It's true that SEUs can be mitigated through hardware design – space satellites often use radiation-hardened microprocessors and memory devices. However, these are not used in airliners (and not even in some satellites) because radiation-hardened devices are bigger, heavier, slower and more power-hungry. Airliners do typically use Error-Correcting Code (ECC) memory, which helps to mitigate against SEUs, though ECC memory does not mitigate against SEUs that occur in the microprocessor itself, such as in cache memory and in registers.

The main mitigation against SEUs in avionic systems is using redundancy in system design. For example, consider triple modular redundancy (TMR). This is where three channels perform the same calculation and a voter is used to decide on a single output. An SEU will only affect one of the channels. The other two channels will produce the same, correct output and will therefore outvote the channel that suffered the SEU.




As an additional layer of defence, SEUs can be mitigated by software design. For example, I was one of the software developers on the Airbus A380 Landing Gear Extraction and Retraction System (LGERS) in the early 2000s [6]. While the main mitigation against SEUs was the fact that LGERS is a multiple redundant system, EASA also required that we mitigated against SEUs in our software design. We conducted a software hazard analysis to identify the critical data and we kept three copies of all critical data (one of the copies was the one's complement of the other two). This meant that we were able to detect when an SEU had corrupted one of the copies and restore the correct value from one of the other two copies. We recorded the number of data corruptions in non-volatile random-access memory (NVRAM). It would be interesting to know how often SEUs have occurred in practice – I suspect that SEUs occur quite often, especially at cruising altitude.

**"I suspect that Single Event Upsets occur quite often, especially at cruising altitude"**

EASA Certification Memorandum CM–AS-004 [7] provides guidance on how to consider the effects of Single Event Effects (SEEs) on aircraft systems and equipment. EASA has also published a Safety Information Bulletin (SIB) 2012-10R1 on SEEs [8]. Meanwhile, the FAA published Single Event Effect Mitigation Techniques Report DOT/FAA/TC-15/62 [9]. An SEE is the umbrella term for any measured change or failure in an electronic device caused by a single, high-energy particle (like a cosmic ray or neutron) hitting a sensitive spot on a chip. SEEs can either be soft errors, which are temporary glitches or hard errors, which cause permanent damage to the chip. An SEU is a specific type of soft error that occurs when a particle flips a bit of data, as described above.

RTCA DO-248C [10]/EUROCAE ED-94C [11] Discussion Paper (DP) #21 provides an explanation as to how SEUs can be mitigated in software. It explains that the most common protection mechanisms against SEUs are error detection and correction mechanisms, such as:

- Parity
- Cyclic redundancy code
- Hamming code
- Reed-Solomon code
- Convolutional codes
- Watchdog timers
- Voting
- Minimising the use of unprotected cache memory
- Periodically flushing the cache
- Storing triple versions of critical values
- Minimizing the use of non-ECC random access memory (RAM) on the processor
- Minimising stack and heap usage
- Using stack and heap overrun protection
- Minimising the use of variables whose corruption could significantly impact the behaviour of the software
- Periodically performing checksums on critical areas of memory
- Performing checksums on permanently stored data
- Providing a mechanism where the application can be reset

DP #21 also explains there are additional practices that can be used where appropriate, including:

- Repeated calculations to overcome transient errors
- Define constants in read only memory (ROM)
- Do not rely on RAM data to be accurate
- Write output discretes to hardware latches every frame
- Avoid using equality
- Continuously check the configuration state of devices that have been initialised by software
- Filter input data
- Whenever a built-in test equipment (BITE) or built-on test (BIT) detects a failure, rerun the test a second time to confirm the failure
- When using bi-directional I/O ports, re-assert the configuration of the I/O port often
- Where registers are used to define the CPU configuration, the configuration should be re-asserted often
- Pointers should be range checked when used.

SEE and SEU have been known about for a long time and are the topic of guidance from FAA and EASA. It is surprising that this software update resulted in an in-flight upset so soon after it was released. It suggests that SEU are relatively common but also that most avionic system and software designs are robust in the presence of SEUs.

The NTSB investigation is still ongoing and no further information has been released by Airbus. We can therefore only guess at the cause of the in-flight upset.

I suspect that ELAC B standard L104 introduced new functionality that was not cross-checked by the other channel(s) and that there was insufficient software mitigation (e.g. they did not keep triple versions of critical data).

Fortunately, the problem was resolved by reverting to the previous standard L103+. I assume that the intent is to develop a new software standard that re-introduces the new functionality in a manner that is not susceptible to SEUs.

**"It is surprising that this software update resulted in an in-flight upset so soon after it was released"**

# References


[1] https://aviation-safety.net/wikibase/560989, retrieved 10 March 2026.

[2] Aviation Investigation Preliminary Report, Incident Number ENG26LA004, National Transportation Safety Board, https://data.ntsb.gov/carol-repgen/api/Aviation/ReportMain/GenerateNewestReport/201942/pdf, retrieved 10 March 2026.

[3] Airbus Alert Operator Transmission (AOT) A27N022-25 Rev 01, 28 November 2025, https://downloads.regulations.gov/FAA-2025-5395-0002/attachment_1.pdf, retrieved 10 March 2026.

[4] EASA Emergency Airworthiness Directive No. 2025-0268-E, 28 November 2025, https://ad.easa.europa.eu/blob/EASA_AD_2025_0268_E.pdf/EAD_2025-0268-E_1, retrieved 10 March 2026.

[5] FAA Emergency Airworthiness Directive # 2025-24-51, 28 November 2025, https://drs.faa.gov/browse/excelExternalWindow/DRSDOCID1701465859202511290342430001?modalOpened=true, retrieved 10 March 2026.

[6] Safety Aspects of a Landing Gear System, Dewi Daniels, Fourteenth Safety-critical Systems Symposium, Bristol, UK · Feb 1, 2006.

[7] Single Event Effects (SEE) Caused by Atmospheric Radiation, Certification Memorandum No. CM-AS-004 Issue 01, European Union Aviation Safety Agency, 8 January 2018, https://www.easa.europa.eu/sites/default/files/dfu/CM-AS-004%20Issue%2001.pdf, retrieved 10 March 2026.

[8] Single Event Effects on Aircraft Systems caused by Atmospheric Radiation, Safety information Bulletin No. 2012-10R1, European Union Aviation Safety Agency, 29 April 2021, https://ad.easa.europa.eu/blob/EASA_SIB_2012_10_R1.pdf/SIB_2012-10R1_1. Retrieved 10 March 2026.

[9] Single Event Effect Mitigation Techniques Report, DOT/FAA/TC-15/62, Federal Aviation Administration, February 2016, https://www.faa.gov/sites/faa.gov/files/aircraft/air_cert/design_approvals/air_software/TC-15-62.pdf, retrieved 10 March 2026.

[10] Supporting Information for DO-178C and DO-278A, DO-248C, RTCA, Inc., 13 December, 2011

[11] Supporting Information for ED-12C and ED-109A, ED-94C, EUROCAE, January 2012


Image attribution:
top image: AI generated (Midjourney and Gemini)
Triple Modular Redundancy: IjonTichyIjonTichy, CC0, via Wikimedia Commons


**Dewi Daniels, Software Safety Limited**

Dewi is a Chartered Engineer with nearly 45 years' experience in the development and verification of safety-critical software. Dewi was one of the authors of DO-178C/ED-12C. He is currently a member of the RTCA/EUROCAE Forum on Aeronautical Software and one of the UK experts on the IEC 61508 committee.