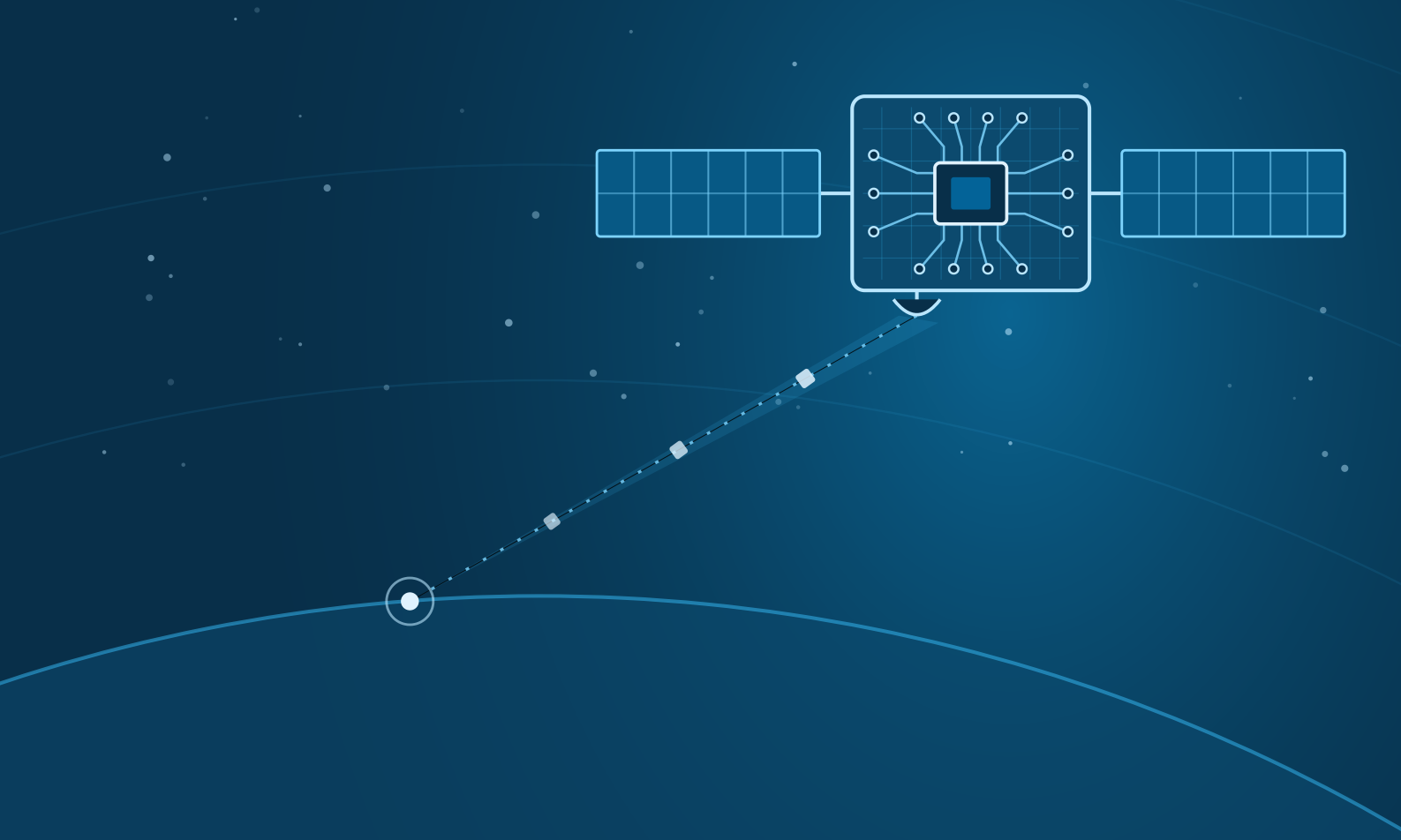


THE SATELLITE SERIES, COMPUTER BY COMPUTER

# Computing in Orbit

*From the guidance computer to the data center node*

Computing in orbit is a trade of four budgets, energy, heat, radiation and downlink, against cost and risk, decided per workload and against a stated objective. Flight computers from Apollo to today show how radiation, qualification and mitigation shaped each choice. Processing on board pays only when the trade study says so; announced data center plans are kept apart from anything flown.

**Juan Cardena**

Enterprise AI & Agentic Systems Architect · blog.jcardena.com

## CONTENTS


BEFORE THE COMPUTERS

Reader's Brief: Four Budgets, Three Places, Separate Status Fields

Prologue: Two Computers, Fifty-Seven Years Apart

**01** Where Should the Computing Happen?

PART I: COMPUTERS THAT HAD TO FLY

**02** 1962 to 1996: Computers That Had to Fly, Line by Line

**03** Radiation: Effects, Environment and Assurance

**04** Mitigation, Layer by Layer, and What It Leaves

PART II: PARTS AND EVIDENCE

**05** Processors and the Avionics Around Them, 1990 to 2026

**06** Part Choice and Qualification Evidence: A Matrix, Not a Ladder

**07** Space FPGAs: Where the Configuration Lives, What Protects the Rest, and the Process Under It

PART III: PROCESSING IN ORBIT

**08** Software-Defined Payloads: What Can Be Changed from the Ground

**09** Processing the Data On Board: Compress, Filter, Decide

**10** On-Board AI: Accelerators, Models and Trust

**11** When Processing in Orbit Is Worth It: The Trade Study

**12** Computing Platforms in Orbit: Launched, Demonstrated, Offered, Announced

Epilogue: Reading the Next Announcement

APPENDICES

**A** The Trade Sheet and the Link Ledger: Blank Pages and Index

**B** Timeline of Flight Computers by National Line, 1962 to 2026, with Cases Sought

**C** Glossary

**D** Dated Register: Parts, Qualification and Computing in Orbit as of November 2026

**E** Sources

BEFORE THE COMPUTERS

# Reader's Brief: Four Budgets, Three Places, Separate Status Fields




Every satellite system divides its work between the user's end, the satellite and the ground. This series asks of each system how the work was divided, what limits drove that split, and whether it worked. This book asks the question of one kind of work: computing. Where should a computation run, on board, on another satellite or on the ground, and what does each choice cost?

## Four budgets, three places, four kinds of computing

A computer in orbit lives inside four budgets: **energy** (power in watts, energy in watt-hours, from arrays, batteries or, on some deep space probes, radioisotope generators), **heat** (nearly every watt drawn ends as heat, which must in the end be radiated away), **radiation** (damage that builds up, and single particles that flip, glitch or short a circuit) and **the link** (what can go up and come down, and when). Cost, schedule and the proof a mission asks for sit on top of them.

A computation has three places to run: on the ground after the downlink, on board, or on another satellite. And "the computer on the satellite" means one of four things that this book never mixes: the platform computer that keeps the satellite alive, communication processing, payload data processing, and hosted general-purpose computing rented to customers. Chapter 1 sets out all three ideas with their units.

## Status fields, never a single word

A system in this book never gets one status word. It gets fields, and one system can fill several: **launched** (date and vehicle); **demonstrated function** (what a source says it did in orbit); **current health**; **service access** (who can buy or apply, in which country, since when); **qualification** (standard, level, date, issuer); **announced**; **retired, failed or deorbited**. "Launched" does not mean "works", and "qualified" does not mean "flown". Where no source was found, the text says so with the scope of the search and its date, because an absence in the sources read is not proof of absence.

## Four kinds of proof, and how a part's numbers are written

A claim about a part or a computer rests on one of four kinds of proof, and chapter 6 shows why they are not interchangeable: a **maker's statement**; a **ground test report** (whose, when, which lot, under which conditions); **flight data**; and a **certification** against a standard. "Tested to 25 krad" and "qualified to QML Class V" are different sentences, and so are "launched" and "flight proven".

A radiation figure in this book is written with its conditions: the dose or particle energy, the test, the lot where the source gives it, the failure criterion, the date and whose test it was. A power figure says what kind of figure it is: a nominal value, a thermal design power or a measured draw. A figure without its conditions is quoted as the source gives it, labeled, and not used to decide anything.

## The trade sheet and its verdict

The book's own tool is the trade sheet: fifteen lines, filled for the on-board case and the ground case, from the objective and its payer through data, link, energy, heat, radiation, mass, updates and trust, lifecycle cost and the ground alternative, to uncertainty, sensitivity and the limits of the sheet. It ends in a verdict of one form only: "advantageous for whom, against which objective, if which assumption holds", or "not advantageous", or "cannot be decided from public sources". A sheet that leaves out the reader or the objective has not reached a verdict. Chapter 1 teaches the sheet on an invented example; chapters 10 and 11 fill it for real and illustrative workloads; Appendix A holds the blank.

Where computing changes a link, the book also uses the series form, the Link Ledger: page 1 describes the service in ten fields, and page 2, which in the first book of this series, *The Bit's Path*, is a hop budget in decibels, is not printed: this book's ledgers are service ledgers, and the data volume (produced, stored, sent per contact, removed by processing) is worked in the chapters' tables, with a blank form in Appendix A. This book prints no filled budget in decibels; Appendix A keeps the blank form of the series.

## Where each number comes from

Every number on a sheet or a ledger page carries one of seven classes, with its source and date: **measured** (by a named party, on a date), **calculated** (from stated inputs by a stated formula), **simulated** (by a named model), **operator claim** (a maker or operator stated it), **standard** (a published standard, clause named), **assumption** (printed "assumption, to verify", with what would settle it) and **published history** (an official agency history or labeled scholarship). No number in this book was measured by its author. Trade press is labeled as trade press, and a preprint as a preprint.

Dates are kept apart in the same way: the date a thing happened, the date a source reported it, and the date the source was read. Volatile facts, such as a part's qualification or a platform's status, carry their date in the text and are collected in the dated register of Appendix D.

## How the book is built

The prologue sets two computers side by side, 1969 and 2026. Chapter 1 sets out the frame and the sheet. Part I follows the computers that had to fly from 1962 to 1996, national line by national line, then radiation and the mitigations against it. Part II covers processors and the avionics around them, how parts are chosen and qualified, and space FPGAs. Part III covers processing in orbit: payloads that can be changed from the ground, data processed on board, on-board AI, the trade study, and the computing platforms that are launched, demonstrated, offered or only announced. The epilogue applies the four budgets and the status fields to the next announcement.

Each numbered chapter carries a failure, a recovery or a trade-off from the record, told from its sources; a security note on who can command the computer and change its software, who can interfere, and how it is recovered; and last a **Monday rule**: one sentence a reader can act on the next working week. The Monday rules are the author's practice, offered as such.

# Prologue: Two Computers, Fifty-Seven Years Apart




In 1969 the computer that guided Apollo to the Moon weighed 70.1 pounds, drew 70 watts and ran on a one megahertz clock, according to NASA's history of its computers (MIT's final report on the computer is exact: a 2.048 MHz oscillator divided to a 1.024 MHz reference). Its fixed programs were woven into core rope memory at the factory, months before each flight, and could not be changed easily once the rope was sealed; only a small erasable memory could be written in flight. On 11 January 2026, according to its operator, two computing nodes went to low orbit to be sold as a service, built to accept new software and new models by radio after launch. Fifty-seven years separate the two. This book is about what changed between them, and what did not.

### 1969: the ground knew where the spacecraft was

The Apollo Guidance Computer was designed by MIT and built by Raytheon. It used a 16 bit word, 14 bits of data with a sign bit and a parity bit, and it held 36K words of fixed memory and 2K words of erasable memory. One flew in the command module and an identical one in the lunar module.

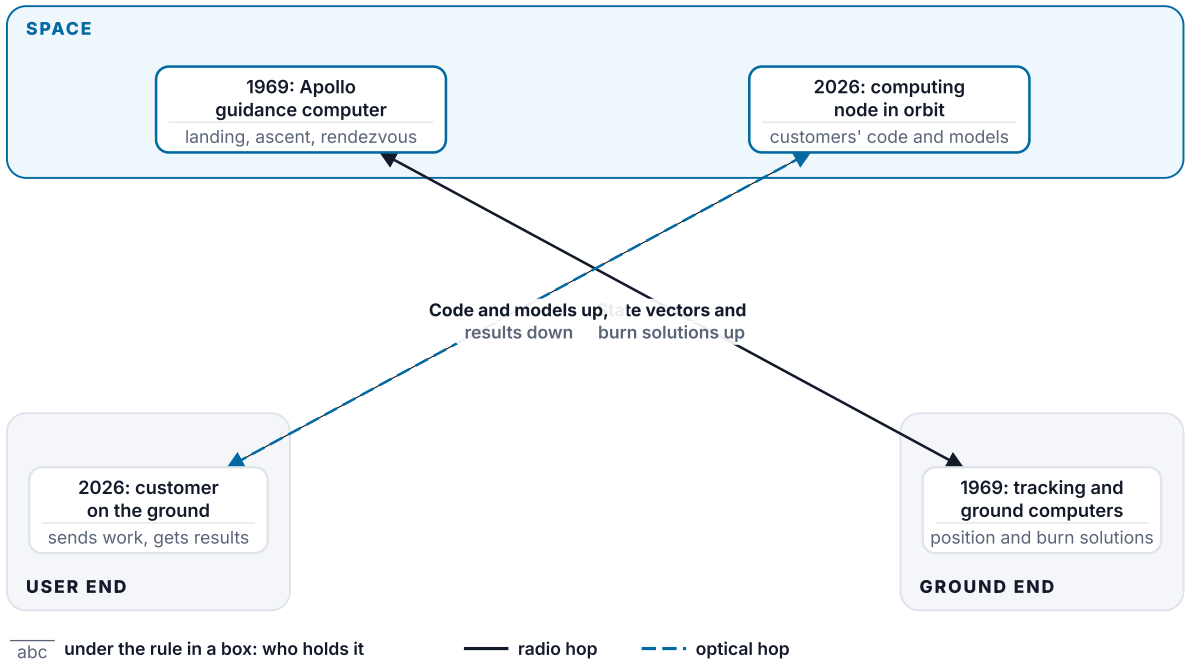
It did not do all of Apollo's computing, and it was not meant to. NASA's history records that "the ground computers became the prime determiners of the vehicle's position in three-dimensional space". Tracking stations measured the spacecraft, computers on the ground worked out where it was and what burn it needed, and the solution was sent up: the plan was to "calculate the lunar orbit insertion burn on the ground and then transmit the solution to the spacecraft computer". The command module's computer handled navigation between the Earth and the Moon with the ground as its backup, and "if contact with the ground were lost, the CM system had autonomous return capability".

Where the ground could not help, the computer on board did the work. The lunar module's computer provided "autonomous landing, ascent, and rendezvous guidance", because the landing "did not allow the ground to act as an effective backup". The same history gives the limits that put a computer on board at all: a signal delay it gives as 1.5 seconds from the

Earth to the Moon and back (light takes about 1.3 seconds each way at the Moon's mean distance, so about 2.6 seconds there and back, calculated), and a lunar orbit insertion burn on the far side of the Moon, "out of contact with the earth". It sums up the result plainly: the spacecraft "was only autonomous in the sense it could return safely to earth without help from the ground". Chapter 2 tells how that division of labor worked, and what happened when the landing computer was overloaded.

FIGURE P.1

### 1969 and 2026: where the computing was done




In 1969 the ground computed Apollo's position and burns and sent them up, and the computer on board flew the landing; in 2026 a node in orbit is offered, by its operator's account, to run what a customer on the ground sends up.

*Author's diagram based on NASA, Computers in Spaceflight, chapter 2 (pages 28 to 31), and Axiom Space, Orbital Data Centers (read 2026-11-22).*

## 2026: computing sold in orbit

Axiom Space states that its first two orbital data center nodes launched on 11 January 2026, with Kepler's first tranche of satellites, and that they support software and model updates over the air. TechCrunch (trade press) reported on 13 April 2026 that Kepler's cluster carried about 40 Nvidia Orin processors on 10 satellites, with 18 customers. Another company's satellite, Starcloud-1, launched in November 2025 with an Nvidia H100, a processor designed for data centers, as TechCrunch (trade press) reported on 30 March 2026.

What these systems have computed in orbit, for whom and how reliably is stated here only as far as a source states it, and for most of it the sources cited here do not. A launch is not a demonstration, an offer is not a service with a record, and an announcement is neither. Chapter 12 keeps those status fields apart for every system it names, with the date each was checked.

The division of labor has turned around in one respect. The Apollo computer received what the ground computed. A computing node of 2026 is offered, by its operator's account, to receive what a customer on the ground wants computed: code and models go up, results come down.

## What changed, and what did not

Three things changed. Transistors: one chip now holds far more logic than Apollo's whole computer, which was built from flat packs of two logic gates each. Launch: small satellites now share rides to orbit, as Phi-Sat-1 did in 2020, and some carry commercial computers chosen for speed over flight record. The link: crosslinks and optical links can now carry data between satellites and to the ground at rates the Apollo era could not.

Two things did not. Every watt a computer in orbit draws still has to come from the spacecraft's power system, and nearly all of it ends as heat that must be radiated away, with no air to carry it. And the particles are the same: the trapped belts, the solar storms and the cosmic rays that a computer of 1969 had to survive are still there for one of 2026, and a smaller transistor is not automatically a safer one. The chapters that follow walk the years between the two computers, one budget at a time.

CHAPTER ONE

# Where Should the Computing Happen?




The Apollo Guidance Computer of 1969, met in the prologue, held 36K words of fixed memory and 2K of erasable memory, according to NASA's history of its computers. The memory had grown from MIT's first estimate of 4K and 256 words, and the same history records that the lack of memory still "caused constant and considerable software development problems". The computer was sized before anyone had written down, in detail, what it had to compute. This chapter starts at the other end. Before any processor is chosen, it asks what is being computed, where it could be computed, and which budget each choice spends.

## Three places

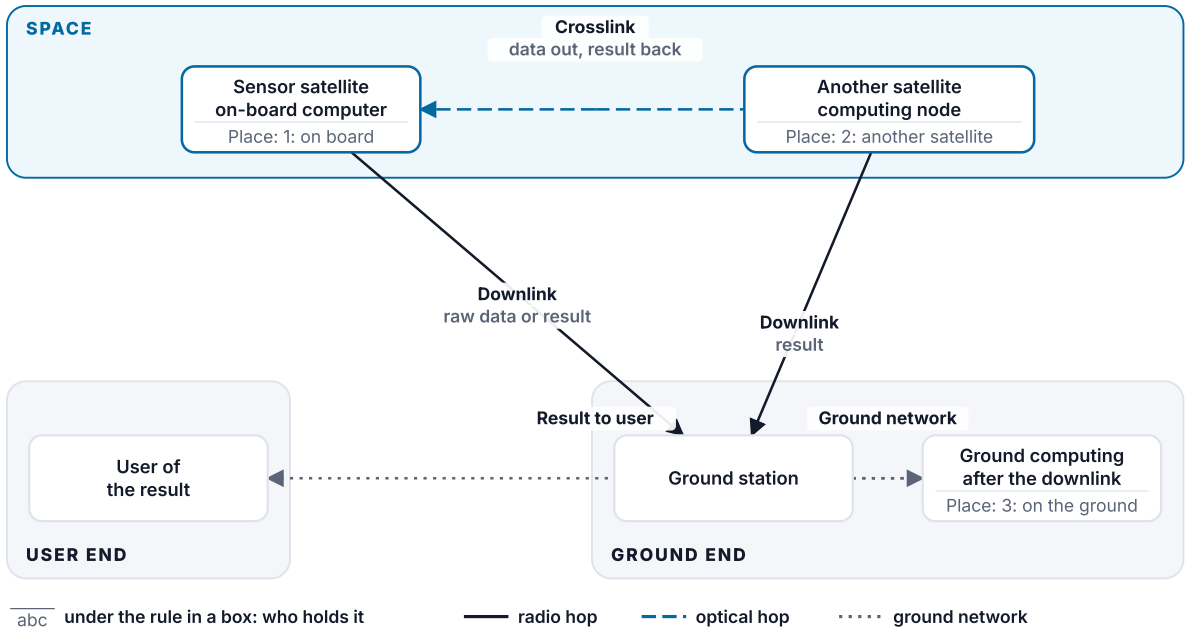
A satellite that produces data, or relays it, offers three places where a computation can run.

- **On the ground, after the downlink.** The data comes down as it was produced; computers on the ground do the work. Power, cooling and repair are cheap there. The cost is the link: every bit that is processed on the ground first has to be sent down, and it arrives when a ground station, or a relay, can see the satellite.
- **On board.** The satellite's own computer does the work, and often only the result comes down. The link carries less, and a result can be ready before the next contact. The cost is paid in the satellite's own budgets: energy, heat, radiation, mass, and, on most satellites, a computer that nobody will repair.
- **On another satellite.** The data crosses a link between satellites to a node that has more power or more computing, and the result goes down from there, or back. This adds a hop and a second satellite's budgets, and it moves the computing away from the sensor without bringing it to the ground.

The three are not exclusive; a workload is often split. The question is not "on board or on the ground?" for a whole satellite but "where should this computation run?", one computation at a time. Figure 1.1 draws the three places on one path.

FIGURE 1.1

## One computation, three places it can run




A sensor's data can be processed on board, on another satellite reached by a crosslink, or on the ground after the downlink; each place spends different budgets.

*Author's diagram based on the three places defined in this chapter; no real system is drawn.*

### Four kinds of computing

"The computer on the satellite" usually means one of four different things, with different owners, different risks and different histories. This book keeps them apart in every chapter.

- **The platform computer.** Command and data handling, and attitude and orbit control: it receives commands, runs the satellite, and reports its health. On many satellites it is split into a low-power computer that is always on and keeps the satellite safe, and higher-power computers that may be switched off when the satellite enters its safe mode. If it fails, the satellite may be lost.
- **Communication processing.** What a communication payload does to the signal beyond filtering, amplifying and shifting it: demodulating, decoding, routing and modulating again. Chapter 8 follows it.
- **Payload data processing.** What is done to the data a sensor produces: compressing it, screening it, extracting a result from it. Chapters 9 and 10 follow it.
- **Hosted general-purpose computing.** A computer in orbit that runs software its owner did not write, for customers who rent time on it. The second book of this series, *From Telstar to the Phone in Your Pocket*, gives its short history in its chapter 13; chapter

12 of this book sets out what is launched, demonstrated, offered and announced.

The platform's computing must keep working, or recover within a set time, through every eclipse and every credible fault, so it is designed for low power and for radiation tolerance that can be shown, by test or by flight record; a payload processor may run only when there is data and power to spare; a hosted computer is judged on work done. Figure 1.2 sets the four kinds against the four budgets of the next section.

FIGURE 1.2

### Four kinds of computing meet the four budgets differently

| Kind | What it does | Energy | Heat | Radiation | The link |
|---|---|---|---|---|---|
| Platform computer | Commands, data handling, attitude and orbit control; keeps the satellite safe | Small but continuous, through every eclipse and safe mode | Small; must stay in range at every temperature the satellite sees | Must survive the whole mission; a fault can lose the satellite | Commands up, health reports down; small volumes, high importance |
| Communication processing | Demodulates, decodes, routes and modulates again | Grows with traffic carried | Grows with traffic carried | An upset can corrupt traffic or routing | Is the link: changes what each hop must carry (chapter 8) |
| Payload data processing | Compresses, screens or extracts results from sensor data | Can be scheduled when power is available | Can be scheduled, within the radiator's capacity | An upset can silently change a result | Reduces what goes down; may add model or software uploads |
| Hosted general-purpose computing | Runs customers' software on a shared computer | Judged by work done, so it tends to draw more | Grows with the power drawn; the radiator area grows with it | Faster commercial parts, watched and restarted | Input and result may both cross the link; code goes up |

Illustrative tendencies, not definitions: a qualitative summary of the frame used in this book; later chapters give dated figures for named systems.

*Author's summary of the frame of this chapter; the four kinds follow the plan of this book and the second book of this series, chapter 13.*

## Four budgets, with their units

A computer in orbit lives inside four budgets that a computer on the ground rarely meets together. Each has a unit, and a trade that leaves the unit out is not yet a trade.

- **Energy**: power in watts while running, energy in watt-hours per orbit or per day. Every watt comes from the spacecraft's power system: on most satellites solar arrays and batteries, which carry the load through each eclipse; on some deep space probes radioisotope generators, as on Voyager. A computation competes for that energy with the radio, the instrument, the heaters and the attitude control.
- **Heat**, in watts to be rejected, square meters of radiator and allowed temperatures. Most of the electrical power a computer draws ends as heat inside the spacecraft, and that heat leaves by radiation from the spacecraft's outer surfaces to space, after it has been conducted to them from the part.
- **Radiation**, in dose (krad in silicon) accumulated over the mission, and in single-event rates (upsets per bit per day, functional interrupts, the chance of a destructive event). Chapter 3 defines each effect and how it is tested.
- **The link**, in bits per contact, contact minutes per day, the delay until a result reaches its user, and what must go up as well as come down: commands, software and, for a learned model, its weights.

Cost, schedule and the proof a mission asks for sit on top of the four, as the price of moving a line in one of them.

### Why the same processor behaves differently in orbit

A processor that runs well on a desk can fail in orbit for four reasons that have nothing to do with its speed.

- **No air outside.** There is no convection from the spacecraft to space. Inside the spacecraft, heat travels by conduction through the board, its mounting and the structure to a radiator; inside a pressurized module, such as on a space station, air can also carry it. A part cooled by a fan on the ground needs a conduction path in orbit.
- **Eclipse.** A satellite in low orbit passes through the Earth's shadow on many of its orbits. For a circular orbit at 500 km the period is about 94.5 minutes, and the longest shadow, with the Sun in the orbit's plane, lasts about 36 minutes, a little over a third of the orbit, by the cylindrical shadow approximation. In shadow the arrays give nothing and the battery carries the load, and the thermal load on every unit changes with each passage.
- **Particles.** Trapped protons and electrons, solar particles and cosmic rays deposit charge in the silicon. Some damage builds up slowly; a single particle can flip a bit, or start a short that destroys the part unless the power is removed in time. Low orbit is not free of this (chapter 3).

- **No repair.** A free-flying satellite that nobody services keeps the computer it was launched with. Its software may be changed from the ground, if the upload path was designed for it, and then the upload path becomes part of the computer.

## The trade sheet, line by line

The tool this book uses for "on board or on the ground?" is a sheet of fifteen lines, filled once for the on-board case and once for the ground case. It is a first-order tool: it shows which line decides, and where a full system model is needed before anyone builds. The blank sheet is in Appendix A; chapter 10 fills it twice and chapter 11 four times. When the third place, another satellite, is a candidate, the sheet is filled a third time, with the crosslink's capacity and availability, the remote node's queue and energy, its operator and trust boundary, and the outages the two satellites can share.

The example below is invented to show the arithmetic. It is not a real satellite, and its numbers are assumptions chosen to be easy to follow. A small Earth observation satellite in a 500 km circular orbit produces 400 Gbit of images a day. It has four ground contacts a day of 8 minutes each at a net payload data rate of 100 Mbit/s, after coding and protocol overhead. A customer wants maps of open water after floods, quickly. The on-board case screens every image with a 10 watt processor and sends only the images that show the region of interest without cloud, plus a small water map.

**The trade sheet, filled for an illustrative flood-mapping workload (inputs invented for the example; results calculated from them)**

| Line | What it asks | The illustrative example, on-board case |
|---|---|---|
| 1 Objective and payer | Mission return, delay, lifecycle cost or margin; for whom | Useful flood images delivered per day, and the delay to the first water map; for the operator's customer |
| 2 Workload and result quality | What is computed, on what input, how often; how good the on-board result is against the ground result | Screen every image for cloud and for the region; build a water map. The screen must not discard useful images, and the cloudy images it keeps spend downlink: its false discard rate and its false acceptance rate are the quality lines |

| Line | What it asks | The illustrative example, on-board case |
|---|---|---|
| 3 Data | Produced per orbit; fraction worth sending; size of the result | 400 Gbit a day produced; 25 percent worth sending (100 Gbit); water maps about 1 Gbit a day |
| 4 Link | Contact time, rate, delay, radio energy, added uplink | 4 contacts of 8 minutes at 100 Mbit/s: 192 Gbit a day. Without screening, 48 percent of the day's data fits. With screening, 101 Gbit needs about 17 minutes of the 32. Uplink: a new screen or model when it changes |
| 5 Energy | Power in sunlight and eclipse; duty cycle; share of the budget | 10 W for 20 minutes an orbit, about 15 orbits a day: 50 Wh a day, run in sunlight. Screening 400 Gbit in those 18,000 seconds needs about 22 Mbit/s of processing throughput, to be shown by a benchmark. A 30 W transmitter switched off for the contact time no longer needed, about 15.2 minutes (32 less 16.8), saves about 7.6 Wh. Net cost about 42 Wh a day, about 3 percent of an assumed daily energy budget of 1,440 Wh (a 60 W orbit average) |

| Line | What it asks | The illustrative example, on-board case |
|---|---|---|
| 6 Heat | The balance, the conduction path, allowed temperatures, radiator area | 10 W more while running. A surface of emissivity 0.85 at 20 degrees C emits at most about 355 W per square meter, so at least about 0.03 square meters of radiator, an ideal lower bound and not a radiator design, before absorbed sunlight, albedo and Earth infrared reduce the net. Sunlit steady state; eclipse transients need a thermal model |
| 7 Radiation | Effects, rates and mitigation, from an environment run and test data | Not computed here: no environment run or device test is cited. Qualitative: a screen that falls back to keeping every image discards nothing, but the contacts still carry only 192 Gbit a day, so the backlog grows by about 208 Gbit a day (400 less 192) until the store fills; an upset that changes a decision silently is the risk to state |
| 8 Mass and volume | Added to the satellite; launch cost from a dated register | One board and its harness; no launch price is used in this example |
| 9 Updates and trust | Who changes the software or model; how an upload is authenticated; how it is rolled back | The operator uploads the screen; the sheet asks for the authentication and the way back to the previous version |
| 10 Lifecycle cost | Engineering, qualification, operations, replenishment | Not estimated in this example |

| Line | What it asks | The illustrative example, on-board case |
|---|---|---|
| 11 The ground alternative | The same job on the ground; when it is infeasible | Sending all 400 Gbit needs about 67 minutes of contact a day, about twice the four contacts. If the ground case cannot choose which images to send, the 192 Gbit it can send hold about 48 Gbit of useful images at the same 25 percent, and the other 208 Gbit wait in the store or are overwritten; if the satellite sends small previews first and the ground picks, more. The first water map waits for the downlink of the images that contain it |
| 12 Uncertainty | The range on each line and its evidence class | The 25 percent is the weakest input: it depends on season, region and cloud. Every input here is an assumption, and every result is calculated from them |
| 13 Sensitivity | The one or two lines that change the verdict | The fraction worth sending, and the screen's false discard and false acceptance rates |
| 14 Assumptions and limits | What the sheet does not model; when a system model is needed | Battery depth of discharge, thermal transients, scheduling against other loads and the cost of the extra contacts are not modeled |

| Line | What it asks | The illustrative example, on-board case |
|---|---|---|
| 15 Verdict | Advantageous for whom, against which objective, if which assumption holds | Advantageous for the customer, against useful images delivered per day, if the ground case cannot select images without screening them, the screen keeps useful images at a stated false discard rate, and its false acceptance rate is low enough that what it keeps still fits the 192 Gbit a day; against delay, only if the screen runs before the next contact and the flood images are sent first, which this sheet does not model. Not advantageous against the spacecraft's electrical energy alone: it spends about 42 Wh a day more than it saves |

Three things in the example hold beyond it. First, on-board processing did not save energy here: the radio time it saved was worth less than the processor's time. The case for it rested on the link and the delay, not on power. Second, line 2 decides whether the whole sheet holds: a screen that discards useful images saves downlink by losing the mission's product. Third, the verdict has a reader and an objective. For an archive that wants every raw image, the same sheet ends "not advantageous".

Here every input is an assumption and every result is calculated from them, which is why the sheet is labeled illustrative. A sheet built for a decision replaces each assumption with an operator's figure, a test report, a standard or a measurement by a named party, and says which.

## Where the Link Ledger fits, and where it does not

The series tool, the Link Ledger, describes a link: page 1 the service in ten fields, page 2 the hop budget in decibels. The first book of this series, *The Bit's Path*, explains it with the radio theory behind it. A computer is not a link, so this book uses the ledger only where computing changes one.

- **Where it fits.** When processing changes what a link must carry, page 1 holds the argument in five of its fields: the requirement (1), the ends and the path (2), the capacity (5), trust (7), and how it fails and how anyone would see it (8). The data volume (produced per orbit, stored, sent per contact, removed by processing) is worked in a chapter table rather than on a ledger page 2, which this book does not print (figure 9.1; the blank form is in Appendix A). Chapters 2, 9 and 12 work service ledgers. No filled hop budget in decibels is printed in this book; Appendix A keeps the series' blank form.
- **Where it does not.** The platform computer keeps the satellite alive and is not a link. The radiation environment is a place, and the choice of a part is a purchase. Chapters 2 to 7 use tables for these, and line 7 of the sheet carries what they find.

## What would make the frame wrong

The sheet assumes that a computation goes on board because a budget says it should. That is not always why it goes. A processor may fly to give a part a flight record, to show investors or an agency that something can be done, or because a program needs a demonstration before it can sell a service. These are legitimate reasons, and the sheet cannot weigh them. When a chapter meets such a case, it names the reason its source gives and does not force it into the four budgets.

Military needs, a regulator's rules on where data is processed, and plain demand also lie outside the frame. The book says so where it meets them, and keeps three verdicts apart: whether a link or a computation works, whether a service delivers what it promises, and whether a business recovers what it spent.

> **FAILURE: MEMORY THAT RAN SHORT ON THE APOLLO COMPUTER**
>
> This chapter sets out a frame, so its case is about the frame's first line. NASA's history of the Apollo computer records that MIT's original design called for 4K words of fixed memory and 256 words of erasable memory, that both grew, to 36K and 2K, and that the shortage still caused constant software problems. It gives a reason: MIT received a contract based on "only a short, very general requirements statement", and "the requirements started changing immediately". The 16 bit word, chosen when the memory was small, then made addressing the larger memory harder for the programmers. The computer flew and did its job; what ran short was the memory, and the history ties that to requirements that were general at the start and kept changing. What follows from it is modest: size the computing from the workload, and expect the workload to grow.

**SECURITY NOTE**

**Who can interfere or command.** Five questions recur in every chapter of this book. Who is authorized to command the computer and to change its software, its configuration or its model? Who can read what it holds or sends? Who can interfere, through the command link, through code hosted on it, or through the supply chain of a part or a configuration file? How is a command or an upload authenticated? How is the system recovered? There are three places to attack a computer in orbit: the ground segment that sends the commands, the command and upload path, and the software on board. Moving a computation on board moves it to the third, out of reach of hands. **How it is seen.** Detection, diagnosis, attribution and recovery are four separate steps. A satellite can detect that its state is wrong long before anyone can say whether a particle, a software fault or an intruder caused it. **How it is recovered.** By a way back to known software that a bad upload cannot remove, and a command path that does not depend on the software being recovered. Each later chapter asks the five questions of its own computer.

**Monday rule:** before asking how fast the on-board computer is, write which of the four kinds of computing it does and which budget runs out first.

CHAPTER TWO

# 1962 to 1996: Computers That Had to Fly, Line by Line



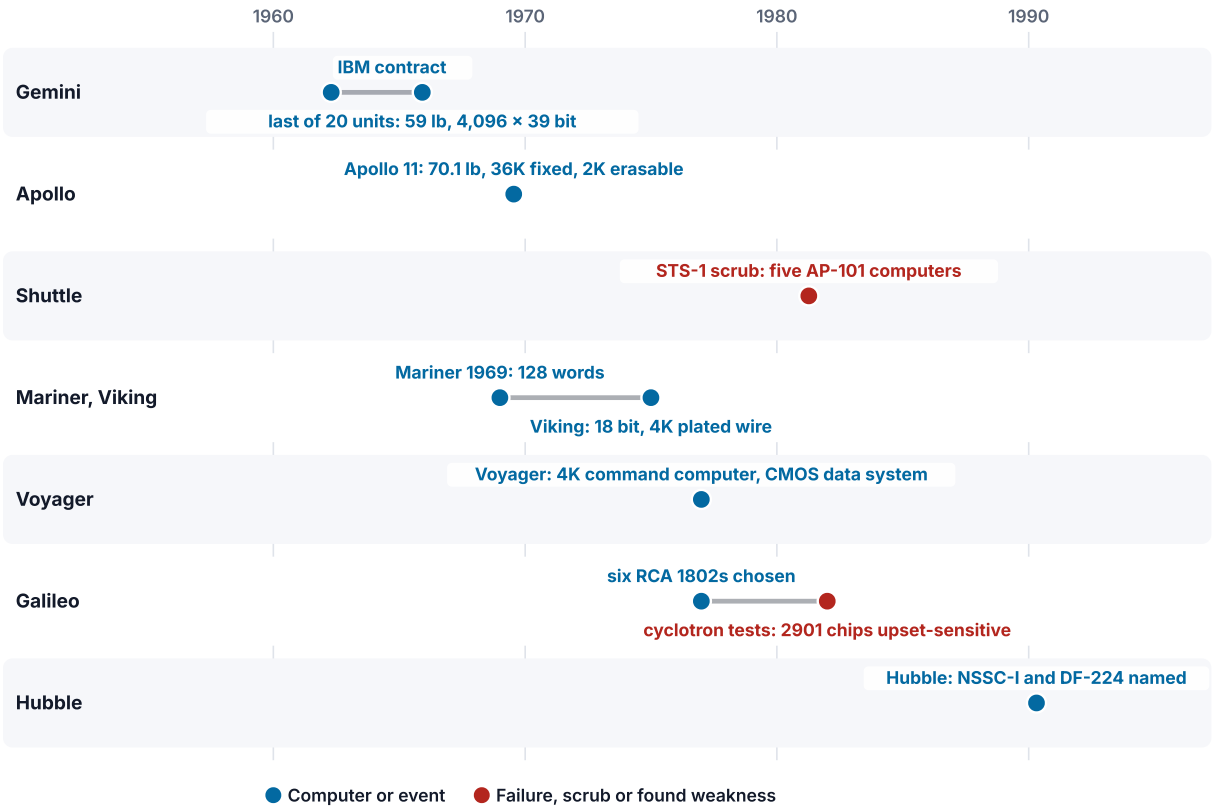

At 102 hours, 38 minutes and 26 seconds into the Apollo 11 mission, on 20 July 1969, Neil Armstrong reported a program alarm from the lunar module's computer: "It's a 1202." The mission report, as quoted in NASA's Apollo Lunar Surface Journal, logs the alarm at 102:39:02, and a 1201 followed at 102:41:32. The computer was restarting itself while it flew the descent. It dropped the jobs it did not need, kept the ones that steered the engine, and the landing went on. That recovery had been designed years earlier, by people who had to decide which computing could not be left to the ground. This chapter follows that decision and asks one question: when a flight computer weighed tens of pounds and held a few thousand words, what was done on board, what was done on the ground, and how did a program come to trust a computer with a crew or a long mission?

Its cases run from 1962 to 1990. Most come from NASA's own history of its flight computers, published in 1988; Hubble, launched in 1990, comes from a 2015 NASA Goddard review. The sources read give no case from 1991 to 1996, and chapter 5 takes up the 1990s.

The record behind this chapter is uneven, and it is better to say so at the start. NASA's official history, *Computers in Spaceflight: The NASA Experience* (1988), covers the United States line in detail, and most numbers below come from it. The Soviet, European and Japanese computers of the same period were sought in builder and agency sources and could not be read on the date this chapter was written; the section on the other lines says what was sought, and Appendix B lists every case with the date. A short section is a gap in what could be read, not a judgment on what was flown. Figure 2.1 places the dated points that could be sourced.

FIGURE 2.1

## Flight computers of the United States line, 1962 to 1996, at the points NASA's history dates




Each point sits at the date its source gives (a year alone is placed at 1 January); mass and memory are given where the source gives them; the other national lines are not drawn because no primary record of them was read.

*Source: NASA, Computers in Spaceflight: The NASA Experience (1988), chapters 1, 2, 4, 5 and 6; NASA History, STS-1 first launch attempt (2021); D. McComas, NASA Goddard, Lessons from 30 Years of Flight Software (2015).*

## Before the general-purpose computer

Mercury, which NASA's history calls "America's first man-in-space effort", carried no digital computer on board. In the same history, Mercury's re-entry "was calculated by a realtime computing center on the ground, with retrofire times and firing attitude transmitted to the spacecraft while in flight", so the capsule did not need one. The trajectory arithmetic was done on the ground, and the spacecraft received the answer.

The early robotic probes carried sequencers instead: timers and fixed logic that issued commands in a set order. The step that matters for this book came with Mariner Mars 1969, which carried a 26 pound "programmable Central Computer and Sequencer", designed at the Jet Propulsion Laboratory and built by Motorola. Its 128 words of memory "could be

altered in flight". The requirement had been 20 words; the memory was soon full, and the programmer who wrote most of it replaced the launch and mid-course routines with new code once they had run. The older fixed sequencer flew as a backup. During critical maneuvers the two ran in parallel, and a disagreement aborted the maneuver. Even so, the two spacecraft were the most commanded to that date, in the history's words: Mariner VI received 946 radio commands and Mariner VII 957.

Two habits that run through the rest of this book start here. A program can change the software after launch, so the ground keeps a hand on the computer. And a second, simpler device checks the first, so a fault does not act alone.

## United States: the crewed line

**Gemini.** Gemini needed rendezvous, a backup for ascent guidance and more accurate re-entry, and these "dictated that the spacecraft carry some sort of active, on-board computing capability", in NASA's words. IBM received the contract on 19 April 1962, for 26.6 million dollars, and delivered the last of 20 machines by December 1965. The computer weighed 58.98 pounds, was built from discrete components rather than integrated circuits, and had a core memory of 4,096 addresses of 39 bits, each word split into three 13 bit syllables. It worked in six mission phases: prelaunch, ascent backup, insertion, catch-up, rendezvous and re-entry.

Gemini also flew a 26 pound Auxiliary Tape Memory of 1,170,000 bits, which raised the storage seven and a half times. Tape then made about one error in 100,000 bits; IBM recorded each program three times and passed each set of three bits through a voting circuit on its way into core. On Gemini VIII the spacecraft began to spin after docking with its Agena target, and the re-entry program was loaded from the tape while it was still spinning, beyond the tape's vibration specification. The history records one in-flight failure of a computer part: on the 48th revolution of Gemini IV the computer would not turn off and could not be used for the planned re-entry; IBM could not reproduce the fault, and a bypass switch was fitted for Gemini V. The computer's designers listed it among their firsts as "the first digital computer on a manned spacecraft"; an IBM engineer called it the "last of a dying breed".

**Apollo.** NASA's history gives three early reasons for a computer on board, to avoid jamming, to prepare for planetary missions and to avoid saturating the ground stations, and says none of them became the main one. The main reason was "the 1.5-second time delay in a signal path from the earth to the moon and back". The history's figure is low: at the Moon's mean distance of about 384,400 km, light takes about 1.3 s each way, about 2.6 s there and back (calculated). Either way the delay ruled out steering a landing from Earth, and the choice of lunar orbit rendezvous, which put the insertion burn on the far side of the Moon,

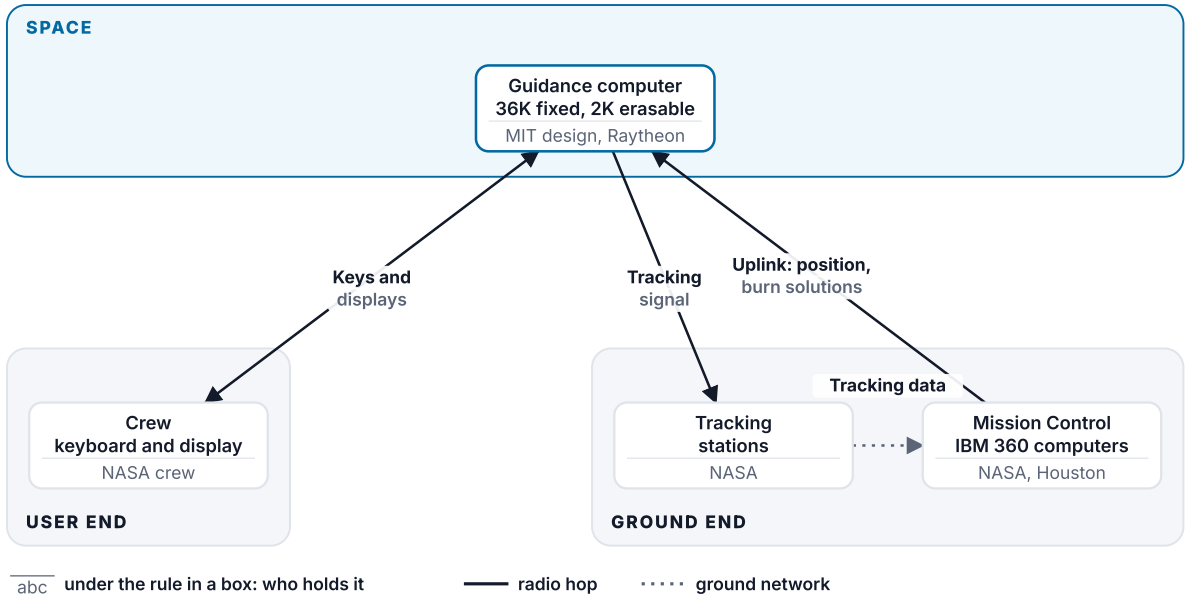
out of contact. The Apollo Guidance Computer was designed by MIT's Instrumentation Laboratory and built by Raytheon. In its Block II form it used a 16 bit word (14 data bits, a sign and a parity bit), held 36K words of fixed memory and 2K words of erasable memory, weighed 70.1 pounds, drew 70 watts at 28 volts, and ran from what the history calls a one megahertz clock (in MIT's 1972 final report, a 2.048 MHz oscillator divided to a 1.024 MHz reference). Its logic was built from flat packs that each held two logic gates. The same machine flew in the command module and the lunar module.

The fixed memory was core rope, woven at the factory. In erasable memory each core stored one bit; in core rope, NASA's history explains, up to 64 wires ran to each core, a wire passing through the core read as a one and a wire bypassing it as a zero, so that one core held four 16 bit words. For the Block II rope that flew, MIT's final report gives six modules, each of 512 cores and 192 sense lines, so that one core carried 192 bits, twelve 16 bit words; the report puts the growth to 36,864 words down to more sense lines in each module. That made it robust, and it also meant that software had to be delivered months before a mission so the rope could be made and tested. A late change in fixed memory meant a new rope. The 2K erasable words held every variable, and one programmer of the lunar module software recalled that some locations were "shared seven ways" between programs at different times.

The division of labor then moved back toward the ground. "The ground computers became the prime determiners of the vehicle's position in three-dimensional space 'at all times' (except during maneuvers)," the history says, and planners "even decided to calculate the lunar orbit insertion burn on the ground and then transmit the solution to the spacecraft computer". In the end, in the same source, the spacecraft was "only autonomous in the sense it could return safely to earth without help from the ground". Figure 2.2 draws that split, and the ledger at the end of this chapter records the link that carried it.

FIGURE 2.2

## Apollo: the ground knew where the spacecraft was, the computer flew it




Tracking data were worked by Mission Control's computers, which sent position and burn solutions up to the guidance computer; the computer and the crew acted on them, and behind the Moon the computer worked alone.

*Author's diagram based on NASA, Computers in Spaceflight, chapters 2 and 8, and the Apollo Lunar Surface Journal (Apollo 11).*

The computer was built to recover from its own software failures, because a real-time system cannot simply stop. A restart could be triggered by a voltage failure, a clock failure, the processor stuck in interrupt mode, or a program called NIGHT WATCHMAN that noticed when the operating system had stopped scheduling work. A restart jumped to a fixed address, read tables that the software kept up to date as it ran, and restarted each job from its last recorded point. It also cleared every output channel, "such as control jet commands, warning lights, and engine on and off commands", so that nothing outside the computer's control would happen during the restart. That design is what the Apollo 11 crew relied on in the case at the end of this chapter.

**The Shuttle.** The Shuttle carried five general-purpose computers, each an IBM AP-101 with a custom input and output processor, built from transistor-transistor logic integrated circuits. The history explains the count. The original requirement, fail operational, fail operational, fail safe, needed five; NASA lowered it to fail operational, fail safe, which needed four, and since five were already designed in, the fifth became the Backup Flight System, with enough functions for ascent, selected aborts and descent in one memory load.

"In the interest of avoiding a generic software failure, NASA kept its development separate" from the primary software; Rockwell won that contract, under a different NASA directorate. The four primary computers ran as a redundant set, kept in step at defined synchronization points, and the voting was done at the actuators: each was quadruple redundant, so the commands of one failed computer were "physically out-voted" by the other three. The fifth was kept "as insurance against a generic software error in the primary system". Not everyone agreed it helped: John R. Garman of the Johnson Space Center said "we probably did more damage to the system as a whole by putting in the backup", because it took pressure off the primary software's developers, and joining its time-slice operating system to the primary set's priority-driven one forced compromises.

The first launch attempt shows a redundancy that blocked a launch. On 10 April 1981, in NASA's account, controllers switched Columbia's computers to launch mode and the backup computer "couldn't communicate with two of the primary units". The countdown held at T minus 9 minutes and the launch was scrubbed. "It took engineers one day to determine that the primary computers had a 40-millisecond timing problem", fixed by a software update, and the second attempt followed on 12 April. The backup existed to guard against a common software fault in the primary set; here the joining of the two sets was itself where the timing fault showed. The mechanism is told in detail in Garman's own paper of October 1981, which this book did not read; the account above stays with NASA's.

## United States: the robotic line

**Viking.** The two Viking spacecraft, launched in 1975, each carried an orbiter and a lander with "dual redundant computer systems". The orbiter's Command Computer Subsystem was a serial machine with 18 bit words and 4K words of plated-wire memory, with an average instruction time of 88 microseconds. The lander's Guidance, Control and Sequencing Computer used two Honeywell HDC 402 processors, each with 18K words of plated-wire memory. NASA's history calls the orbiter's redundancy "single fault tolerance": each component had a backup.

**Voyager.** Voyager, launched in 1977, divided its computing among three computer types. The command computer was nearly identical to Viking's, with the same 4K of 18 bit plated-wire memory. The attitude and articulation control computer was a separate machine. The Flight Data System was a new design built from CMOS integrated circuits with volatile CMOS memory, chosen because the faster circuits would have been wasted on slow plated wire. The command computer watched the others. The attitude computer sent it a "heartbeat" code about every two seconds and ran self-tests in between; after about 10 seconds without a heartbeat, the command computer switched to the backup processor.

That switch happened on Voyager 2 sixteen seconds after separation from its solid rocket stage, when a rough separation knocked the spacecraft off attitude, and the backup, with no record of the gyro readings, executed a simple orientation and stopped the roll.

Voyager also shows the other half of the bargain: a computer of fixed size kept busy by new software. In-flight reprogramming was "a nearly routine task" by its launch; no fewer than 18 loads were sent to Voyager 1 during its Jupiter encounter, and in long cruise new loads came every three months. After Voyager 2's primary radio receiver developed an internal short, leaving no redundant receiver, a "disaster backup sequence" that would run minimum experiments and send data to Earth was stored in its command computer for the Uranus and Neptune encounters, taking 20 percent of the 4K memory. Voyager carried no RCA 1802: NASA's history of its computers describes custom CMOS logic in the Flight Data System and does not mention the 1802 there.

**Galileo.** The RCA 1802 flew on Galileo. NASA chose it, around 1977, for Galileo's Command and Data Subsystem, which used six 1802 processors with 176K of memory (the history does not give the unit). The choice was "nobody's favorite", in the history's words, but the 1802 was a CMOS device with low power, about 30 milliwatts, and wide voltage tolerance; it was also slow, with a 5 microsecond cycle and about two cycles per instruction. Counting the instruments, the attitude computers and the probe, the design that the 1988 history describes had 19 microprocessors with about 320K of random access memory and 41K of read-only memory. Its attitude computer was built from 2900 series bit-slice parts.

Galileo is also where radiation became a design driver for the computer, late. Its parts were chosen in 1977 on the basis of Pioneer data, before Voyager reached Jupiter, and, in the history's words about the project's choice, "nothing was known about SEU vulnerability" of those parts, so no error correction and no special shielding were planned. A single event upset (SEU) is a bit changed by one energetic particle; chapter 3 defines it fully. By 1980 to 1981 the problem was clear, and in 1982 the project arranged cyclotron tests at Berkeley. The 2901 bit-slice chips of the attitude computer were "highly SEU sensitive, with 20% to 50% of hits causing probable software failures", while the older, less dense 1802s were much less sensitive. The project weighed two fixes: a radiation-hardened processor that emulated the 2901 but needed a new compiler, for 20 million dollars, or custom radiation-hardened 2901s from Sandia National Laboratories, so that no software had to change, for 5 million. It chose Sandia. The history adds that if Galileo had launched on its 1982 schedule, the problem would not yet have been understood. The launch slipped to 1986 and was then postponed indefinitely when the Shuttle fleet was grounded; chapter 9 follows the flight.

**Hubble.** One line from a NASA Goddard engineer's 2015 review of 30 years of flight software: the Hubble Space Telescope, launched on 24 April 1990, carried two computers that the review names, the NASA Standard Spacecraft Computer, NSSC-I, and a DF-224

"with 32K 24-bit words available at one time"; the review's word for the DF-224 is "co-processor". It does not say how the work was divided between the two machines, and this book does not infer it.

FIGURE 2.3

## Where the period put the dependability: in the part, the architecture or the software

| System | Where | How | What it cost or left open | Source |
|---|---|---|---|---|
| Mariner Mars 1969 | Architecture | The old fixed sequencer ran beside the new computer in critical maneuvers; a disagreement aborted the maneuver | A second device carried | CiS ch. 5 |
| Gemini tape memory | Architecture | Each program recorded three times; each set of three bits voted on its way into core | Three times the tape for one program | CiS ch. 1 |
| Viking orbiter | Architecture | Dual redundant computers: a backup for each component | | CiS ch. 5 |
| Apollo guidance computer | Software | Restart from recorded points, outputs cleared first | Every program had to keep its restart tables current | CiS ch. 2 |
| Voyager | Architecture and software | Heartbeat about every 2 s; switch to the backup after about 10 s of silence | Voyager 2's backup took over with no gyro history | CiS ch. 6 |
| Shuttle | Architecture | Four synchronized computers, voting at the actuators; a fifth with separately developed software | Synchronization the hardest task; the STS-1 scrub | CiS ch. 4; NASA 2021 |
| Galileo attitude computer | The part | Radiation-hardened 2901 chips made by Sandia, so no software changed | 5 million dollars, against 20 million for a hardened emulator | CiS ch. 6 |

Seven cases from NASA's history, each with what the choice cost or left open where the source says so; an empty cost means the source does not say.

*Source: NASA, Computers in Spaceflight: The NASA Experience (CiS, 1988); NASA History, "40 Years Ago: The Countdown Begins for STS-1; First Launch Attempt Scrubbed" (2021).*

## The other lines, and what this book could read

Several cases were sought for this chapter. For the Soviet Union and Russia: the on-board computers of the Zond lunar spacecraft, of the Salyut and Mir stations, and of the Buran orbiter (the Biser-4). For Europe: the computers of Spacelab and of the first science missions of the European Space Agency. For Japan: its first on-board computers and the line that leads to the HR5000 processor. Each was sought in a builder, institute or agency source. On 23 November 2026 none of those sources could be read, and no primary record of them was among the sources available. No date, word size, memory or mass is printed for them here, because none was read.

Two things can be said from sources that were read. First, the later lines are not missing: chapter 5 places European, Japanese and Indian processors from 1990 onward on their builders' own pages and in peer-reviewed papers. Second, the United States record is itself an official history written by the agency that flew the computers, and it is labeled as such wherever it is used. Appendix B gives each case sought, the source that would settle it, and the date it was sought.

## What the period fixed

NASA's 1988 history shows five practices that later flight computers inherited.

- **Reprogramming from the ground.** From Mariner Mars 1969's 128 alterable words to Voyager's 18 loads at Jupiter, the software was expected to change after launch. A fixed memory could serve more functions over a mission than it held at launch, because programs for finished phases were replaced by new ones.
- **The watchdog.** Apollo's NIGHT WATCHMAN and Voyager's heartbeat are the same idea in two forms: a simple check that the main work is still being scheduled, and a defined action when it is not.
- **Restart with protected state.** Apollo restarted jobs from recorded points and cleared its outputs first; Voyager kept a stored sequence that would run with no ground contact. Both decide in advance what the computer must do alone.
- **Redundancy with a different design.** Mariner ran an old sequencer beside the new computer; the Shuttle flew a fifth computer with separately written software. The STS-1 scrub shows that the join between the two is also a design, with its own faults.
- **Flying a part chosen long before launch.** Galileo's processors were chosen in 1977 for a launch planned for 1982 that slipped by years. The parts were then tested against an environment that had not been known when they were chosen. Chapter 5 asks whether that lag is still the right answer.

### THE SYSTEM'S FAILURE, AND THE RECOVERY: APOLLO 11, 20 JULY 1969

**What failed.** NASA's history calls it "a software failure causing restarts"; what ran out was the computer's time. In the design, hardware counter increment requests were served ahead of instructions; during the descent the rendezvous radar, through a switch position and "an arcane series of circuitry", made so many requests that about 15 percent of the computer's resources went to answering them. Scheduled jobs could not finish before they were due again, and the computer restarted, "three of which happened in a 40-second period" while the descent program ran. A programmer of the lunar module software later explained the codes: 1202 meant the executive had no free core set for a new job, 1201 no free vector area. **How it was recovered.** Each restart dropped the jobs the computer did not need and brought back the steering and the display, which the design protected; the radar's counter requests themselves continued, and kept taking their share of the computer's time. In Mission Control, Steven G. Bales and John R. Garman recognized the alarms as ones the computer recovered from and judged that guidance was still sound, and Bales called the computer GO for landing; the radar interface behind them was traced after the flight. **What the case shows, and no more.** Recovery from an unknown error was designed and tested before flight, and that design, not a spare computer, carried the landing. NASA's history gives the solution as correcting a switch position on the radar; the mission report's engineering account was not read. The history draws the lesson that built-in software recovery became part of NASA's crewed flight software from then on.

### SECURITY NOTE

**Who can interfere or command.** In the cases of this chapter, the program's own ground stations commanded the machine: Mission Control sent words to Apollo's erasable memory through the uplink (its fixed rope could not be changed from Earth), and every memory load to Voyager's other computers passed through its command computer. The Apollo crew could key values in and saw an UPLINK ACTY light whenever data arrived from the ground; on Apollo 14, a software patch to ignore a faulty abort switch was radioed up and entered by the crew. How the uplinks were protected against a false sender is not recorded in the history, and this book does not guess. **How it is seen.** Apollo's computer sent data words down to drive displays in Mission Control, so the ground saw selected values from the computer, which is observation, not proof that the values were right; the restarts of Apollo 11 appeared both in the spacecraft and in Mission Control. **How it is recovered.** By restart from protected state on board, by a switch to a backup processor, or by a reload of memory from the ground. On Apollo 11 these were separate steps by separate parties: the computer detected the overload and recovered by restart; two controllers judged that guidance was sound and called GO; the crew flew on; the cause, the radar interface, was found after the flight.

## The link that carried the ground's answer

**LINK LEDGER · LL-1969-01 · SERVICE LEDGER**

1969-07-20

System: Apollo: position and burn solutions computed on the ground, sent up to the guidance computer (Apollo 11)

Retired | Class: service only

| | |
|---|---|
| **1 Requirement** | Keep the computer's knowledge of position and of the next burn good enough to steer, while the ground does most of the navigation: ground computers were the prime source of position, except during maneuvers. |
| **2 Ends and path** | Mission Control, Houston, with IBM 360 computers; the guidance computer in the command or lunar module, with the crew at its keyboard and display.<br>**Path:** Tracking data to Mission Control by ground lines; solutions computed there, sent up by radio, held in erasable memory. The crew could also key values in. |
| **3 Orbit and time** | Earth to Moon and lunar orbit; about 2.6 s there and back (calculated); no contact behind the Moon, where lunar orbit insertion took place.<br>**Time:** Updates while in view, for example after coming around the Moon; the insertion burn was computed on the ground and sent up. |
| **4 Band and permission** | **Uplink.** S band (named in the crew's debrief); frequencies not in the sources read; NASA's own network |
| **5 Capacity** | **Up.** Not in the sources read; 2K words of erasable memory held all variable data.<br>**Down.** Data words sent down to drive Mission Control's displays; rate not in the sources read. |
| **6 Who owns each end** | **Space segment.** NASA; computer designed by MIT, built by Raytheon<br>**Ground segment.** NASA Mission Control, Houston; IBM computers<br>**Customer relationship.** NASA and its crews<br>**The exit.** Not applicable |
| **7 Trust** | Mission Control sent the words; an UPLINK ACTY light showed the crew when data arrived. Uplink authentication: not in the sources read. |
| **8 How it fails, and how we would see it** | A wrong or late update leaves the computer steering from a stale position.<br>**How we would see it:** On the crew's display, and in Mission Control, which received selected data words; observation, not proof a value was right. |
| **9 Cost shape** | Not costed per update in the sources read. |
| **10 Evidence** | Ground the prime source of position; insertion burn computed on the ground published history NASA, Computers in Spaceflight, ch. 2, p. 29, 2026-11-23<br>About 2.6 s Earth to Moon and back calculated 2 × 384,400 km at 299,792 km/s, 2026-11-24<br>State vector update after coming around the Moon published history Apollo Lunar Surface Journal, Apollo 11 landing, 2026-11-23 |

**Service only, no hop budget:** Uplink frequencies, power and rates are not in the sources read; the Apollo network's technical reports would settle them.

**Monday rule:** ask what the satellite must do with no ground contact for a named period; that is the computing that cannot be moved.

CHAPTER THREE

# Radiation: Effects, Environment and Assurance




On 23 November 1962, after its 1,242nd orbit, Telstar 1 stopped answering commands. Its builders at Bell Telephone Laboratories had seen the command channel "act erratically" since the fourth week of November, and their telemetry showed that "the density of electrons of high energy was much higher than had been anticipated". Their working hypothesis, printed in the Bell System Technical Journal in July 1963, was radiation damage to transistors in the command decoder. Telstar is an early, well documented case of a satellite whose electronics met an environment they had not been designed for. This chapter changes one decision: no dose figure, no upset rate and no "radiation tolerant" label means anything until the orbit, the mission length, the phase of the solar cycle and the shielding are written beside it.

This chapter answers three questions in order. What does the space environment do to a chip? How much of it is there in a given orbit? And how does a mission show, before launch, that a part will survive it? Chapter 4 then takes each effect and the design answers to it. The radiation line of the trade sheet in chapter 1 (line 7) is filled from both chapters.

## Two kinds of damage

Radiation harms electronics in two different ways, and the difference decides everything that follows. NASA Goddard's 2017 tutorial on radiation effects, by Jonathan Pellish, divides them into cumulative effects and single-particle effects. ESA's Space Environment and Effects Section, writing in 2011, makes the same split: two effects "are cumulative", and single event effects are "most often observed". Spacecraft charging, on the surface and inside, is a further hazard to electronics with its own environment and design rules; this book does not cover it.

**Cumulative damage** builds up over the mission. It comes in two forms.

- **Total ionizing dose (TID)** is the energy that ionizing radiation deposits in a material, per unit mass. In Pellish's words it "results in cumulative parametric degradation that can lead to functional failure", and in space it is caused "mainly by protons and electrons". In metal oxide semiconductor devices the tutorial's examples are threshold voltage shifts and increased off-state leakage; in bipolar devices, excess base current.
- **Displacement damage** comes from the part of a particle's energy that knocks atoms out of their places in the crystal lattice, leaving what the tutorial calls "permanent electrically active defects"; its measure is the displacement damage dose (DDD). It too is cumulative and comes mainly from protons and electrons. Its examples in the tutorial are reduced gain and damage in light-emitting diodes and optical sensors, which matters for detectors and optical links more than for logic.

The rate at which a dose arrives can matter, not only its total. Pellish's tutorial describes enhanced low dose rate sensitivity (ELDRS), first observed in bipolar devices in the early 1990s: some parts degrade more at a given dose when the dose arrives slowly, as it does in orbit, than when it arrives fast, as it does in many ground tests. The opposite can also happen: damage can partly recover. Telstar's builders removed all voltages from the command decoders in December 1962, and "as had been predicted, this action allowed recovery of the transistors", for a time.

**Single event effects (SEEs)** are triggered by single particles and need no accumulated dose, although dose can shift how susceptible a part is. Pellish defines one as "a disturbance to the normal operation of a circuit caused by the passage of a single ion (proton or heavy ion) through or near a sensitive node in a circuit". One particle, one event. The tutorial sorts them into two groups, and the sorting is the most useful single fact in this chapter.

- **Non-destructive:** a single event upset (SEU) changes a stored bit; one particle can change several cells at once, a multiple-cell upset (MCU); when several of them fall in one protected word it is a multiple-bit upset (MBU), which a code that corrects one bit per word cannot repair, and how the memory interleaves its words decides which (chapter 4); a single event transient (SET) is a short voltage glitch in a circuit, logic or analog; in logic it becomes an upset if a storage element captures it; a single event functional interrupt (SEFI) puts a device into a state where it stops working until it is reset, reconfigured or power cycled.
- **Destructive:** single event latch-up (SEL), a parasitic structure that turns on and draws a high current, which ESA calls "a runaway short circuit phenomena that burns out the entire circuit" unless power is removed in time; whether a given latch-up destroys the part, leaves latent damage or none depends on the part and its supply, which only a test shows; single event burnout (SEB) and single event gate rupture (SEGR), which destroy power transistors and gate insulators.

NASA's State-of-the-Art of Small Spacecraft Technology, in its avionics chapter (dated 19 May 2026, page updated 21 October 2026), uses the same categories: total dose and displacement damage for the long term, and single events, with upsets as non-destructive and latch-up as destructive. Figure 3.1 puts each effect beside its plain meaning and its two classifications. Keep the classifications apart when reading any claim: a part rated for a large dose can still latch up, and a part that showed no latch-up in test can still upset.

FIGURE 3.1

## Seven effects of one particle and two of many: what each does, and whether it destroys

| Effect | Plain meaning | Kind | Destructive? |
|---|---|---|---|
| Total ionizing dose (TID) | Energy deposited by ionizing radiation builds up and shifts transistor behavior (thresholds, leakage, gain) until the part fails its limits | Cumula-tive | Can end in failure |
| Displacement damage (measured as DDD) | Atoms knocked out of the crystal lattice leave permanent defects; worst for detectors, light emitters and gain | Cumula-tive | Can end in failure |
| Single event upset (SEU) | One particle changes one stored bit | Single | No |
| Multiple-bit upset (MBU) | Several bits of one protected word changed by one particle (several cells hit is a multiple-cell upset); can defeat a code that corrects one bit per word | Single | No |
| Single event tran-sient (SET) | A short glitch in a circuit, logic or analog; in logic it becomes an upset if a storage element captures it | Single | No |
| Single event func-tional interrupt (SEFI) | The device stops working properly until it is reset, reconfigured or power cycled | Single | No, but an outage |
| Single event latch-up (SEL) | A parasitic structure turns on and draws a high current; it can burn the part or leave latent damage unless power is removed in time | Single | Potentially, unless power is cut in time |
| Single event burnout (SEB) | A particle triggers destructive current in a power transistor | Single | Yes |

| Effect | Plain meaning | Kind | Destructive? |
|---|---|---|---|
| Single event gate rupture (SEGR) | A particle breaks down a transistor's gate insulator | Single | Yes |

A part's figures must be read effect by effect: surviving a dose says nothing about latch-up, and immunity to latch-up says nothing about upsets.

*Source: J. Pellish, NASA GSFC, Tutorial: Radiation Effects in Electronic Systems, Applied Space Environments Conference, May 2017 (NTRS 20170004674), slides 5, 8 and 12; ESA, Radiation: satellites' unseen enemy, 4 July 2011; NASA, State-of-the-Art of Small Spacecraft Technology, chapter 8 (2026). Plain meanings are the author's wording of those sources.*

The single-event side of the subject is younger than the dose side. Pellish's tutorial, after Buchner, dates the idea of single event upsets to a 1962 paper by Wallmark and Marcus, the first reported satellite anomalies from upsets to Binder, Smith and Holman in 1975, upsets on the ground from alpha particles in packaging to May and Woods in 1978, and the first report of a destructive single event in a memory operating in space, a proton-induced latch-up, to Adams and others in 1992. Telstar's problem in 1962 was dose.

## The environment: where the particles come from

ESA's 2011 description lists three sources: "high-energy particles from the Sun, belts of protons, electrons and ions collected within Earth's magnetic field" and "cosmic rays", particles "arriving from beyond the Solar System". Each source affects a computer differently.

- **The trapped belts** hold protons and electrons in Earth's magnetic field. They deliver most of the dose in many Earth orbits. Where trapped particles of the inner belt reach unusually low altitudes, over the South Atlantic, a satellite in low orbit crosses a region of enhanced flux, the South Atlantic Anomaly. Kazak and others (Electronics, 2026) note that during passages through it, and during solar flares, the upset rate of a memory "can increase by several orders of magnitude", a statement about the memories they studied, not a rate for any part in this book. The same paper names the dominant particles by orbit: in low orbit, protons from 10 MeV to 1 GeV; in geostationary orbit, also a significant contribution from electrons of up to several MeV.
- **Solar particle events** come in bursts tied to solar activity. ESA notes that "the 11-year cycle of solar activity also plays an important role", and NASA's tutorial shows the "Halloween storms" of 18 October to 7 November 2003 as an example of single events driven by solar eruptions.

- **Galactic cosmic rays** include heavy ions of very high energy; Kazak and others give their make-up as up to 90 percent protons, up to 13 percent helium nuclei and less than 1 percent heavier ions, with a flux modulated by the Earth's magnetic field and by solar activity. ESA's group notes that "we see more light ions than heavy ions, and much more protons and electrons than light ions", but a heavy ion deposits a lot of charge along its track, which is why heavy ions dominate many single-event tests.

**Low orbit is not free of radiation.** Pellish's tutorial notes that in "proton-dominated environments", and it names low Earth orbit, protons "can be a large portion of the overall SEE rate". A proton deposits little charge directly; most of its upsets come from nuclear reactions in the chip, whose fragments have appreciable linear energy transfer, usually below 15 MeV cm2/mg in the tutorial's figure. The same tutorial notes that low-energy protons can also cause single events by direct ionization, which matters for the most sensitive modern parts, so a proton test has to cover the energies that reach the part. NASA's 2026 avionics chapter adds that "longer LEO missions compel developers to consider reliability requirements and incorporate radiation-mitigation strategies". ESA's engineers said in 2011 that with smaller transistors "not only do we have to be concerned with heavy ion-related effects but also increasingly with proton effects", and that "proton-induced effects are particularly problematic due to their high fluxes in Earth orbit".

### How much: dose against shielding, for three orbits

The quantity a design needs is the dose a part will receive behind the material around it, over the mission, at a stated confidence. It is computed, not looked up: an environment model of the trapped particles, a model of solar particles at a confidence level, a launch date that fixes the phase of the solar cycle, and a calculation of how much of the radiation the shielding stops. The result is a dose-depth curve: dose against shielding thickness, usually in millimeters of aluminum or its equivalent. This book has not run such a model. It prints two published model runs and one literature value, each with its own inputs, and does not merge them, because they differ in orbit, duration, solar phase and depth.

- **550 and 600 km, sun-synchronous, five years.** The German Aerospace Center (DLR) designed its CubeSat laser terminal for five years at 550 km. Its 2023 paper gives the model: OMERE with the AX-9 trapped-particle mean model, the ESP solar proton model at 90 percent confidence, the Psychic model for solar ions, and a launch on 1 January 2024, near a solar maximum. Behind at least 1.5 mm of aluminum equivalent, the expected dose is 17 krad(Si) at 550 km and 18 krad(Si) at 600 km.
- **800 km, sun-synchronous at 98.5 degrees, three years at solar maximum.** Samwel and others (Advances in Space Research, 2019), as reported in a 2020 review by Knap, Vestergaard and Stroe, give 3.73 Mrad with no shielding and 33.1 krad behind 1.5

mm of aluminum. The original paper was not read for this book; the figures are the review's.

- **Geostationary orbit, five years.** Turyshev's 2026 preprint on orbital data centers (arXiv 2604.27197, not peer reviewed), labeled here as research, uses a literature example of about 20 krad(Si) behind 200 mil of aluminum (5.1 mm) for five years, and states that its own dose values are "surrogate design brackets rather than percentile-specific AE9/AP9 mission outputs".

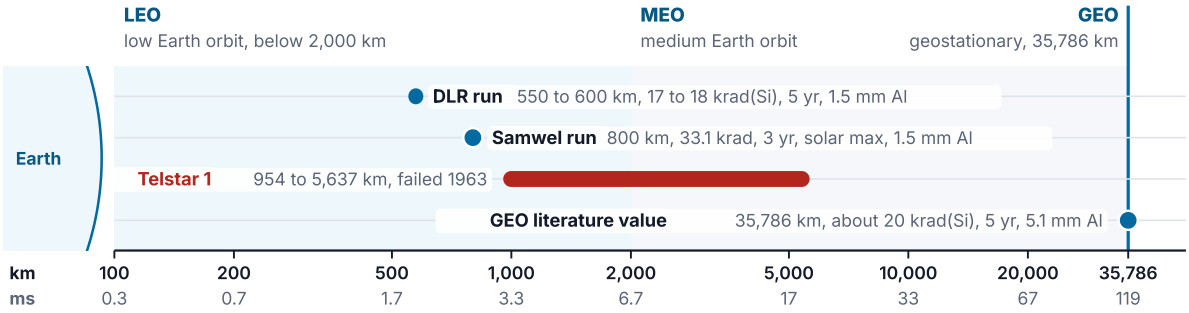
Two lessons follow from these numbers without stretching them. First, a millimeter or two of aluminum makes an enormous difference to dose: in the 800 km case the shielded figure is about one hundred-thirteenth of the unshielded one (3.73 Mrad divided by 33.1 krad, calculated). The particles that deliver most of the unshielded dose have low energy and stop in the first layers of metal. Second, the shielded doses of these low orbits, tens of krad(Si) over a few years, are of the same order as the dose figures some makers give for commercial-based computers (a Zynq-based computer "tested to 25 krad", chapter 6) and well below the ratings of hardened processors (300 krad(Si) for the GR740, chapter 5). That is why total dose alone need not rule out some commercial-based parts in low orbit, and why it settles nothing by itself: a maker's test figure is not an acceptance, which needs the test's conditions, a margin over the mission dose and the single-event work below.

What shielding does not do is just as important. The fragments of a proton's nuclear reaction are created inside the chip itself, and Pellish's tutorial contrasts their short ranges with those of cosmic rays. The particles that cause most single events are not the low-energy ones that a few millimeters stop, so the large fall in dose above says nothing about upset rates. The arXiv preprint gives the general form of a rate: the integral over energy of the particle flux after the shielding times the device's cross-section, both orbit- and shielding-dependent. No source read for this book gives a single figure for how much a few millimeters reduce upset rates, and none is printed here. Shielding is also mass, which the trade sheet counts on line 8.

Figure 3.2 places the two runs, the literature value and Telstar's orbit on the altitude axis. Proton and electron dose drawn separately for each orbit would be more useful; the sources read do not separate them, so the figure gives the totals with their inputs.

FIGURE 3.2

## Two published dose runs, one literature value and Telstar 1's orbit, each with its own inputs




km: altitude above the surface (log scale). ms: light time straight up, one way (altitude divided by the speed of light).

The dose figures come from three different studies with different durations, solar phases and shielding depths, so they place the cases on the axis and are not to be compared as one curve.

*Source: Rodiger and others, DLR, SmallSat Conference 2023, SSC23-IX-04 (OMERE, AX-9 mean, ESP 90 percent, launch 1 January 2024); Samwel and others, Advances in Space Research 64 (2019), as reported by Knap, Vestergaard and Stroe, Energies 13 (2020) 4097; Turyshev, arXiv 2604.27197 (2026 preprint, not peer reviewed; geostationary altitude and literature value); NASA History, Telstar page, and Bell System Technical Journal 42 (4), July 1963. Read 2026-11-24.*

## Units on a datasheet

A radiation line on a part's datasheet carries four kinds of number. Each is a result stated under conditions, measured or calculated from a test, not a property of the part alone.

**Dose, in krad(Si).** The total dose a part withstood in a test, in thousands of rad absorbed in silicon (one rad is 0.01 joule per kilogram). NASA's 2026 avionics chapter defines total dose as "the ionizing radiation absorbed by the device material over time" and states it in krad. A dose rating says the parts tested still met their limits after that dose, at that dose rate, under that bias. It does not say how long the part lives in a given orbit; the mission's dose-depth figure and a margin do that.

**Linear energy transfer (LET), in MeV cm2/mg.** The energy a particle deposits per unit length of its track, divided by the density of the material. It is the yardstick of single-event tests: a heavy-ion test exposes a part to ions of known LET and counts events. A latch-up figure such as "latch-up immune to 125 MeV cm2/mg", the form Frontgrade Gaisler uses for its GR740 processor beside a dose of 300 krad(Si) (chapter 5), means that no latch-up was seen up to that LET in the tests run, at the voltage, temperature and fluence of those

tests. Zero events bound the cross-section; they do not set it to zero. For a Poisson count, zero events at a fluence F give a 95 percent upper bound of about 3 divided by F (minus the natural logarithm of 0.05 is 3.0, calculated), which is why a test report gives its fluence.

**Cross-section, in cm2 (per device or per bit).** The number of events observed divided by the fluence, the number of particles per square centimeter that struck the part during the test, whether they caused an event or not; a per-bit figure also divides by the number of bits monitored. For heavy ions it is plotted against LET. A NASA Goddard presentation of 2016 by Berg and LaBel shows such curves for shift registers in two FPGA families, in cm2 per bit, from 1E-12 to 1E-5, rising with LET. The cross-section curve is what turns an environment into a rate.

**A rate, in events per device per day.** A rate combines the particle spectrum in the orbit behind the shielding with a matching cross-section curve, and states the operating mode (clock speed, voltage, which blocks are on). A per-bit cross-section is scaled by the number of bits actually in use; a per-device one is not scaled again. A rate quoted without these inputs is a figure for someone else's mission.

A test result, finally, speaks for a sample. Chapter 6 shows a maker's summary of one ground test, twenty parts from five wafer lots, with its stated voltage and temperature, and explains how a buyer reads it.

## Ground tests: what each one shows

Nobody waits for orbit to learn how a part behaves. Radiation tests on the ground use sources chosen to stand in for the environment.

- **Dose** is usually tested with gamma rays. ESA's component laboratory described its approach in 2011: "we use radiation sources representative of the worst case we see in space, for example with the cobalt-60 gamma source we have on site". Dose tests are run at a stated rate, which is why ELDRS matters: a fast test can flatter a part that degrades more at the slow rates of orbit.
- **Single events** are tested with heavy ions and protons at accelerators. NASA Goddard's Radiation Effects and Analysis Group states on its page (updated 18 November 2026) that it "regularly travels to high- and medium-energy domestic cyclotron and synchrotron facilities to perform heavy-ion single-event effects (SEE) testing", uses the NASA Space Radiation Laboratory for high-energy heavy ions, and keeps "30+ years" of results in Goddard's radiation database.
- **Published compendia** make results reusable. NASA's Electronic Parts and Packaging program publishes yearly compendia of single-event, dose and displacement damage results for candidate spacecraft electronics, and states that ground testing is used to

assess flight risk.

Every result is bound to its conditions: the lot or wafer the parts came from, the bias applied during exposure, the temperature, the dose rate or the ion and its energy, and the failure criterion, which is the line between "still working" and "failed" that the tester chose. A result without its failure criterion cannot be compared with another. Some parts must also be prepared: the tutorial shows a 28 nm commercial FPGA "thinned in preparation for SEE testing", a step that belongs to the test and not to the flight part.

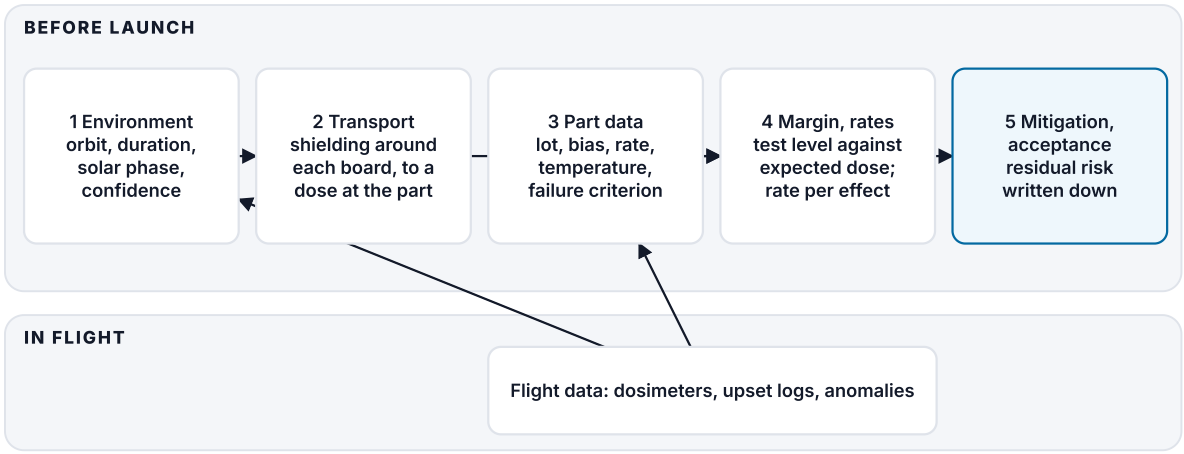
## Assurance: a chain, not a certificate

Radiation hardness assurance is the name for the work that connects the environment to an acceptance decision. NASA Goddard's group describes its own work as end-to-end: environmental modeling, radiation assessments and testing, "requirements generation and verification, system analyses, mitigations, and strategic recommendations". It adds that "radiation transport analyses bring that environment down to the part level to minimize unnecessary conservatism". The European standard for the same work is ECSS-Q-ST-60-15C, titled "Radiation hardness assurance, EEE components" (2012, as cited by Knap and others); its text was not read for this book, so its clauses are not described.

Figure 3.3 draws the chain in five links. Each link needs an owner and produces an output, and each can break the result.

FIGURE 3.3

## Hardness assurance is a chain of five links, each with an output that the next one uses




A weak link anywhere, such as a model run at the wrong solar phase or a test at an unrepresentative dose rate, carries through to the acceptance decision.

*Author's diagram based on NASA GSFC Radiation Effects and Analysis Group, capabilities page (updated 2026-11-18), and the worked chain in Rodiger and others, DLR, SmallSat 2023 (18 krad(Si) modeled, 25 percent margin, 23 krad(Si) tested).*

1. **Environment specification:** the orbit, the mission length, the launch date and so the phase of the solar cycle, and the confidence level, turned into particle spectra by named models.
2. **Transport to the part:** the spacecraft's actual mass around each board, turned into a dose and a spectrum at the part. A simplified geometry gives a first estimate, whose conservatism has to be shown rather than assumed; a detailed transport analysis, in the Goddard group's words, minimizes "unnecessary conservatism" that a design would otherwise pay for in mass or part grade.
3. **Part data:** test results for this part, ideally this lot, under conditions close to the flight use.
4. **Margin and rates:** the test level against the expected dose, with a stated factor; single-event rates from cross-sections and spectra, for each effect.
5. **Mitigation and acceptance:** what the design does about the rates that remain (chapter 4), and the decision, with its residual risk, written down.

   The DLR laser terminal shows the chain in one paragraph of a public paper. The model gave 18 krad(Si) for five years at 600 km behind 1.5 mm of aluminum. With a 25 percent safety margin that is 22.5 krad(Si) (calculated); the team irradiated the parts to 23 krad(Si) with a cobalt-60 source at the Helmholtz Center in Berlin, at a constant 1 krad per hour, and the team reports that "all components will operate within the defined specifications and

margins". Several links of the chain are named in public: environment, depth, margin, test source and rate. The paragraph is not a complete qualification record: it does not give the sample, the bias or the numerical acceptance limits. And it does not cover single events; the authors set them aside for that orbit in their own analysis, and the paragraph does not show why; a reader judging this part, or another, in the same orbit would ask for that reasoning.

## Smaller transistors: neither direction is a rule

Two published views point in opposite directions, and both are right about something. ESA wrote in 2011 that "circuit features have become smaller and run on less charge, making them more vulnerable to disruption by incoming charged particles". Berg and LaBel's 2016 NASA Goddard presentation shows FPGA logic capacity rising from 150 nm to 16 nm and draws the other consequence: "As geometries get smaller, more gates are available for mitigation". Less charge per bit means a smaller particle can flip a bit, and one track can cross several bits at once; more gates per chip means more room for correction and triplication.

Pellish's tutorial adds the practical point: new transistor structures (FinFETs, ultra-thin silicon on insulator and others) bring "additional evaluation challenges", and the field's test techniques and analysis must evolve with them. The process does not decide the answer. A part on the newest process is not barred from flight, and a part on an old one is not safe by age; each is characterized on its own, with its conditions. Chapter 7 shows how three FPGA makers state that characterization for their parts.

## Line 7 of the sheet, for three orbits

The trade sheet's radiation line takes the environment from a cited model run and keeps four outcomes apart: raw events, errors the design corrects, outages, and hazardous failures. With only the sources in this chapter, the line can be written for dose with numbers and for single events only in words. Table 3.1 does so. Chapter 4 adds what mitigation leaves.

**Table 3.1. Line 7 of the trade sheet for three orbits, from published runs**

| Orbit and mission | Dose behind the stated shielding (source, inputs) | Single events, qualitative | What the line still needs |
|---|---|---|---|
| 550 to 600 km, sun-synchronous, 5 years, launch near solar maximum | 17 to 18 krad(Si) behind 1.5 mm Al equivalent (simulated: DLR, 2023; OMERE, AX-9 mean, ESP 90 percent) | Drivers: trapped protons, with peaks in the South Atlantic Anomaly; solar events; cosmic rays. No rate cited | Cross-sections of the parts used; a proton and heavy-ion rate per effect |
| 800 km, sun-synchronous, 3 years at solar maximum | 33.1 krad behind 1.5 mm Al; 3.73 Mrad unshielded (simulated: Samwel and others, 2019, as reported by Knap and others, 2020) | The same drivers. No rate cited | The original run's models and confidence; the rates |
| Geostationary, 5 years | About 20 krad(Si) behind 5.1 mm Al (assumption, to verify: a literature value quoted in a 2026 preprint, not peer reviewed; the original model run was not read) | Drivers: cosmic rays and solar events; electrons add to the dose. No rate cited | A mission model run; surface and internal charging, which this book does not cover |

The single-event column names drivers and gives no rank, because no cited source gives rates for these orbits and these parts, and a rank without a rate would only look like one.

### FAILURE AND RECOVERY: TELSTAR 1'S COMMAND DECODER, 1962 TO 1963

Telstar 1 was launched on 10 July 1962 into an elliptical orbit of 593 by 3,503 miles (about 954 by 5,637 km, calculated); its builders wrote that the region it crossed "includes essentially the whole of the inner Van Allen belt". In its first months it relayed television across the Atlantic 47 times. Its builders' account in the Bell System Technical Journal of July 1963 gives the rest in sequence. In the fourth week of November 1962 "the command channel began to act erratically". To protect the batteries the operators left the telemetry on and stopped switching on the traveling-wave tube. After orbit 1,242 on 23 November the command channel ceased responding. Telemetry showed high-energy electrons at a much higher density than expected, and the builders' working hypothesis was radiation damage to transistors in the command decoder; reviewing earlier data, they found that one of the two decoders had already failed in August.

Recovery came from the ground. Laboratory tests pointed to the transistors most likely at fault, and the team devised special codes that bypassed them. On 20 December 1962 a modified code was accepted; with all voltages then removed from the decoders, the transistors recovered, as predicted, and public television demonstrations ran until 4 January 1963. From 14 February the decoder degraded again, and on 21 February "the satellite misinterpreted a command and operated the relay that disconnects most of the electronic system from the power plant". By the paper's writing on 18 March it had not responded since.

The companion paper on the spacecraft's radiation experiments explains the cause the builders named: the satellite met "not only the natural particles in the radiation belts" but "particles artificially introduced into the belts by U. S. and Russian high-altitude nuclear explosions", which "were destined to play a major role in the radiation effects sustained by the satellite's electronic equipment". One lesson follows from the source, and it has lasted: the environment a design assumes is itself an input that can be wrong, and a satellite that can still accept a command can sometimes be worked around the damage. ESA, writing in 2011, calls Telstar 1 "an early high-profile casualty".

**SECURITY NOTE**

**Who can interfere or command.** Radiation is not an adversary, but it sets the baseline every security design must live with: corrupted state and odd behavior happen without anyone's intent. The fault log is the record of that baseline. Five questions apply to it: who may command the computer and clear the log; who can read the log on board and on the ground; who could interfere with it, by filling it or by editing it in a hosted system; how a command to clear or change it is authenticated; and how the log survives a reset.

**How it is seen.** A computer can detect a corrupted memory word, a failed check or an unexpected reset without knowing whether a particle or an intruder caused it. Detection, diagnosis, attribution and recovery are separate steps. The first needs only the check; the second needs context, such as whether the event matches a pass through the South Atlantic Anomaly or a solar event; the third needs more than any single log line can give.

**How it is recovered.** The recovery paths are the same whatever the cause: a reset, a reload from a protected copy, a switch to a spare. Telstar's operators kept the satellite under command from 20 December 1962 until the decoder degraded again in February 1963, by sending codes that worked around damaged parts, which is also a reminder that a command path accepting unusual codes is a capability to protect. Chapter 4 follows the repair mechanisms themselves.

**Monday rule:** before quoting any dose or upset figure, write the orbit, the mission length, the phase of the solar cycle and the shielding depth beside it, and name the model that produced it.

CHAPTER FOUR

# Mitigation, Layer by Layer, and What It Leaves




At the IEEE Aerospace Conference in March 2017, a team from the University of Southern California and NASA Goddard reported a flight experiment on the International Space Station. Two commercial processors inside a Xilinx Virtex-5 FPGA ran a benchmark for 16 days under software checks, watched by a third processor built in the FPGA's logic. The experiment, in their words, "collected 19,400 checkpoints, processed 253,482 status messages, and incurred 0 faults". The checks cost about 1.3 percent of the processors' time. A zero like that is easy to quote and difficult to interpret: faults may have occurred and been masked or caught, or no particle may have struck anything that mattered in 16 days, and the record cannot tell which. This chapter takes each radiation effect from chapter 3, names the design answer to it, and asks two questions of every answer: what does it cost, and what risk does it leave? It changes how a reader judges a block diagram full of protections.

## One table: each effect and its answer

Chapter 3 split radiation damage into cumulative effects and single events, and single events into those that destroy and those that do not. Each row of that split needs its own answer, and coverage of one row does not establish coverage of another. Figure 4.1 sets them side by side.

FIGURE 4.1

## Each effect has its own answer, and each answer has a cost and leaves a risk

| Effect | Mitigation | Where it applies | What it costs | What it leaves |
|---|---|---|---|---|
| Total dose | Part choice, shielding, margin, derating | Every part, before launch | Part grade, mass, analysis | Lot-to-lot spread; slow dose-rate effects; a mission longer than planned |
| Displacement damage | Part choice, shielding | Detectors, optical parts, bipolar parts | As above | As above |
| Upsets in memory | Error correcting codes, interleaving, scrubbing | Memories and caches | Extra bits, access time, scrub bandwidth | Errors beyond the code's strength, such as two in one word before a scrub; addresses outside the scrub; faults in the code or scrub logic |
| Upsets in logic and flip-flops | Triplication and voting (local, distributed, global) | Logic, state machines | Three times the logic and more; power; timing; voters | Unrepaired copies accumulate faults; voter and clock upsets; unverified insertion |
| Configuration upsets (SRAM FPGA) | Scrubbing, with triplication to mask until repaired | SRAM-configured FPGAs | Scrub controller, readback time | Wrong state left in the logic after the bit is fixed; faults between scrubs |
| Transients | Filters, triplication, timing margin | Combinational paths, clocks | Speed | Transients in unhardened global routes |
| Functional interrupts | Watchdog, external supervisor, reset, power cycle, reconfiguration | Processors, complex devices | A second, more robust device; downtime | Detection latency; the supervisor's own failure |

| Effect | Mitigation | Where it applies | What it costs | What it leaves |
|---|---|---|---|---|
| Latch-up | Parts with no latch-up in test, within that test's coverage; current limiting and power cut | Every powered part | Protection circuit; restart time | Damage faster than the cut; supplies not watched |
| Burnout and gate rupture | Part choice, voltage derating, tests at the mission's bias | Power transistors and converters | Derating limits; test campaigns | Damage faster than any power cut; parts not tested at the bias they see |

Read across a row, never down a column: coverage of one effect does not establish coverage of another, and every answer is itself a circuit or a program that can fail.

*Source: M. Berg and K. LaBel, NASA GSFC, SEE Symposium and MAPLD 2016 (NTRS 20160007971); NASA, State-of-the-Art of Small Spacecraft Technology, chapter 8 (2026); AMD at ESA SEFUW, 25 March 2025 (EDAC and interleaving); Schmidt, French and Flatley, IEEE Aerospace 2017; author's arrangement.*

Four points in the table need more than a cell.

**Dose is answered before launch.** Total dose and displacement damage are not corrected in flight; they are kept below the part's limit, with margin, by choosing the part, the shielding and the mission length (chapter 3's chain). The one exception is partial recovery, such as Telstar's transistors recovering when their voltages were removed, which a design cannot plan on without data for its own parts.

**Upsets in memory are answered by codes and by scrubbing.** An error detecting and correcting code (EDAC) stores extra bits with each word so that a changed bit can be found and put right. With a common form of code that corrects one bit and detects two in each word, two bits changed in the same word, for example by one particle, are detected but not corrected, and three or more can be miscorrected or pass unseen. Designers therefore spread the bits of each word apart in the memory (interleaving), which reduces that risk without removing it: AMD's 2025 description of its space FPGAs states that their configuration memory is "protected by EDAC and interleaving". Correcting a word as it is read does not by itself repair the stored copy; that needs a write back. A scrubber reads and rewrites memory on a schedule, including words the software rarely reads, so that single errors are removed before a second one arrives in the same word. The interval is the design choice, and the actual code must be named before its residual risk can be stated.

**Functional interrupts are answered from outside the device.** A device in a functional interrupt cannot be relied on to notice its own state. A watchdog timer, which resets the device unless the software clears it in time, or a supervisor that watches heartbeats, has to stand outside the failing part's fault domain: its logic, and also the clock, power and reset it depends on.

**Latch-up is answered by removing power fast enough.** A current limiter or a latch-up protection circuit detects the rise in current and cuts the supply. It helps only if detection and shutdown come before the damage, and only for the circuits it watches. Test results with no latch-up, such as the GR740's stated immunity to 125 MeV cm2/mg (chapter 5), can support a decision to omit dedicated protection, subject to the mission's own assurance and to the coverage of those tests.

NASA's State-of-the-Art of Small Spacecraft Technology, in its avionics chapter as updated on 21 October 2026, describes how these answers are usually combined in small satellites: "COTS components first (e.g., processor and memory), combined with rad-hardened supporting electronics such as ECC, watchdog timers, scrubbing, and redundancy". The rest of this chapter follows what each of those costs.

## Triplication and voting: what three copies buy

Triple modular redundancy (TMR) runs three copies of a circuit and passes on the answer that at least two agree on. A single upset in one copy is outvoted. The idea is old and simple; what it buys depends on details that a block diagram does not show. The clearest public account is a 2016 presentation by Melanie Berg and Kenneth LaBel of NASA Goddard, given at the Single Event Effects Symposium and the Military and Aerospace Programmable Logic Devices workshop. One of its slides is titled "Redundancy Is Not Enough".

**Three copies without repair wear out faster.** Berg and LaBel work through block TMR, in which a whole function is copied three times and voted only at its outputs, with a classical reliability model: identical copies that fail independently at a constant rate, a perfect voter, and no repair. If one copy has a constant failure rate $\lambda$ and nothing repairs a failed copy, the three-copy system works while at least two copies work. Its reliability is $3e^{-2\lambda t}$ minus $2e^{-3\lambda t}$, and its mean time to failure is 5/6 of one copy's. Triplication without repair therefore fails sooner on average than a single copy. It is more reliable than one copy only early in the mission: solving the two expressions shows the crossover where one copy's reliability falls to one half, at a time of ln 2 divided by $\lambda$, about 0.69 of one copy's mean time to failure (calculated). After that the triplicated system is less reliable than a single copy. Berg and LaBel's advice follows: block TMR suits functions that are flushed regularly, such as a transmission channel reset after every packet, or systems reset every so often. Three copies with no repair do not give three times the reliability.

**Where the voters sit matters.** The presentation names four forms. Local TMR copies only the flip-flops. Distributed TMR copies the data paths too. Global TMR also copies the clocks and other global routes. Block TMR copies whole blocks. The right form depends on which part of the device is most sensitive. For SRAM-based FPGAs the presentation is direct: "Proven via NEPP experiments: SEU data for LTMR implemented in Xilinx FPGA devices are similar or worse than no added mitigation", because the configuration memory and the routing, not the flip-flops, carry most of the sensitivity. It adds that unhardened global routes "do cause system upsets", and that even the hardened Virtex-5QV and the flash-based ProASIC3 "do not have hardened global routes".

**Three times the logic, and more.** The costs are area, power and timing: triplicated logic plus voters, and voters in the path that lengthen it. Global TMR adds three clock trees, which must meet the design's timing with their skew included; the presentation's simplified rule is that skew between clock domains must stay below the shortest routing delay from one flip-flop to the next, a simplification of the hold check; the design's full timing analysis, for minimum and maximum delays, decides. The presentation calls global TMR "theoretically" the strongest scheme and lists why it is often not used: power, area, the tools that insert it, and verification.

**The protection itself must be verified.** In their 2016 presentation Berg and LaBel warned that "there are currently no available techniques that can provide complete and reliable confirmation of TMR insertion", and that "complexity is a RISK". A tool that triplicates a netlist can break the logic or merge the copies, and a design that claims TMR may not have it where the tests say it is needed. "We know it will work," the slide reads, is not a good enough answer: "Ask how and if the scheme has been verified!"

**The voter is one more circuit.** A single voter after three copies is a single point of failure. Distributed and global TMR place voters inside each copy so that a voter upset is itself outvoted at the next stage. A designer who places one voter at the output has moved the weak point, not removed it.

## Scrubbing, and what a fixed bit does not fix

An SRAM-based FPGA holds its circuit in configuration memory, and an upset there can change the circuit itself. Berg and LaBel define the answer: "Scrubbing is the act of simultaneously writing into FPGA configuration memory as the device's functional logic area is operating with the intent of correcting configuration memory bit errors." Configuration scrubbing in this sense repairs SRAM configuration; memory scrubbing, above, applies the same idea, periodic inspection and correction, to data, using the memory's error correcting code. Chapter 7 explains why flash and antifuse parts hold their configuration differently.

Their warning is the reason scrubbing is not a complete answer: "Fixing a configuration bit does not mean that you have fixed the state in the functional logic path." After a configuration bit that mattered is repaired, the flip-flops it fed may hold wrong values. Getting back to a known state takes the fixed bit plus a reset or a correction of the flip-flops, or a full reconfiguration. Between scrubs, as the 2017 flight paper puts it, "there remains a possibility, based on the periodicity of the scrubbing, where a fault may still occur", which is why scrubbing is paired with triplication, so that a fault is masked until the next scrub repairs it. Chapter 7 draws the full repair loop for an FPGA.

### Redundancy above the chip

When the chip cannot be trusted alone, the next layer is the board or the computer. Berg and LaBel list the main forms and what each cannot do.

- **Dual redundancy** compares two copies. "Dual redundant systems cannot correct (roll-back is an exception); they can only detect." Two copies in lockstep must be kept exactly synchronized, which is "challenging and sometimes unpredictable" with caches and interrupts; on a mismatch the usual answer is a reset or a power cycle.
- **Cold sparing** keeps a second unit switched off until the first fails. It restores service after a failure, with downtime for detection and switch-over; Berg and LaBel's slide puts it as "availability is not improved... there is downtime": a cold spare cannot avoid an interruption. Over a mission, against a single unit that fails for good, it does raise availability and restores service; by how much depends on detection, switch-over and the spare's own survival.
- **Voting computers** run the same software on several machines and compare outputs. Chapter 2 follows the Space Shuttle's redundant set of four computers and the fifth that ran separately written backup software, and the scrubbed first launch attempt of 1981, which showed that the join between a voting set and its backup is itself a design with its own faults.
- **A hardened supervisor over a fast commercial processor** is the common pattern in small satellites today. NASA's avionics table (updated 21 October 2026) lists, for example, a Belgian onboard computer built on a Zynq-7045 with a "radiation hardened supervisor MCU" and a "radiation hardened watchdog", and gives its radiation entry as heavy ions (62.5 MeV) with "no destructive event"; the table gives no LET, fluence, voltage or temperature for that test, so it says what was reported, not under which conditions. The supervisor does little computing; its job is to restart, switch or power cycle the processor when the heartbeats stop.

Above all of these sits fault protection: software on the spacecraft that detects faults and puts the spacecraft into a safe configuration until the ground can act. Chapter 2 traces the watchdog from Apollo's NIGHT WATCHMAN and Voyager's heartbeat, and how flight software implements them belongs to a later book of this series, on flight software. Here the point is narrower: every layer of redundancy needs a detection step, and every detection step has a latency. Berg and LaBel's review questions apply at every layer: can the system detect all the error signatures, fast enough, and how much downtime will recovery take?

## Mitigation in software

When the hardware cannot be triplicated, software can check its own work. The 2017 flight experiment is a clean example because its authors explain why they needed it. The Virtex-5's two embedded PowerPC 440 processors have internal state, such as caches and registers, that configuration scrubbing cannot read or correct, and two processors per FPGA are not enough to triplicate. So the team used four techniques taken from high performance computing:

- **checkpoint and restart:** the state of the computation is saved at intervals and restored after a fault;
- **heartbeat monitoring:** each processor reports regularly to a controller, a soft MicroBlaze processor in the FPGA's logic;
- **control flow assertions:** markers in the program check that it passes the expected points in the expected order;
- **watchdog timers:** each processor's built-in watchdog, set to reset the processor after 5 seconds without a heartbeat.

The authors report overhead "typically <1.3%" and "a 3.3x gain in processing performance as compared to the equivalent traditionally radiation hardened processor", a result of their own benchmark that is not the ratio of the MIPS figures below. Their comparison table gives the RAD750 at 300 MIPS for 10 to 20 W and the SpaceCube 2.0 at 5,000 MIPS for 10 to 20 W; these are the authors' figures for their own platform, not an independent benchmark. Software mitigation finds a fault after it happens and restores a good state; it does not prevent a fault, and it does not protect what it cannot observe. How such checks are written belongs to that later book on flight software; this chapter needs only their place in the stack.

## What mitigation cannot do

The simple reliability models behind these mitigations assume random, independent events: one particle, one place, one time. One particle can still upset several cells, which interleaving and physical separation answer, or a resource the copies share. Three things lie outside that model.

- **A design fault common to all copies.** Three copies of a wrong circuit vote unanimously for the wrong answer. Berg and LaBel add that "incorrect implementation can increase error": mitigation is itself a design, with its own faults.
- **A dose beyond the part's limit.** No code or vote restores a transistor whose threshold has drifted out of range. Redundant copies of the same part share the same dose and age together.
- **A wrong answer nobody checks.** A result that is plausible, comes out on time and is not compared with anything passes every mechanism in this chapter. The presentation's discussion of "safe" state machines shows a small case: a tool's "safe" option returns an upset state machine to a reset state when it lands in an unused state, but does nothing when an upset moves it into another valid state, which the presentation calls a misnomer. Chapter 10 applies the same point to learned models.

### A worked product: one small-satellite computer, row by row

One maker's own description shows the layers assembled. AAC Clyde Space describes its Sirius onboard computer as running a LEON3FT soft processor at 50 MHz, with TMR on all FPGA flip-flops, EDAC-protected memories, latch-up protection and the RTEMS real-time operating system, at 1.3 W nominal and a total dose of 20 krad. NASA's avionics table gives the same 20 krad as a "system level" figure and lists the orbits flown as low Earth orbit and a lunar lander. Read against figure 4.1:

- **Upsets in the processor's flip-flops:** answered by TMR on all flip-flops, inside a fault-tolerant processor design (the "FT" in LEON3FT).
- **Upsets in memory:** answered by EDAC; neither the maker's page nor NASA's table states the scrub interval.
- **Latch-up:** answered by a protection circuit; its response time and which supplies it watches are not stated.
- **Dose:** 20 krad at system level. It is numerically close to chapter 3's modeled five-year dose at 550 to 600 km behind 1.5 mm of aluminum, but a margin can be judged only with the same units, shielding, test conditions and acceptance criteria on both sides.
- **What it leaves:** whether the FPGA's configuration is protected, and by what, is not stated on either page; nor are single-event test results with their conditions. Both are questions for the maker's data package, not reasons to doubt the product.

A 50 MHz processor at 1.3 W is slow by ground standards. Neither source apportions that speed or that power among the protections, so this book does not say how much of either they cost here.

## Line 7 of the sheet, with mitigation

Chapter 3 filled the radiation line of the trade sheet with the environment. Mitigation adds a second half: what reaches the result after the design has done its work. The sheet keeps four outcomes apart, and mitigation moves events from one to another rather than removing them.

**Table 4.1. Line 7 reworked: where each kind of event goes, before and after mitigation**

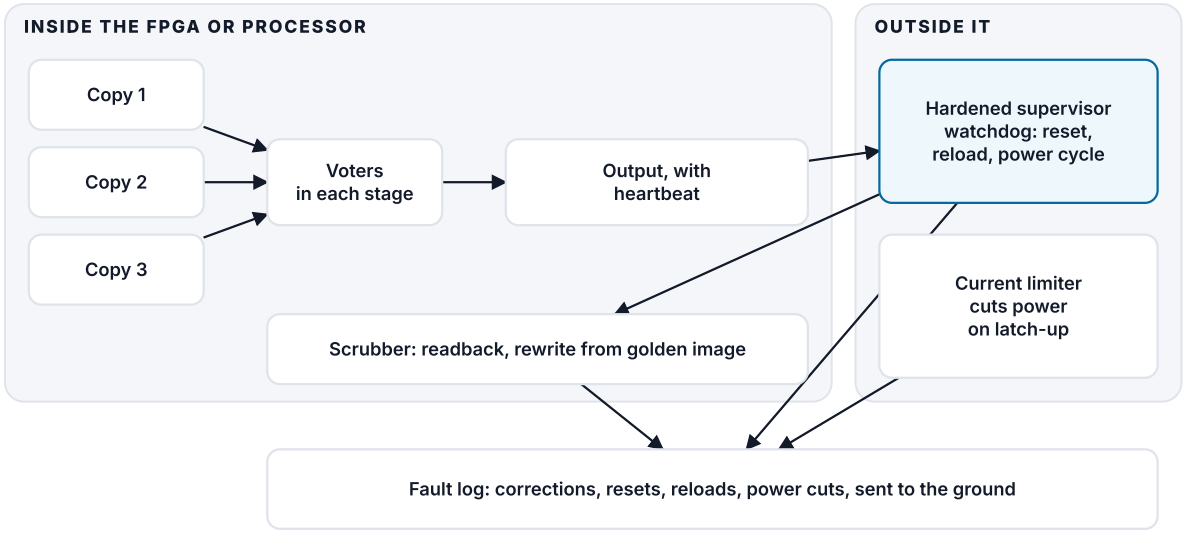
| Outcome | Without mitigation | With the layers of this chapter | What the sheet must still state |
|---|---|---|---|
| Raw events (upsets, transients) | Set by the environment and the part (chapter 3) | Not removed by codes, voting and watchdogs, which act after an event; the rate is that of the hardware actually built, whose extra bits and copies add targets | The rate per effect, from cited cross-sections and spectra, or "not cited" |
| Corrected or masked errors | None | Most single upsets in coded memory are corrected; voting masks an error in one copy, which stays wrong until scrubbing or a reset repairs it | Scrub interval against upset rate; multiple-bit upsets per word |
| Outages (resets, reloads, switchovers) | Functional interrupts hang the device | Shortened by watchdogs and supervisors; each recovery has a downtime | Detection latency and recovery time against the workload's deadline |
| Hazardous failures (wrong output used, part destroyed) | Latch-up can destroy the part; silent errors pass | Latch-up cut off if fast enough; silent errors only caught if checked | What checks the result; what remains common to all copies |

Without cited rates the line stays qualitative. With them, Berg and LaBel's reliability model gives one guide, under its own assumptions: an unrepaired triplicated block is better than one copy only for about the first 0.69 of one copy's mean time to failure, so repair, by

scrubbing or reset, matters more than the number of copies. A scrub does not restore corrupted state by itself, and a reset does not repair permanent damage, so the guide is not a rule for choosing a scrub interval.

FIGURE 4.2

## A voted, scrubbed design under a supervisor: three layers, each watching the one below




The supervisor helps only if the supervised whole is more dependable than the processor alone, which depends on its independence from the failures it must recover from (shared clock, power and reset), its coverage and its false interventions.

*Author's diagram based on Berg and LaBel, NASA GSFC, 2016 (distributed TMR, scrubbing, watchdogs); NASA, State-of-the-Art of Small Spacecraft Technology, chapter 8 (2026: radiation hardened supervisor and watchdog over a commercial processor); Schmidt, French and Flatley, IEEE Aerospace 2017 (heartbeats, watchdog).*

**WHAT THIS STEP REQUIRES**

The bench step of this series for mitigation is stated as requirements, because no result exists yet. It requires: a small counter or state machine written in a hardware description language and triplicated with voters, on an FPGA board that an open toolchain supports; the design, tool versions and build script kept in a versioned repository so the bitstream can be rebuilt; a simulation that flips a chosen bit in one copy and shows the voted output unchanged; a fault injection on the board, by writing a changed value into one copy through a debug port, with the same check; and a log of every injected fault and its outcome. A second run injects a fault into the voter itself, to show the limit. A write through a debug port tests the voter, not radiation: it does not reach configuration bits, routing, several cells at once or a transient caught in flight. Results, when they exist, will be reported with their dates; until then nothing in this book says this step was done.

### A CLEAN RESULT: ZERO FAULTS IN 16 DAYS, AND WHAT IT SHOWS

The lead case of this chapter is a measurement of a mitigation in flight, and its result was zero faults observed. That is not a failure, and the case is chosen because a clean result is what proposals most often quote. Schmidt, French and Flatley report that their experiment was an add-on to the STP-H4 ISS SpaceCube Experiment 2.0, a Department of Defense payload activated on the space station in August 2013. They expected to run for "a few months"; "due to a power switching unit trip on the ISS that removed power to STP-H4, our experiment was able to run for 16 days". In those days the experiment collected 19,400 checkpoints and 253,482 status messages, "incurred 0 faults", and offline analysis of the checkpoints found no silent data corruption. Zero in 16 days bounds the rate of the faults those checks could see: if such faults arrive at a steady random rate, the 95 percent upper bound is about 0.19 a day (minus the natural logarithm of 0.05, divided by 16, calculated). It says nothing about faults the checks could not see, and nothing about another orbit or another workload.

What the record shows: the software checks ran in orbit for 16 days at low overhead, with no alarms raised, on processors in a commercial FPGA. What it does not show: how many particles struck the processors in those 16 days, and therefore whether the checks ever met a fault to detect. The paper treats configuration upsets as outside the scope of this experiment, handled by the platform's existing scrubbing. The earlier support for detection came from fault emulation and laser fault injection on the ground, which the paper describes, and it covers the fault classes injected in those tests. Reading the flight result correctly means writing it as "no faults observed in 16 days in the station's orbit", next to the ground results for the injected faults. The authors' own conclusion is the same: they plan "much longer duration" tests. The experiment's end is its own lesson: the run was cut short by a power trip on the station, a failure outside the computer and outside any mitigation in it.

SECURITY NOTE

**Who can interfere or command.** The mechanisms that repair upsets are also paths to change the design. A scrubber writes configuration memory; a reload replaces a whole FPGA image from a stored copy; a supervisor can reset or power cycle the processor. The five questions apply to each: who is authorized to trigger a reconfiguration or a reload; who can read the stored golden copy; who could interfere, through the command link or through hosted code that reaches the configuration port; how a new image is authenticated before it is written; and how the system returns to a known-good image if a new one fails. The 2017 experiment shows the path concretely: a compressed bitstream of about 360 KB was uploaded in 106-byte segments over about 8 hours, checked with CRC checksums, written to nonvolatile memory, and loaded by command. A CRC checks that the bytes arrived intact; it does not show who sent them.

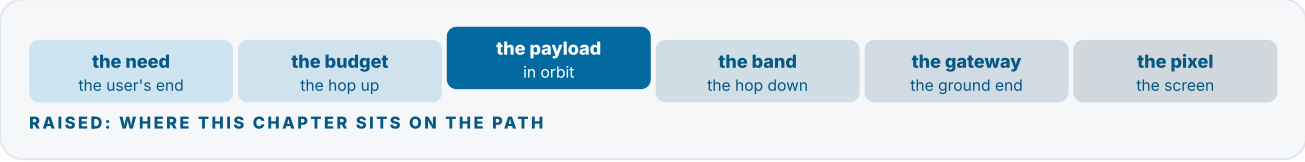
**How it is seen.** A fault injected on purpose, by a laser on the ground or by a write through a debug or configuration port in orbit, looks to the design like a natural upset: one changed bit, one mismatch, one correction. Logs that record where corrections occur, how often, and whether they cluster in ways the radiation environment does not explain are the means to tell the two apart. They support diagnosis, not proof.

**How it is recovered.** Keep the protected copy protected: a golden image in memory that only an authenticated command can rewrite, and a first-stage loader designed and verified to fall back to it when a new image fails. Recovery from a bad image is then the same path as recovery from radiation, which is why it must not be the weakest path on the spacecraft.

**Monday rule:** for every mitigation on a block diagram, ask which effect it answers, what risk it leaves, what checks the mitigation itself, what happens when it is the part that fails, and what repairs it before the mission outlasts it.

CHAPTER FIVE

# Processors and the Avionics Around Them, 1990 to 2026




In February 2026, according to NASA, the first email went out from a High Performance Spaceflight Computing processor, HPSC: the words "Hello Universe". The contract behind it, awarded to Microchip Technology on 15 August 2022 for 50 million dollars, targeted "at least 100 times the computational capacity of current spaceflight computers". In March 2026, NASA's page said, the processor was still in power, performance, reliability and radiation testing, and the project would end once testing showed it "officially 'space qualified'". The processor NASA chose for its Perseverance rover is a RAD750, a radiation-hardened version of a commercial PowerPC 750 that a 2023 policy study dates to 1997. This chapter asks why the processor on a flagship mission of the 2020s is so much older than the phone in an engineer's pocket, whether that is still the right answer, and what around the processor decides how much work it really does.

Chapter 2 ended in the middle of the 1990s with a habit: flying a processor chosen long before launch. This chapter follows the processors that flight computers have used since about 1990, line by line, and then the parts of a computer that are not the processor: memory, storage, buses, clocks and power. Radiation figures are given exactly as each maker states them, with the test condition where the source gives it; chapter 3 explains the units, and chapter 6 explains what each kind of qualification certifies.

## United States: from the 80386 to the RAD5545

A review of 30 years of flight software by a NASA Goddard engineer, given in September 2015, lists the processors of that center's missions with their clocks and memories. Its table runs from an 80386 at 6 MHz with 512 kilobytes of local memory on SAMPEX, through a Mongoose V at 12 MHz on MAP and a ColdFire at 40 and 20 MHz on MMS, to RAD750 processors at 115 MHz on SDO, 118 MHz on the James Webb Space Telescope's instrument module and 132 MHz on GPM. GPM, launched on 27 February 2014 from Tanegashima,

carried a RAD750 "operating at 132 MHz with 36MB RAM, 4MB EEPROM, and 4GB of Solid State Recorder memory". The same review lists a newer RAD750 at 200 MHz and notes that "most next generation flight processors are multicore".

The RAD750 is the processor of record for this period in the United States. Its maker, BAE Systems, states on its own page that the RAD750 "powers more than 100 satellites" and, in its words, is used in the navigation payload of the GPS III constellation; those are the maker's words and its count; the page does not say in what role the processor serves on each satellite, and this book did not check the count. The RAD6000 that came before it appears in a 2019 doctoral thesis among the few radiation-hardened processor platforms that flight software concentrated on, with the LEON3 and the PowerPC 750; no BAE or NASA page on the RAD6000 was read, so no clock, mission list or radiation figure is printed for it here.

The next step in BAE's line carries the RAD5545 name, which covers a multi-core processor and the board built on it; the page read describes the board. BAE describes the RAD5545 as a multi-core single board computer in the SpaceVPX form, one card that can replace several earlier cards and can host software-defined radio payloads (chapter 8 covers such payloads). The processor's datasheet, with its dose and latch-up figures, was not read, so figure 5.2 leaves those cells empty.

### Europe: ERC32, LEON and the GR740

Europe's line is built on SPARC, an open processor architecture. The ERC32, in the description of the RTEMS operating system's board support package, is "based on the European Space Agency's ERC32", a SPARC V7 processor with on-chip peripherals. A 2008 doctoral thesis at the University of Surrey records that ESA's Proba small satellite used "an ERC32 based computer" for "all the platform functions", with a separate payload processing unit for its imager. The LEON processors that followed implement SPARC V8; a LEON3FT soft core running at 50 MHz inside an FPGA is the small-satellite computer worked through in chapter 4, and the NASA Goddard review of 2015 lists a dual-core LEON3FT part, the GR712RC, at 100 MHz.

The GR740 is a quad-core European flight processor, and its maker's page gives its figures in one place. It is "a radiation hardened system-on-chip featuring a quad-core fault-tolerant LEON4FT SPARC V8 processor", with an eight-port SpaceWire router, a PCI interface, CAN and gigabit Ethernet. It "was developed as the European Space Agency's Next Generation Microprocessor (NGMP)". The page gives a system frequency of 250 MHz, a 64 bit PC100 SDRAM memory interface protected by a Reed-Solomon code, SpaceWire links at 300 Mbit/s (a signalling rate; the data that gets through is less, after character and packet overhead), a total ionizing dose of 300 krad(Si) and latch-up immunity to 125 MeV cm2/mg, and ceramic versions qualified to QML Class V and Class Q. It lists the GR740 in the

European Preferred Parts List. These are the maker's figures; the page gives no dose rate, bias, temperature or failure criterion, and the data package that would was not read. Chapter 3 explains why a dose or a latch-up figure without those conditions does not rank parts for a mission. The RTEMS project's home page says that NASA's Nancy Grace Roman Space Telescope uses three GR740 processors in its design (a use, not a flight record; no launch is in the sources read), each running RTEMS; the page does not say which units they serve, and a NASA source for that count was not read.

## India, Japan, China and Russia

**India.** On 5 March 2025, according to ISRO, the first production lots of two 32 bit processors for space applications, VIKRAM3201 and KALPANA3201, were handed over to ISRO's chairman. ISRO's Vikram Sarabhai Space Centre designed them with the Semi-Conductor Laboratory (SCL) in Chandigarh. ISRO describes VIKRAM3201 as "qualified for use in the harsh environmental conditions of launch vehicles", fabricated in SCL's 180 nm CMOS process, with a custom instruction set, floating point and support for the Ada language, and calls it an advanced version of the 16 bit VIKRAM1601, "flying in the Avionics system of ISRO's launch vehicles since 2009", made in SCL's own fab from 2016. KALPANA3201 is a 32 bit SPARC V8 processor that ISRO says "has been tested with flight software". The first VIKRAM3201 devices "successfully validated in space in the Mission Management Computer" of the PSLV Orbital Experimental Module, POEM-4, on the PSLV-C60 mission. POEM is, by its name, an orbital experimental module of the launch vehicle, and this book keeps the two records apart: the source shows launch vehicle heritage and a validation in orbit on that module, not a flight on a satellite.

**Japan.** A 2021 paper by the Space Robotics Laboratory of Tohoku University, in the transactions of the Japan Society for Aeronautical and Space Sciences, describes the processors of Japanese microsatellites: the SuperH series of Renesas Electronics, "widely used especially in Japanese microsatellite projects"; an SH-3 computer on the Earth observation microsatellite REIMEI; and computers based on a silicon on insulator system on chip (SOI-SoC) with an SH-4 core on the microsatellites Hodoyoshi-3 and 4, later used on the interplanetary micro-spacecraft PROCYON. A 2014 paper in the same transactions describes that SOI-SoC processor as "developed for dual use in space and consumer applications", with 64 MB of RAM and SpaceWire as the board's own interface. Both are labeled scholarship, not builder documents. JAXA's own pages on its HR5000 processor and on SDS-1 were sought and could not be read.

**China and Russia.** A 2024 survey in IEEE Communications Surveys and Tutorials compares the BM3883, which it describes as Chinese, with European and US aerospace-grade processors. A builder or institute document with the processor's architecture and

flight record was sought and not read, so no figure from the survey's chart is printed. No Russian flight processor document could be read. Appendix B lists both cases with the date sought.

## Why hardened parts lag, everywhere

The doctoral thesis cited above, written at Leiden University in 2019, puts the lag between the IT industry and space avionics at "between 10 to 40 years", citing an earlier study, and gives the reasons in the industry's own terms. For large satellites, "technology readiness, design maturity, and space heritage of a technology through prior use aboard other spacecraft are essential, and often seen a prerequisite". Construction "often takes many years, sometimes even decades". Chapter 2 showed the same thing in 1977: Galileo's processors were chosen for a 1982 launch that slipped by years.

The lag has more than one cause, and the documents cited here support each only in general terms. Small production volumes are often given as a reason; none of these documents quantifies them. Qualification takes years and is tied to a part and its process (chapter 6). A newer process has to show its radiation behavior again: in Galileo's tests the older, less dense 1802 proved less sensitive than the newer bit-slice chips (chapter 2), and chapter 3 explains why neither direction is a rule. And customers value a flight record, which only an older part can have. The thesis also notes that commercial and agency spaceflight concentrated on "few major OSs (e.g., RTEMS) and processors (e.g., LEON3 and RAD750)": a software base, its tools and its trained people are a reason to keep a processor long after it is fast.

Figure 5.1 gives three dates for each processor where the sources give them: when the architecture was new on the ground, when the part was qualified or handed over, and a documented use in the sources read, which is not necessarily the first flight. Few rows are complete, which is itself the finding: the dates a buyer most needs are often not on the pages a buyer can read.

FIGURE 5.1

## Three dates per processor: few rows can be filled from the pages a buyer can read

| Processor | Architecture new on the ground | Part qualified or handed over | Documented use in the sources read (not necessarily the first flight) | Sources |
|---|---|---|---|---|
| RCA 1802 | | | Chosen c. 1977 for Galileo; Galileo (launch slipped beyond 1986; chapter 9) | NASA history |
| RAD750 | PowerPC 750, 1997 | | GPM, launched 27 Feb 2014 (one documented use; the maker counts more than 100 satellites) | CSET 2023; NASA Goddard 2015; BAE |
| GR740 | | QML Class V and Q (date not on the page) | Roman telescope, per the RTEMS project (use, not a flight record) | Maker; RTEMS |
| VIKRAM1601 | | Made in SCL's fab from 2016 | ISRO launch vehicles since 2009 | ISRO |
| VIKRAM3201 | | Handed over 5 Mar 2025; qualified for launch vehicles | Validated on POEM-4, PSLV-C60 (a launch vehicle module) | ISRO |
| HPSC | RISC-V, an open instruction set (date not given) | Not space qualified as of Mar 2026 | None; contract 15 Aug 2022, first parts late 2025 | NASA JPL; NASA |

Each date is the one its source gives; an empty cell means the sources read do not give it, and a launch vehicle or module flight is kept apart from a satellite flight.

*Source: NASA, Computers in Spaceflight (1988), ch. 6; CSET, Onboard AI: Constraints and Limitations (2023); D. McComas, NASA Goddard (2015); BAE Systems, Radiation-hardened electronics (read 2026-11-22); Frontgrade Gaisler, GR740 (read 2026-11-23); RTEMS project home page (read 2026-11-22); ISRO, VIKRAM3201 (read 2026-11-23); NASA JPL (2022) and NASA's HPSC page (read 2026-11-23).*

## RISC-V arrives

RISC-V is an open instruction set, and two programs bring it to flight from different directions. NOEL-V, from the maker of the GR740, is a processor written in VHDL, from 32 bit to 64 bit with the hypervisor extension. Its maker ships an open-source version in its GRLIB library without fault tolerance, and sells a commercial version that adds error correction and scrubbing of its caches; it calls the core "flight proven in Earth orbit". That is the maker's statement, without a mission named on the page read.

HPSC comes from NASA. NASA's page dates the project from 2021: critical design review in 2024, tape-out in mid 2025, the first processors from the foundry later in 2025, and the first email in February 2026. The page says HPSC "delivers over 100 times the computing capability of current space processors", that its design adds "significant fault tolerance, radiation tolerance, and a full security suite" to an open instruction set, and that it "will be commercially available from Microchip". The 100 times is a design target from the contract and a claim on the agency's page, not a benchmark this book has seen. On 27 October 2026 Microchip reported early radiation results for the PIC64-HPSC1000-RH, in a release republished by a financial news site because the maker's own site could not be read: a total ionizing dose of 200 krad(Si), latch-up tested to 78.2 MeV cm2/mg, no uncorrected errors or external resets in single event testing "to date", and kits available to selected early access customers only. As of NASA's last status on its page, March 2026, HPSC was not declared space qualified.

KALPANA3201, above, shows the other road: an established open architecture, SPARC V8, built in a national fab. Each road trades differently. An open architecture gives a buyer more than one source of tools and software; a national fab gives control of supply; neither gives a flight record on the day it is announced. Figure 5.2 sets the processors of this chapter side by side, with the same columns for each and an empty cell wherever the sources read do not establish a value.

FIGURE 5.2

## Processors side by side, with the same columns and the maker's own figures

| Processor (maker) | Architecture, cores, clock as stated | Dose, krad(Si) | Latch-up LET as stated, MeV cm2/mg | Qualification or status, date | Whose figures, read | Documented use in the sources read |
|---|---|---|---|---|---|---|
| RCA 1802 (RCA) | CMOS; 5 microsecond cycle; about 30 mW | | | | NASA history (1988) | Galileo: six in the command and data subsystem |
| RAD750 (BAE Systems) | PowerPC 750 based; 115 to 132 MHz on Goddard missions; a 200 MHz version listed | | | | BAE page (2026-11-22); NASA Goddard (2015) | More than 100 satellites and the GPS III navigation payload, in BAE's words; GPM, SDO, the Webb instrument module |
| RAD5545 (BAE Systems) | A board: multi-core SpaceVPX single board computer that can host software-defined radio payloads (BAE) | | | | BAE page (2026-11-22) | No flight named |
| GR712RC (Frontgrade Gaisler) | Dual-core LEON3FT, 100 MHz | | | | NASA Goddard (2015) | |
| GR740 (Frontgrade Gaisler) | Quad-core LEON4FT SPARC V8, 250 MHz | 300 | 125 (immune to, maker) | QML Class V and Q, ceramic; in the European Preferred Parts List | Maker page (2026-11-23) | Three planned for the Roman telescope, per the RTEMS project |
| NOEL-V (Frontgrade Gaisler) | RISC-V soft core, 32 to 64 bit; fault tolerant in the commercial version | | | | Maker page (2026-11-22) | "Flight proven in Earth orbit", in the maker's words |
| PIC64-HPSC1000-RH (Microchip, for NASA) | RISC-V, multi-core | 200 | 78.2 (tested to, maker) | Early results 27 Oct 2026; not space qualified as of Mar 2026 (NASA) | Maker release, republished (trade press); NASA page (2026-11-23) | |
| VIKRAM3201 (ISRO with SCL) | 32 bit, custom instruction set, 180 nm CMOS | | | Qualified for launch vehicles; handed over 5 Mar 2025 | ISRO page (2026-11-23) | POEM-4 on PSLV-C60, a launch vehicle module |
| KALPANA3201 (ISRO with SCL) | 32 bit SPARC V8 | | | Handed over 5 Mar 2025 | ISRO page (2026-11-23) | Tested with flight software; no flight named |
| SOI-SoC SH-4 (Japan) | SuperH SH-4 on silicon on insulator | | | | Trans. JSASS papers (2014, 2021) | Hodoyoshi-3 and 4; PROCYON |

Every figure is the statement of the source named in its cell (maker, agency, project or paper), as read, with its date; a maker's figure is an operator claim; no source read gives the dose rate, lot, test date or failure criterion behind any dose or latch-up figure, so the figures do not rank the parts for a mission (chapter 3); an empty cell means not established in the sources read.

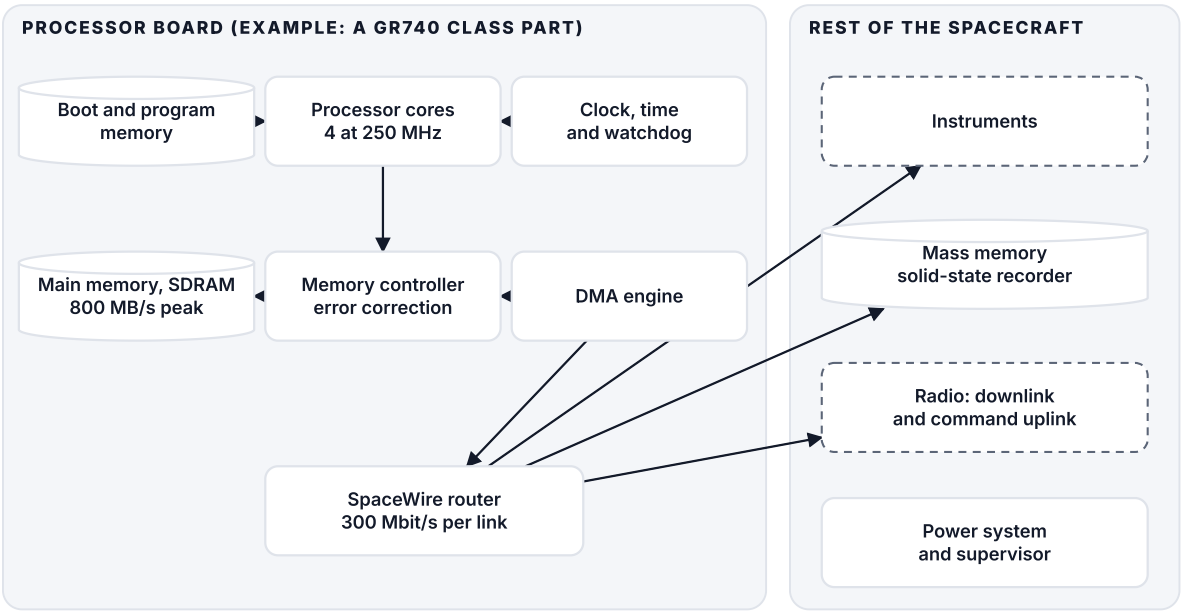
*Source: as named in each row; NASA, Computers in Spaceflight (1988); D. McComas, NASA Goddard, Lessons from 30 Years of Flight Software (2015); Microchip release of 2026-10-27 as republished by Finviz (trade press).*

## The avionics data path

A processor's clock says little about the work a flight computer does. The work arrives through memory, storage and buses, and figure 5.3 draws the parts around the processor that decide it.

FIGURE 5.3

### The avionics data path: the processor is one of several rates, and the smallest decides




Work reaches the processor through memory, storage and links, each with its own rate, set by hardware fixed at launch and by operating conditions; the power system and its supervisor can restart the board from outside it; the example figures are a GR740 class part's, as its maker states them.

*Author's diagram based on Frontgrade Gaisler, GR740 product page (read 2026-11-23); the 800 MB/s peak is calculated from its 64 bit PC100 interface; the DMA path follows NASA, Computers in Spaceflight, ch. 6.*

- **Main memory.** The GR740 reaches its SDRAM through one 64 bit PC100 interface. Its peak transfer rate is 64 bits times 100 million transfers a second, 800 megabytes a second in decimal units (calculated); correction handling, refresh, bus contention among four cores and the access pattern all take from that peak, and no measured figure was read. Four cores at 250 MHz share it.
- **Boot and program memory.** Non-volatile memory holds the first code that runs after a reset. The GR740's maker offers a boot loader it calls "flight qualified" (the qualification basis is not on the page), which shows that the first step after a restart is itself flight

software.

- **Mass memory.** Instruments fill a solid-state recorder between contacts with the ground: 4 GB on GPM in the Goddard table, against 36 MB of processor memory. The store and the downlink bound what can be buffered and sent; the processor's throughput and how much it reduces the data can set a further limit.
- **Direct memory access.** Voyager's data computer shows why this matters. NASA's history records that a study showed its processor speed would have to double, and that one fix was direct memory access on each instruction cycle, so that data "could be sent directly to the memory without having to go through the central processor", with every instruction taking the same time so the memory access rate could be guaranteed. Galileo's distributed computers moved all their messages the same way. The whole Voyager data system drew 14 watts; its computer, a third of a watt.
- **Buses and links.** The GR740 carries an eight-port SpaceWire router at 300 Mbit/s per link, with support for time synchronization; SpaceFibre is the faster serial link in the same family of European standards (ECSS-E-ST-50-11C, as named in a 2022 preprint). The editions and dates of the SpaceWire and SpaceFibre standards were not read.
- **Clocks, time and power supervision.** A watchdog, a reset path and a time source may be on the chip, on the board or in an independent unit. A supervisor that shares the processor's power can still reset a hung processor; only one independent of a failed clock or supply can recover from that failure, and only one that controls the supply can power cycle the processor (chapter 2's heartbeat, chapter 4's supervisor). Chapter 4 follows which memories carry error correction and what scrubbing repairs.

For a stated workload, each of these parts sets a limit that can be written in the same unit, for example images processed or delivered per orbit, once data reuse and compression are counted; the smallest of those limits bounds the work done. The downlink can limit what is delivered without limiting what is computed on board. The hardware sets the envelope on launch day; commands can change operating modes and allocations within it, software can rearrange the work, and parts can degrade.

### TRADE-OFF: A HARDENED PROCESSOR AND A PHONE-CLASS ONE ON THE SAME MISSION

**The case.** NASA's Perseverance rover runs on a RAD750, according to a NASA software repository and a 2023 policy study, which describes it as "a reengineered and radiation-hardened version of the PowerPC 750 processor that was released in 1997" with "very low compute capacity". NASA's Ingenuity helicopter on Mars used a Qualcomm Snapdragon, a commercial system on chip, according to the same repository; a 2023 peer-reviewed paper names the Snapdragon 801 and cites a 2022 JPL benchmarking study of Snapdragon parts for space use. **What it shows, and no more.** One mission flew both: the rover's computer on a hardened part with a long flight record, the helicopter's on a commercial part designed for phones. Both computers rely on their system design and software; what the part choices establish is a contrast in radiation assurance and in role, not that all the dependability sits in one place. Neither source says that either choice failed, and this book does not compare their performance, because no measurement of both on the same work was found. **The point for a buyer.** "Is the old processor still the right answer?" has no single answer: it depends on which computer must not stop, and what else carries the mission if it does.

### SECURITY NOTE

**Who can interfere or command.** A processor family that flies for decades carries a software base for decades: a compiler, an operating system, a boot loader and flight applications. Whoever can still build and sign software for that family, the mission's operator and its suppliers, controls what the computer runs; no document cited here says how any of the missions above authenticate a software upload, and this book does not assume it. A processor's own protections differ by family and are not in the pages read for most rows of figure 5.2: whether payload code and platform code run in separate protected memory has to be read, per family, in its documents. HPSC's page names "a full security suite" without detail; that is a claim, not a description. **How it is seen.** By the computer's own telemetry: restarts, error-correction counts and watchdog events, sent to the ground like any other data. **How it is recovered.** By the boot path: a protected first image in non-volatile memory, a watchdog that forces a restart, and a ground station that can reload the software. A processor with a flight-qualified boot loader carries that path as a product; a new processor has to build it before it has a flight record.

**Monday rule:** for any flight processor, write three dates (architecture new on the ground, part qualified, earliest documented flight found) and the memory bandwidth it actually gets.

CHAPTER SIX

# Part Choice and Qualification Evidence: A Matrix, Not a Ladder




On 19 November 2026 Microchip announced that one of its radiation-tolerant FPGAs, the RT PolarFire RTPF500ZT, "has achieved Defense Logistics Agency (DLA) Qualified Manufacturers List (QML) Class V qualification", and in the same release called it "flight proven". The release also said that a second device, the RT PolarFire SoC RTPFS460ZT, had completed "QML Class Q and MIL-STD-883 Class B qualifications". Three labels, two parts, one date. A buyer who reads those sentences as three rungs on one ladder, Class B below Class Q below Class V, with "flight proven" at the top, will choose badly. Each label answers a different question, and none of them answers the question a mission most needs answered: will this part, in this orbit, for this long, in this mode, do its job? This chapter separates the questions, shows what each kind of claim actually rests on, and ends with a matrix a reader can fill for one function bought several ways.

## Six axes, not one ladder

"Space grade" sounds like one property. In the documents makers publish it is at least six, and they vary independently.

- **Hardening method.** Whether the part was designed to resist radiation (hardened by design), built on a process chosen for it (hardened by process), or neither. AMD's March 2025 product table for its space FPGAs lists the Virtex-5QV as "Radiation Hardened By Design" and the Kintex UltraScale XQRKU060 as "Radiation Tolerant".
- **Radiation performance.** A total dose figure and single-event figures, each meaningful only with its test conditions. The same table gives the XQRKU060 a total ionizing dose of 120 krad and, in its row headed "SEL Immunity", a linear energy transfer (LET) of 80 MeV cm2/mg; for the Virtex-5QV it gives 1,000 krad and ">125". The table gives no test conditions for those rows. Chapter 3 explains the units.

- **Package.** Ceramic, plastic, or an organic substrate. AMD puts the XQRKU060 in a ceramic column grid array and its space-grade Versal XQRVC1902 in an organic ball grid array; Frontgrade sells its UT24CP1008 FPGA in a 484-ball plastic package.
- **Quality and screening flow.** What every delivered unit went through: burn-in, electrical tests at temperature, inspection, serialization. A flow is a manufacturing promise, not a radiation figure.
- **Temperature grade.** The junction temperature range the part is specified for. In AMD's grade table, industrial parts cover minus 40 to plus 100 degrees Celsius at the junction and military-temperature parts minus 55 to plus 125.
- **Mission assurance.** What the mission itself adds: its own radiation analysis, derating, lot tests, and the acceptance decision. No part label supplies this.

A part can be strong on one axis and weak on another. A radiation-hardened die in a part screened to a short flow, or a part screened to a demanding flow with no radiation data at all, are both possible. Figure 6.1 puts each axis beside an example from a maker's own document and the question that example leaves open.

FIGURE 6.1

## Six axes of a space part, each with a maker's example and the question it leaves open

| Axis | The question it answers | Example from a maker's document | What the example leaves open |
|---|---|---|---|
| Hardening method | Designed against radiation, built on a chosen process, or neither | Virtex-5QV "Radiation Hardened By Design"; XQRKU060 "Radiation Tolerant" (AMD, March 2025) | How the part behaves in the reader's orbit |
| Radiation performance | Dose and single-event figures, with conditions | XQRKU060: 120 krad; 80 MeV cm2/mg in the row headed "SEL Immunity" (AMD, March 2025) | Lot, bias, temperature, failure criterion, unless the test report is read |
| Package | Ceramic, plastic or organic substrate | XQRKU060 ceramic column grid array; XQRVC1902 organic ball grid array (AMD); UT24CP1008 484-ball plastic BGA (Frontgrade) | Which screening the package allows, and the mission length the maker supports |
| Quality and screening flow | What every delivered unit went through | "B-Flow (QML-Q Equivalent)" (AMD); "Space PEM" QD, L1, L2 (Frontgrade) | Radiation: AMD's Class B flow marks its total ionizing dose group "N/A" |
| Temperature grade | The specified operating junction temperature range | Industrial, minus 40 to plus 100 C; military, minus 55 to plus 125 C (AMD grade table) | The board's own thermal design |
| Mission assurance | What the mission adds: analysis, derating, lot tests, acceptance | Versal Class B: "Up to 7-year mission duration today" (AMD) | The mission's own decision, and who signs it |

A part can be strong on one axis and weak on another; no single label covers all six.

*Source: AMD, XQR Versal Adaptive SoCs, ESA SEFUW, 25 March 2025, slides 6, 11, 14, 17, 19 and 24; Frontgrade, UT24CP1008 product page, read 2026-11-23.*

## What the labels certify

Four families of label appear on space parts, and they come from different bodies.

**The QML classes.** The Qualified Manufacturers List is kept by the US Defense Logistics Agency under the performance specification MIL-PRF-38535, and makers quote three of its classes, Q, V and Y. The specification's own definitions of the letters are not quoted in this book, because its text could not be read for it; a buyer reads them there. What makers' documents show is enough to make the main point. Microchip's release names the agency and the list. And makers also run their own flows modeled on QML without claiming a listing: AMD's March 2025 table describes its Virtex-5QV as "V-Flow (QML-V Equivalent)", the XQRKU060 as "B-Flow (QML-Q Equivalent)" and "Y-Flow (QML-Y Equivalent)", and its Versal space parts as "Qualified and screened to MIL-PRF-38535 Class B, modified for organic packages", good for "Up to 7-year mission duration today", while it was "Evaluating package enhancements for 15-year missions and class Y qualification". "Equivalent" and "modified" are the maker's words, and they matter: a flow modeled on a QML class is not a listing on the agency's Qualified Manufacturers List, which only the agency's own list shows.

**MIL-STD-883.** The US test method standard for microcircuits, from which the test methods inside these flows are drawn: AMD's qualification tables head a column "MIL-STD-883" and list its test methods by number, among them 5005, 1005, 1008 and 1010. Microchip's November 2026 release names "QML Class Q" and "MIL-STD-883 Class B" as two labels for one part, so a reader keeps them apart too: how the two relate is set by the documents themselves, which were not read for this book. Neither the release nor AMD's deck says which document defines that "Class B" flow; a buyer asks the maker to name it. AMD's summary for the XQRKU060 lists the groups it ran on three qualification lots, "per MIL-PRF-38535": Group A functional and parametric tests, Group B assembly monitors, Group C an operating test of 2,000 hours, Group D, and Group E, which the deck's flow table names total ionizing dose.

**ESCC.** The European Space Components Coordination system publishes its own specifications. In January 2026 NanoXplore and STMicroelectronics announced that their NG-ULTRA FPGA was the first product qualified to ESCC 9030, which their release describes as "a new European standard dedicated to high-performance micro-circuits in flip-chip'ed on organic substrate or plastic package". The standard's own text was not read; that is the makers' description, and the "first" is theirs.

**Maker flows for plastic parts.** Some makers sell plastic-packaged parts with their own space screening. Frontgrade lists its UT24CP1008 in "Space PEM" flows, PEM meaning plastic encapsulated microcircuit, with QD as the base flow and L1 and L2 as options; the product page names the flows without defining them, so a buyer asks for the flow document.

None of these labels is a radiation analysis for a mission. AMD's comparison of the Class B flow of MIL-PRF-38535 with its own organic Class B flow, in the same deck, marks the total ionizing dose group "N/A" in both columns: screening is applied to every delivered unit, while the qualification and periodic groups are run on samples (the deck's qualification tables give lots and unit counts, and its flow table marks Groups C and D "Periodic"), and the radiation data comes from separate tests. Where a radiation test is part of a qualification, it is run at a dose and a dose rate set before anyone knew the reader's orbit: AMD's Versal dose test was run with gamma rays at under 18 krad per minute, and Frontgrade quotes its 100 krad(Si) figure at an effective dose rate of 1 rad(Si) per second. Chapter 3 shows how a mission turns its orbit, shielding and duration into a requirement; the label tells the buyer which screens each unit passed and which tests the samples behind it passed, and the analysis decides whether those tests are the right ones.

## Four kinds of support for a claim

Every sentence about a part rests on one of four kinds of support, and they are not interchangeable.

- **A maker's statement.** The maker says it. AMD's March 2025 product table summarizes the radiation behavior of its space Versal parts as "NO SEL, 100% Correctable SEUs, Ultra-low SEFI".
- **A ground test report.** Someone exposed named parts to a named source under named conditions. The full report is what a buyer reads; a few slides later the same deck gives the maker's summary of the conditions behind its headline: 20 VC1902 parts from 5 wafer lots, zero latch-up events at maximum supply voltage and junction temperature at an LET up to 80 MeV cm2/mg, and a dose test passed at 120 krad(Si) with gamma rays at under 18 krad per minute. The upset rates it quotes for a 500 km orbit at 20 degrees inclination and for geostationary orbit are "estimates based on CREME96 AP8-Max", a model run, not flight counts.
- **Flight data.** Units flew and someone reported what happened; a maker's listing of flight use is the weakest form of it. A small-satellite computer built on an AMD Zynq-7020 is listed by its maker as "in orbit since June 2016". The listing reports flight use from that date, with no operator record cited; it does not say in which orbit, in which mode, or what was reset along the way.
- **A certification.** An independent body lists the part. Microchip's announcement that the RTPF500ZT "has achieved" QML Class V is the maker's statement that the agency granted it; the buyer confirms it in the agency's own list, which was not read for this book.

"Tested to 25 krad" and "achieved QML Class V" are different sentences, and so are "no latch-up" and "no latch-up observed in 20 parts up to an LET of 80 at maximum voltage and temperature". The second sentence of each pair carries its own limits; the first carries them only if the reader goes and finds them. Figure 6.2 sets the four kinds side by side, with what each shows and what to ask next. An empty cell in such a table means that the sources read do not establish the point, not that the part is worse.

FIGURE 6.2

**Four kinds of support for a claim about a part, with one real sentence each**

| Kind | Example sentence (source) | What it shows | What to ask next |
|---|---|---|---|
| Maker's statement | "NO SEL, 100% Correctable SEUs, Ultra-low SEFI" (AMD, space Versal product table, March 2025) | The maker's summary of its own tests | Under which conditions, for which regions of the device |
| Ground test report (here, only the maker's summary; the full report was not read) | 20 VC1902 parts from 5 wafer lots; zero latch-up at LET up to 80 MeV cm2/mg at maximum supply voltage and junction temperature (AMD, same deck) | Behavior of those parts under those conditions, as the maker reports it | The full report and its failure criterion; the rate in the reader's orbit |
| Flight data | "In orbit since June 2016" (Xiphos Q7 listing, read November 2026) | That the maker reports flight from that date; no operator record cited | Which orbit, which mode, what was reset, what failed |
| Certification | RTPF500ZT "has achieved" DLA QML Class V (Microchip release, 19 November 2026) | The maker's statement that the agency granted the class | The agency's own list; the radiation analysis for the reader's mission |

Each kind shows something different, and none of them is the mission's own radiation analysis; an empty cell elsewhere in this book means not established in the sources read.

*Source: AMD, XQR Versal Adaptive SoCs, ESA SEFUW, 25 March 2025, slides 15 and 30; satsearch listing of the Xiphos Q7, read 2026-11-22; Microchip release of 19 November 2026 as republished by GlobeNewswire.*

### "Flight proven"

The phrase is common in makers' material. Microchip's November 2026 release calls the RTPF500ZT "flight proven"; Frontgrade Gaisler calls its NOEL-V processor core "flight proven in Earth orbit"; AMD's March 2025 deck shows a slide of missions using its parts, among them Virtex-5 devices for on-board computing on NASA's Artemis and Orion program and Virtex-5 and Virtex-4 devices in science instruments on ISRO's Chandrayaan-3. None of those sentences, in the documents read, names the lot, the orbit, the hours, the operating mode or the anomalies.

A flight record is real support, and it is narrow. It shows that some units of that part, from some lot, in some design, flew in some orbit, and, where the operator reports it, that they worked for some time in the mode they were used in. It does not show that another lot, a different process revision, a different orbit or a mode the earlier mission never used will behave the same; and a flight with no published failure is not a flight with no failure. The useful version of the phrase is a sentence with its parts filled in: which mission, which orbit, since when, in which role, and what the operator reported. Where a maker cannot fill them in publicly, the buyer asks under agreement and records the answer with its date.

Qualification ages too. A qualification belongs to a part made by a given process at given sites, and makers announce changes to those. AMD's deck shows the Virtex-4QV under "Last Time Buy", with a product discontinuation notice issued in October 2021, and records which wafer fab and assembly site its Versal space parts use. A program that buys late, or buys twice, checks that the part it receives is still the part that was qualified.

## Mission classes and the part list

Space agencies grade missions by how much risk they will accept, and the grade shapes the parts program: a flagship that cannot be repeated asks for more assurance than a technology demonstration that can. NASA's survey of small spacecraft avionics, dated 19 May 2026 and updated 21 October 2026, puts the practical side plainly: "For missions in harsher environments, spaceflight heritage and demonstrated radiation performance become primary differentiators when selecting electronics", while "experimental, high-risk missions typically focus on low-cost, easy-to-develop systems". NASA's formal risk classes for payloads, and the parts requirements tied to them, could not be read for this book and are not described here; a reader with an agency program reads them directly.

Two points hold whatever the class. First, a mission class does not choose a part. It sets how much assurance the program must show and how much risk it may accept; the part follows from the orbit, the function, the criticality of that function and the money. A risk-tolerant mission can still need a hardened part for one function that must not fail, and a demanding mission can fly a commercial part in a role where its failure is contained. Second, the class

sets who must agree. In a high-priority program, under the parts rules of its agency or project, a part outside the approved list typically needs a waiver, an analysis and someone who signs; in a low-priority one the same choice may rest with the project. Either way the decision should leave a record that names the part, the support for it and the person who accepted the risk.

## The commercial road

Many small satellites fly computers built on commercial parts, protected at the level of the board and the system rather than the die. The same May 2026 survey calls "COTS components first" the traditional default in small satellites, "combined with rad-hardened supporting electronics such as ECC, watchdog timers, scrubbing, and redundancy", and states the trade in two halves: commercial off-the-shelf parts "typically offer superior performance, energy efficiency, and affordability", but "tend to be highly susceptible to radiation". The same survey's table of on-board computers shows how thin a public figure can be: one entry gives "Estimated TID: 30krad (not tested)", and the survey warns that the performance advertised has not been independently verified by NASA. Three products show the range, each in its maker's words as listed in November 2026.

- **A Zynq-7020 computer.** Xiphos lists its Q7 with an AMD Zynq-7020 (two Arm Cortex-A9 cores and programmable logic), "in orbit since June 2016", tested to 25 krad total dose, at about 2 W.
- **A Zynq UltraScale+ computer.** Xiphos lists its Q8 with a Zynq UltraScale+ MPSoC, LPDDR4 memory protected by error detection and correction (EDAC), at 4 to 25 W depending on use.
- **An x86 processor beside a flash FPGA.** Unibap lists its iX10-101 with an AMD Ryzen V1000 processor, a Microchip PolarFire FPGA and an optional Intel Movidius Myriad X vision accelerator, at 20 to 40 W.

The listings name device families, not the grade or lot of the device inside, and the dose figure for the Q7 does not say who tested it or how. A buyer asks for the exact part number, which tells the grade.

The device makers themselves draw a line. In its March 2025 space presentation AMD listed the 16 nm Zynq UltraScale+ MPSoC and RFSoC as "Defense Grade" with "Potential for proton based missions", adding "details shared with NDA". In AMD's grade table, defense grade adds ruggedized packages, testing at temperature extremes, a subset of MIL-STD-883 tests and mask set control; space grade adds radiation test reports, burn-in and the full qualification groups. "Potential for proton based missions" is the maker's phrase; the data

behind it is not public, and the phrase names no orbit. Chapter 3 shows that low orbits also see heavy ions and cross the South Atlantic Anomaly, so the phrase is a starting point for a test plan, not a conclusion.

The commercial road is defensible when four things are true and written down: the mission's dose over its life is below what the part has been shown to survive, with margin, on a test of the same part and with the spread between lots allowed for; destructive single events, latch-up above all, have been tested for, or latch-up is guarded against by a circuit shown to cut the power faster than the damage, while burnout and gate rupture in power parts are tested for and derated, since no power cut prevents them; the design detects and recovers from single events and functional interrupts fast enough for the job, with a supervisor that is itself more robust than what it watches (chapter 4); and the consequence of the part failing outright is acceptable for this mission, or contained by redundancy. It is not defensible as a general statement. "Commercial parts work fine in low orbit" and "only hardened parts can fly" are both claims without an orbit, a duration or a function attached.

## The decision matrix

The matrix below compares one function, a reconfigurable processing device for a payload, bought four ways, with each cell filled from a public document of its date. It is the worked page of this chapter: a reader replaces the rows with the parts on their own shortlist and fills every cell from a document, or writes "not established in the sources read". Lead times and prices are not filled in because no dated public source for them was found; a buyer obtains them as dated quotes. The blank form is in Appendix A.

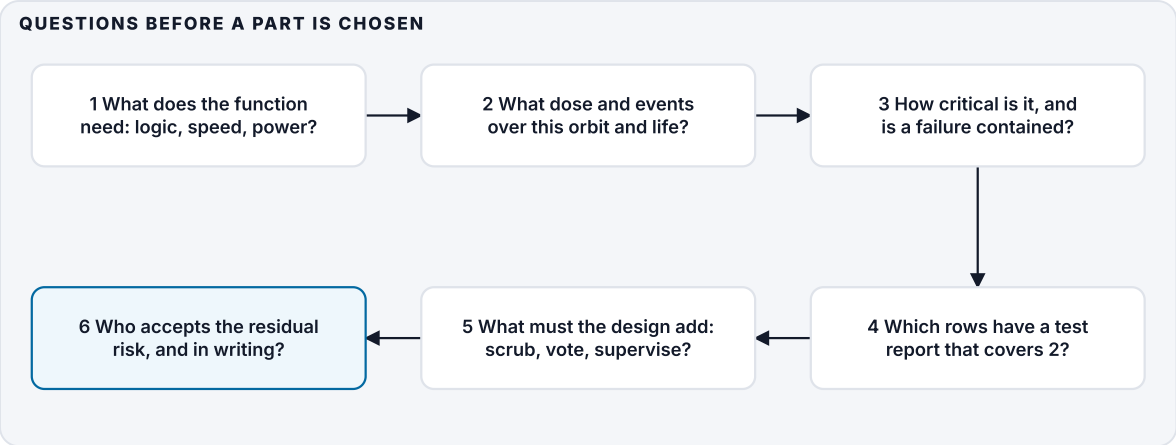
### The decision matrix: one function, four ways to buy it (public documents, as dated)

| Approach | Example | Dose, and who says so | Single events, and who says so | Package and flow | Mission length the maker names | What the buyer still supplies |
|---|---|---|---|---|---|---|
| Hardened by design | Virtex-5QV, 131,072 logic cells | 1,000 krad; maker's table, March 2025 | ">125" in the maker's latch-up immunity row; no conditions given | V-Flow, "QML-V Equivalent"; maker | Not stated in the deck | Rate analysis for the orbit; configuration and user-state mitigation (chapter 7) |
| Radiation tolerant, maker's space flow | XQRKU060, 725,550 logic cells | 120 krad; maker's table | 80 in the same row; no conditions given | Ceramic; B-Flow and Y-Flow, "QML-Q" and "QML-Y Equivalent"; shipping since September 2020 | Not stated in the deck | Scrubbing, mitigation of the design, rate analysis |
| Newer process, space flow (two examples) | UT24CP1008, 96,000 logic cells | 100 krad(Si) at 1 rad(Si)/s (Frontgrade page) | "LET 80 or less", effect not named (Frontgrade) | Plastic BGA, Space PEM QD, L1, L2 | Not stated | The flow documents; the test report behind each figure |
| (the same approach) | NG-ULTRA | 50 krad(Si) (NanoXplore and ST release) | Latch-up immunity tested to 65 (release) | ESCC 9030 (release) | Not stated | The flow document; the test reports behind each figure |
| Commercial part, system mitigation | Zynq-7020 on a small-satellite board | "Tested to 25 krad"; board maker's listing, tester not named | Not established in the sources read | Not stated in the listing | Not stated; "in orbit since June 2016" | The exact part and lot; a radiation test or a cited one; a supervisor and recovery design; acceptance of the residual risk |

Read across a row, not down a column. The part AMD calls hardened by design has the highest dose figure and the least logic; the commercial board has the least public support behind any of its figures, and it is the only row that is a whole computer rather than a device. Which row is right depends on the inputs the matrix leaves to the reader: the orbit and its dose over the mission, the mission length, how critical the function is, the logic and power the function needs, and how much residual risk the program has agreed to carry. Take two hypothetical requirements. A design whose implementation, once built for the part, needs about 700,000 of AMD's logic cells' worth of lookup tables, registers and memory does not fit the Virtex-5QV's 131,072, whatever its dose figure. A function that must work for fifteen years in geostationary orbit cannot rest on a board whose only public dose figure is 25 krad with no tester named; it needs a dose analysis and test support of its own. Figure 6.3 turns the same reading into questions in order.

FIGURE 6.3

## The decision matrix read as questions, in order




Each question removes rows; the last box is a decision someone signs, not a property of the part.

*Author's diagram based on this chapter's decision matrix.*

The empty cells of the decision matrix, and the questions of figure 6.3, become one request to a supplier. It asks for:

- the exact ordering code, die revision, lots, the number of samples, and the sites that made and tested them;
- for dose: the bias during the test, the dose rate, any anneal, and the failure criterion;
- for single events: the ions or protons, their energy and LET, the angle, the fluence, the bias and temperature, what was monitored and how an error was defined;
- the counts of events, the cross-sections and their confidence, including the bound behind any zero (chapter 3);
- the recovery used after each kind of event, and which destructive effects were tested;
- how the test conditions map to the mission's, whether the results apply to the configuration and lot being delivered, and the listing or flow document behind any class letter, with the supplier's duty to notify changes.

Export rules apply to some radiation-hardened parts and to the technical data that comes with them, and they differ by country. Who may buy, ship and receive such parts is the subject of the fourth book of this series, *Who Owns the Sky*, and a question for counsel, not for this matrix.

### TRADE-OFF: CAPACITY AGAINST MARGIN, IN ONE MAKER'S TABLE

No in-flight part failure could be traced to a primary record for this chapter, so a documented trade-off stands in its place: one maker's own range, in its own March 2025 table. AMD's hardened-by-design Virtex-5QV offers 1,000 krad and a latch-up row of ">125" with 131,072 logic cells; its radiation-tolerant XQRKU060 offers 725,550 logic cells, about five and a half times as many (725,550 divided by 131,072) in AMD's own logic cells, an equivalent count whose factor is the maker's (another maker computes its own as lookup tables times 1.2), so fit is shown by building the design, in lookup tables, registers, memory and arithmetic blocks, and not by the ratio, at 120 krad and a latch-up row of 80. The same table lists both as shipping, and so writes down what each costs: capacity on one side, higher dose and latch-up ratings on the other, each a test envelope without stated conditions rather than a mission margin. A buyer who needs both the capacity and the higher ratings cannot buy them in one part from that table, and must either accept the lower ratings and analyze them for the orbit, split the function across parts, or add shielding against dose and mitigation against upsets, paying for them in mass, power and design effort; neither shielding nor mitigation restores the latch-up rating the table gives up. In this range, then, a part that was better on one axis was worse on another, and the maker said so in its own figures.

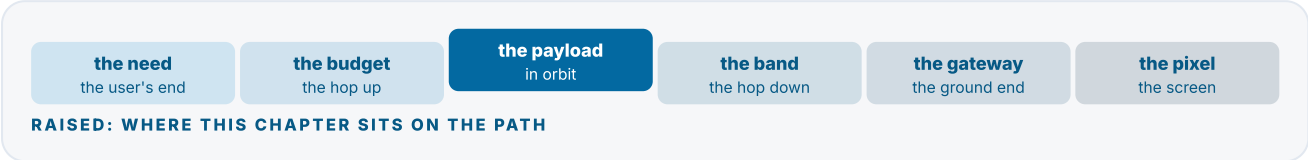
### SECURITY NOTE

**Who can interfere or command.** A part's supply chain is a path into the satellite before launch. Whoever can substitute a part, by selling a counterfeit, shipping a different lot or changing a process or site without notice, can change how the computer behaves in orbit. A commercial part also brings features its buyer may not need and cannot always see, such as test and debug interfaces; in a computer that runs other parties' code, an interface that is documented only under agreement is a question to ask, not a detail. **How it is seen.** By traceability: serial numbers on each unit, a lot record, the wafer fab and assembly sites, and a change notice when any of them moves. AMD's deck records serialization in its space flow and names the fab and assembly sites of its Versal space parts; NanoXplore and ST describe a supply chain "fully based in the European Union, from design to manufacturing and test". A buyer checks that the delivered part matches its records, and runs incoming inspection and tests on a sample. **How it is recovered.** Before launch, by quarantining a suspect lot and replacing it. After launch, only by design: a function that can be moved to a redundant unit, or a supervisor that can isolate a part behaving wrongly. Detecting a suspect part, diagnosing whether it is a bad lot or a radiation effect, and attributing it to a supplier are separate steps: detection and the logs that help diagnosis can be designed in advance, while attribution usually needs the supplier's records and an investigation.

**Monday rule:** ask for the test report, not the adjective: which part and lot, which test, which dose or energy, which failure criterion, and which date.

CHAPTER SEVEN

# Space FPGAs: Where the Configuration Lives, What Protects the Rest, and the Process Under It




On 28 January 2026 NanoXplore and STMicroelectronics announced that their NG-ULTRA FPGA had been qualified to a new European standard for space parts. Two sentences of their release, read side by side, hold most of this chapter. One says that NG-ULTRA "is built on STMicroelectronics' 28nm FD-SOI digital technology platform". The other says that "the SRAM-based architecture of the NG-ULTRA enables an adaptive hardware approach", with "unlimited on-orbit reconfiguration". FD-SOI is the process the transistors are made on; SRAM is where the chip keeps the bits that define the circuit. A buyer who hears "an FD-SOI FPGA" and files it beside "an SRAM FPGA" and "a flash FPGA", as three choices of one kind, has mixed two questions, and has not yet asked the third: what protects the user's own flip-flops and memories, which can be upset whatever holds the configuration. This chapter keeps the three questions apart, places the 2026 families on them with the same fields for each, and ends with the tools that turn a design into the bits.

## Three questions, never mixed

A field programmable gate array (FPGA) is a chip whose circuit is defined by stored bits, the configuration, loaded after manufacture. In most of the families here its logic is made of small lookup tables, flip-flops, memory blocks and arithmetic blocks (antifuse parts use logic cells of their own), joined by programmable routing. In orbit three separate questions decide how it behaves.

- **Where is the configuration held?** In SRAM cells, which are written at power-up from an external memory and can be upset by a particle while the chip runs; in flash cells, which keep their state without power; or in antifuses, links that are made once,

physically, when the part is programmed. An upset in configuration changes a configuration bit rather than a data value, and so it can change the circuit itself, its logic, its routing or its settings, although not every such upset changes what the circuit does.

- **What protects the user's own state?** The flip-flops, memory blocks and clocks of the design can be upset by a particle whatever holds the configuration. They are protected by the chip's own hardened cells where the maker built them in, or by the designer: triplication with voting, error correcting codes on memory, filters against transients (chapter 4).
- **What process is it made on?** Bulk CMOS at 65, 20, 16 or 7 nm, or 28 nm fully depleted silicon on insulator (FD-SOI), each with its own dose and latch-up behavior, which only a test shows.

Two engineers of NASA's Goddard Space Flight Center, M. Berg and K. LaBel, set out the first two questions in a presentation at the 2016 Single Event Effects Symposium. For the configuration, it says antifuse configuration "has been designated as hard regarding SEEs" (single-event effects); flash configuration "has been designated as hard (but NOT immune)"; the configuration of an SRAM device without hardening "has been designated as the most susceptible portion of circuitry"; and a hardened SRAM device is "hardened (but NOT hard)". For the user's state, it lists the flip-flops and clocks as the most susceptible parts of antifuse and flash devices, where the configuration is already hard; and for flash and hardened SRAM it adds that "Susceptibilities also exist in the data paths and global routes (e.g., clocks and resets)". A flash FPGA, in other words, is not a radiation-proof FPGA. It is an FPGA whose most sensitive part is no longer its configuration.

The third question, the process, is not a configuration choice. NG-ULTRA is an FD-SOI device with SRAM configuration. The buried insulator of FD-SOI changes the parasitic paths through the substrate that cause latch-up in bulk CMOS, which is one reason the process is chosen; but whether a given part resists latch-up, and up to what LET, is still a test result for that part, with its LET and conditions: NanoXplore and ST give latch-up immunity "tested up to 65 MeV·cm²/mg" for NG-ULTRA, and Frontgrade's page for its FD-SOI part lists an LET of "80 or less" without naming the effect. Figure 7.1 places the families of this chapter on the three questions.

FIGURE 7.1

**Three separate questions, with the families of this chapter placed on each**

| Family | Where the configuration is held | What protects the user's state | Process |
|---|---|---|---|
| Virtex-5QV (AMD) | SRAM, hardened: "hardened (but NOT hard)" (NASA, 2016) | Dual interlocked cell flip-flops and transient filters in silicon (NASA, 2016) | 65 nm |
| XQRKU060 (AMD) | SRAM, repaired by a scrubbing core in the user's logic (AMD) | The designer's mitigation; built-in protection not established in sources read | 20 nm |
| XQRVC1902 (AMD) | SRAM with error correction and interleaving, repaired by a scrubber on triplicated hardwired processors (AMD) | Not established in sources read | 7 nm |
| RT PolarFire (Microchip) | Nonvolatile, "designed to eliminate configuration SEU concerns" (Microchip); flash (Antunes and Podobas, preprint) | Not established in sources read | Not established in sources read |
| NG-ULTRA (NanoXplore, ST) | SRAM (makers' release) | Not established in sources read | 28 nm FD-SOI |
| UT24CP1008 (Frontgrade) | Not stated on the maker's page | Not established in sources read | 28 nm FD-SOI |

| Family | Where the configuration is held | What protects the user's state | Process |
|---|---|---|---|
| RTAX, RTSX (antifuse) | Antifuse, programmed once: "designated as hard" (NASA, 2016) | Flip-flops triplicated in silicon in rad-hard versions (Los Alamos, 2004) | Not established in sources read |

Configuration, user state and process are independent: an FD-SOI part can hold its configuration in SRAM, and a hard configuration leaves the flip-flops exposed.

*Source: Berg and LaBel, NASA GSFC, New Developments in Error Detection and Correction Strategies for Critical Applications, 2016 (NTRS 20160007971); Stettler and others, Los Alamos, 2004 (CERN-2004-010); AMD, ESA SEFUW, 25 March 2025; Microchip release, 19 November 2026; NanoXplore and ST release, 28 January 2026; Frontgrade UT24CP1008 page, read 2026-11-23; Antunes and Podobas, arXiv 2504.16173 (preprint).*

## The families side by side

Figure 7.2 gives the families of this chapter the same fields, before the sections that follow explain them maker by maker. Every cell gives the kind of source it comes from (a maker's document, an agency, a paper, a preprint or a tool project) with its date, or reads "n.e.", not established in the sources read, which says that this book found no public document for the point, not that the part is worse. Volatile cells, such as shipment and qualification status, are repeated with their dates in Appendix D.

FIGURE 7.2

## Six space FPGA families with identical fields, each cell from a dated public document

| Field | Virtex-5QV | XQRKU060 | XQRVC1902 | RT PolarFire RTPF500ZT | NG-ULTRA | UT24CP1008 |
|---|---|---|---|---|---|---|
| Configuration | SRAM, hardened | SRAM | SRAM, error correction and interleaving | Nonvolatile (maker); flash (preprint) | SRAM | Not stated on the page |
| User-state protection built in | Dual interlocked cell flip-flops, transient filters (NASA) | n.e. | n.e. | n.e. | n.e. | n.e. |
| Latch-up figure | ">125" in the maker's SEL immunity row; no conditions given | 80 in the same row; no conditions given | Zero events in 20 parts, LET up to 80, maximum voltage and temperature | No figure in the release | Tested up to LET 65 | LET "80 or less", effect not named |
| Dose figure | 1,000 krad | 120 krad | 120 krad(Si) test passed, gamma, under 18 krad/min | n.e. | Up to 50 krad(Si) | 100 krad(Si) at 1 rad(Si)/s |
| Process | 65 nm | 20 nm | 7 nm | n.e. | 28 nm FD-SOI | 28 nm FD-SOI |
| Capacity, maker's unit | 131,072 logic cells | 725,550 logic cells | 1,968K system logic cells (899,840 lookup tables); 400 AI engine tiles | n.e. | 537k lookup tables (release); 500K four-input (paper) | 96,000 logic cells (lookup tables x 1.2) |
| Tools | Maker's flow | Maker's flow | Maker's flow | Maker's flow | Impulse; nextpnr places and routes, bitstream needs Impulse | n.e. |
| Package | n.e. | Ceramic column grid array | Organic ball grid array | n.e. | n.e. | 484-ball plastic BGA |
| Assurance, with date | V-Flow, "QML-V Equivalent" (March 2025) | AMD Class B and Class Y passed; shipping since September 2020 | AMD Class B, completion announced 15 November 2022; Class Y under evaluation in March 2025 | QML Class V, announced 19 November 2026 | ESCC 9030, announced 28 January 2026 | Space PEM QD, L1, L2 (page read 23 November 2026) |

| Field | Virtex-5QV | XQRKU060 | XQRVC1902 | RT PolarFire RTPF500ZT | NG-ULTRA | UT24CP1008 |
|---|---|---|---|---|---|---|
| Use stated in sources read | Maker lists Virtex-5 parts on Artemis and Orion computing (family; exact device and mission not stated) | Maker lists Kintex UltraScale in a 2024 imaging constellation (family; grade not stated) | Maker lists Versal parts in constellations launched 2023 and 2024, grade not stated | "Flight proven" (maker), no flight named | "Set to be used" in Galileo, Copernicus (announced) | n.e. |

Each capacity is in its maker's own unit and they cannot be compared directly; "n.e." means not established in the sources read, not worse.

*Source: AMD, ESA SEFUW, 25 March 2025, slides 4, 6, 7, 11, 14, 15, 19 and 30; AMD release of 15 November 2022; NASA GSFC, Berg and LaBel, 2016; Microchip release, 19 November 2026; NanoXplore and ST release, 28 January 2026; Leon and others, ICECS 2024 (arXiv 2409.12253); nextpnr README, November 2026; Frontgrade UT24CP1008 page; Antunes and Podobas, arXiv 2504.16173 (preprint). Status cells give the date of their source; status in November 2026 was not checked. They are repeated with dates in Appendix D.*

Read the table across a row first. Within AMD's own units, capacity rises from the 65 nm part to the 20 nm part, while only the 65 nm part is called hardened by design; on the newer parts the protection of the configuration moves into a scrubber and into the user's design. The FD-SOI parts give lower dose figures than the hardened part, in plastic or organic packages with European or maker flows. No column is better than another until the reader brings an orbit, a mission length, a function and a decision about who carries the residual risk, which is chapter 6's matrix.

## Keeping SRAM configuration correct: scrubbing

An SRAM FPGA's configuration can be upset while the chip runs, so a design that must keep running needs a way to repair it: a full reload when an error is found or at set times, or a repair while the chip runs, which is called scrubbing. Berg and LaBel define it as "the act of simultaneously writing into FPGA configuration memory as the device's functional logic area is operating with the intent of correcting configuration memory bit errors", and note that it "only pertains to SRAM-based configuration devices". It is done two ways. The simple way writes a known-good copy back over the configuration at a fixed interval without looking at it first. The other way, which a 2004 Los Alamos paper describes as "the periodic readback of the FPGA's configuration memory, comparing it to a known good copy, and writing back any corrections required", reads the configuration back, finds the frames that differ, and rewrites only those. The Los Alamos authors note that Xilinx Virtex devices could read back and rewrite part of the configuration without resetting the device, so that "the uninterrupted operation of user logic" continued: the logic kept running, which is not the same as its state being right (chapter 4).

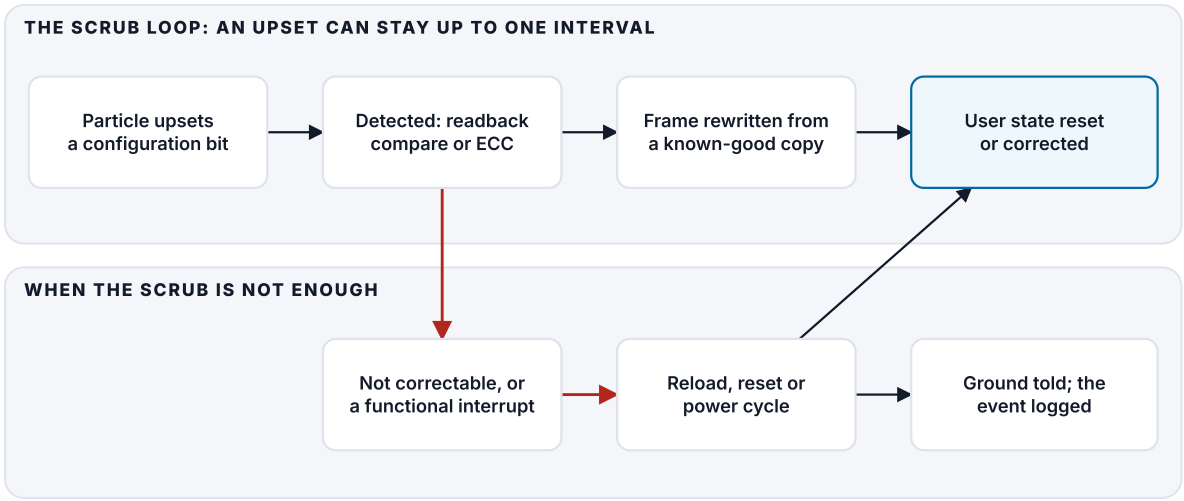
Three facts limit what scrubbing achieves.

- **The interval sets the exposure.** In the Los Alamos authors' words, the scrubbing frequency "defines the rate at which upsets can be detected", and combined with the rate of upsets it "provides the actual tolerance of the design". If the scrubber works and the upset is one it can correct, the upset stays in the circuit for up to one scrub interval; a second upset in that interval that hits another copy of the same voted logic can defeat triplication, and so can one upset in a resource the copies share.
- **Fixing the bit does not fix the state.** Berg and LaBel: "Fixing a configuration bit does not mean that you have fixed the state in the functional logic path." When the upset affected the design's function, they list two ways of reliably getting back to an expected state: repair the bit and reset or correct the flip-flops, or reconfigure the whole device. A design whose voters feed corrected values back into its own registers can do part of that work by itself.
- **The scrubber is a circuit too.** It runs somewhere, it reads a known-good copy from somewhere, and both can fail. The Los Alamos authors observed that even in SRAM FPGA systems "there is often an antifuse part tasked with scrubbing the SRAM devices": the repairer was put on the hardest configuration available.

Triplication and scrubbing are used together because each answers what the other leaves. Triplication masks one upset until the scrubber repairs it; scrubbing stops upsets from piling up until two copies are wrong. The Los Alamos paper of 2004 judged that "a proper TMR implementation combined with fast scrubbing can provide better than an order of magnitude increase in the radiation tolerance of a given design", and also warned that triplication assumes one upset per block, while tests on Virtex-II devices had found that about 0.3 to 0.5 percent of upsets were multiple-bit upsets. That is a 2004 judgment on 2004 parts, labeled as such; the principle, that the scrub interval and the upset rate together set the residual risk, has not changed. Figure 7.3 draws the loop.

FIGURE 7.3

### A configuration upset, from the particle to the circuit running correctly again




The repair has two halves, the configuration bit and the user's state; a fault the scrubber cannot repair goes to a full reload from a protected image, and then to a reset or a power cycle where the device needs one.

*Author's diagram based on Berg and LaBel, NASA GSFC, 2016 (scrubbing definition; repair of the user's state after a configuration fix) and Stettler and others, Los Alamos, 2004 (readback scrubbing).*

## AMD: SRAM configuration, from 65 to 7 nm

AMD's March 2025 presentation lists SRAM-configured space FPGAs, still under the Xilinx names Virtex and Kintex and with its flows still called "Xilinx Class B", on three processes. Every figure below is the maker's, from its presentation at ESA's Space FPGA Users Workshop of March 2025 unless another source is named.

- **Virtex-5QV.** 65 nm, "Radiation Hardened By Design", 131,072 logic cells, a total ionizing dose of 1,000 krad and ">125" MeV cm2/mg in the table's row headed "SEL Immunity", with no test conditions given there, listed as shipping. Berg and LaBel list its built-in mitigation as hardened configuration, "DICE DFFs" (flip-flops built from dual interlocked cells) and "SET filters", with clocks still the most susceptible part and "In some cases additional mitigation may be necessary for configuration and DFFs". Its predecessor, the 90 nm Virtex-4QV, is listed as "Last Time Buy".
- **Kintex UltraScale XQRKU060.** 20 nm, "Radiation Tolerant", 725,550 logic cells, 120 krad, 80 in the same latch-up row, shipping since September 2020, with AMD Class B and Class Y qualification passed. Its configuration is not hardened by design, and a 2025 survey by Antunes and Podobas (below) counts radiation-tolerant Kintex UltraScale parts

among the SRAM families that "still require configuration repair"; AMD's interrupt figures for it assume its soft error manager, a scrubbing core placed in the user's own logic.

- **Versal XQRVC1902.** 7 nm, an adaptive system on chip with 1,968 thousand "system logic cells" (899,840 lookup tables), 400 AI engine tiles, two Arm Cortex-A72 and two Cortex-R5F cores. On 15 November 2022 AMD announced that its Class B qualification was complete; in March 2025 Class Y, for 15-year missions, was still being evaluated. Its configuration memory is protected by error correction and interleaving, and repaired by a scrubber called XilSEM that runs on "Two triple-redundant hardwired MicroBlaze processors", so the repairer is itself triplicated in fixed silicon rather than built from the configuration it repairs.
- **Versal XQRVE2302.** A smaller 7 nm part with 150,272 lookup tables and 34 AI engine tiles, Class B qualification completed, "shipping 4Q2025" as a plan in March 2025. No later statement that it shipped was found.

AMD's summary line for the space Versal parts is "NO SEL, 100% Correctable SEUs, Ultra-low SEFI". This book prints that line only with the conditions the same deck gives for it. The latch-up result is "ZERO SEL events" in 20 parts from 5 wafer lots, at maximum supply voltage and junction temperature, at an LET up to 80 MeV cm2/mg. The upset rates are estimates from a model, CREME96 with the AP8-Max proton model, for a 500 km orbit at 20 degrees inclination and for geostationary orbit. "Correctable" refers to the configuration memory; the processors and other blocks still have functional interrupts, which AMD estimates at about 0.16 a year for the processor system in geostationary orbit, with processor core hangs and the network on chip as the main signatures. "Correctable" is not "no interrupts", and "no latch-up observed to LET 80" is not immunity outside the tested conditions.

## Microchip: nonvolatile configuration

Microchip sells space FPGAs whose configuration is held in nonvolatile cells. Its November 2026 release says that "RT PolarFire devices use configuration technology designed to eliminate configuration Single-Event Upset (SEU) concerns". The 2025 survey by Antunes and Podobas of KTH (a preprint, not peer reviewed) puts the consequence plainly: "flash-based families such as PolarFire largely avoid configuration SEUs and can rely on ASIC-like protection of user state, whereas SRAM families … still require configuration repair for high reliability". "Largely avoid" and "eliminate concerns" are not "immune"; NASA's 2016 wording, "hard (but NOT immune)", is the one to keep.

The release of 19 November 2026 says that the RT PolarFire RTPF500ZT "has achieved" QML Class V and that the RT PolarFire SoC RTPFS460ZT, which adds RISC-V processor cores, "has completed QML Class Q and MIL-STD-883 Class B qualifications" (chapter 6 explains why those are separate labels). It calls the RTPF500ZT "flight proven" without naming a flight in the text read; a flight is printed here only when a mission's own record names it. Microchip's older RTG4 and its antifuse RTAX family are named among the space-grade FPGAs of the market in a 2024 paper on NG-ULTRA by Leon and colleagues; their data sheets could not be read for this book, so figure 7.2 has no column for them.

### FD-SOI: two 28 nm families

Fully depleted silicon on insulator puts the transistors on a thin layer of silicon over a buried insulator. Two FPGA families on 28 nm FD-SOI were offered for space in 2026 in their makers' own documents, and they differ on the other two questions.

- **NG-ULTRA (NanoXplore, made by STMicroelectronics).** SRAM configuration, "537k LUTs + 32 Mb RAM", a quad-core Arm Cortex-R52 processor system, a total ionizing dose "of up to 50 krad (Si)" and latch-up immunity "tested up to 65 MeV·cm²/mg", in the makers' release. The release calls it "the first product qualified to ESCC 9030", a standard it describes as for "flip-chip'ed on organic substrate or plastic package" parts. It says NG-ULTRA "is set to be used" in equipment for Galileo, Copernicus "and potentially IRIS²": that is an announced use, not a flight. A 2024 paper by Leon and colleagues, funded in part by ESA, describes it as 500K four-input lookup tables with about 1.3K DSP blocks and a 600 MHz Cortex-R52 processor; neither document says why the two counts differ.
- **UT24CP1008 (Frontgrade, sold as CertusPro-NX-RT).** 28 nm FD-SOI, 96,000 logic cells, a total ionizing dose of 100 krad(Si) at an effective dose rate of 1 rad(Si) per second, an LET figure of "80 or less" with no effect named, a 484-ball plastic package, and Frontgrade's Space PEM screening flows. Its page lists "AES-256 bitstream encryption and ECDSA authentication". The page does not name the configuration memory.

The capacity numbers above are in three different units: AMD's logic cells, NanoXplore's lookup tables, and Frontgrade's logic cells, which its page says it computes as "LUTs × 1.2 effectiveness". They cannot be compared without converting them, and the conversion depends on the architecture. Figure 7.2 prints each in its maker's own unit, and says so.

## Antifuse, in one page

An antifuse FPGA is programmed once. The Los Alamos authors of 2004 described them as "one time programmable devices" that "use physical shorts between metal routing layers to configure their logic": the configuration is fixed in programmed links rather than held in memory cells, so there are no configuration cells for a particle to upset. NASA's 2016 table lists Microsemi's RTAX and RTSX families in this class (Microsemi is now part of Microchip), with flip-flops and clocks as their most susceptible components and combinatorial logic susceptible but low. The same 2004 paper described rad-hard antifuse versions "with internal flip flops TMRed in silicon", which, in its authors' words, "completely remove the need to TMR user designs". Built-in flip-flops answer the registers only; NASA's 2016 table still lists the clocks of antifuse devices among their most susceptible parts, so the rest of the design is analyzed all the same.

The cost is the other side of the same fact. The design cannot be changed after programming, so the Los Alamos authors judged antifuse "best suited for applications where the initial requirements are stable", and noted that antifuse parts "are not available in as high logic densities as SRAM devices". A mission that wants to change the circuit itself in orbit cannot use them for that function (software and settings held in memory beside the fabric can still change); a mission that wants a small, fixed, robust circuit, such as the supervisor or the scrubber of a larger device, often did, in their account. Which antifuse parts were still offered on the book's date was not checked, and no status is given here.

## Soft processors

An FPGA can carry a processor built from its own logic, a soft core. Chapter 4 described a small-satellite computer whose LEON3FT soft core runs at 50 MHz with triplication on every flip-flop of the FPGA. Gaisler's NOEL-V is a RISC-V soft core in VHDL. The maker distributes an open version under the GPL, in its GRLIB library, without fault tolerance; its commercial version adds error correction and scrubbing for the caches, and the maker calls it "flight proven in Earth orbit". The split matters for a buyer: the open version is a way to learn and to inspect; the maker's page does not say which version its flight claim refers to, so a buyer asks. A soft core inherits every upset behavior of the logic it is built from, which is why fault-tolerant soft cores protect their own registers and caches rather than trusting the FPGA beneath them.

## Tools: from source to bitstream

Each maker supplies its own tool flow, from a hardware description in Verilog or VHDL to the bitstream that is loaded into the part. The vendor flow is the one a maker supports for its space parts, and the one its qualification and timing data assume. For NG-ULTRA, the 2024

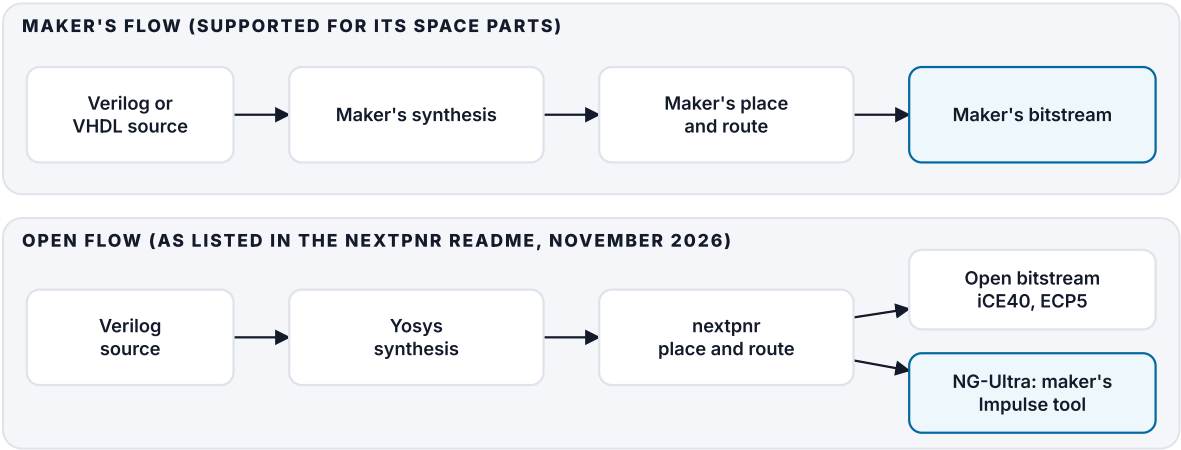
paper by Leon and colleagues used NanoXplore's Impulse tool.

An open flow also exists. Yosys, an open synthesis framework, documents technology libraries for many families, among them Lattice, Gowin, NanoXplore, Xilinx and Microchip's SmartFusion2 and IGLOO2. nextpnr, the open place and route tool that usually follows it, listed in its README, as read in November 2026, support for Lattice iCE40 and ECP5, Gowin, NanoXplore NG-Ultra and Cologne Chip GateMate among others, and marked Cyclone V, Lattice MachXO2 and Xilinx 7-series as "experimental". Two limits are written in the same README. NG-Ultra is placed and routed by nextpnr, but "binary bitstream creation requires Impulse tool from NanoXplore", so the open flow for that part ends at the vendor's tool. And neither AMD's space parts nor Microchip's PolarFire families appear in its list. "Supported" here means listed by the project; the README sets no criterion for flight use, and a release number pins down what was supported when.

What an open flow gives is inspection and repeatability: in a fully open flow every step can be read (for NG-Ultra the last step is the maker's), and with the source, the tool versions, the settings and the random seeds all pinned, a build can be repeated and checked by rebuilding and comparing the result, which matters to a program that must show what it flew. What it does not show is timing closure on a flight part across its temperature and dose range, the maker's support, or a mission's acceptance. Figure 7.4 draws the two flows.

FIGURE 7.4

## Two flows from source to bitstream, and where the open one stops for a space part




For NG-Ultra the open flow places and routes, but the bitstream is written by the maker's tool; AMD's space parts and Microchip's PolarFire families are not in the open flow's list.

*Author's diagram based on the nextpnr README (YosysHQ, library copy of November 2026), the Yosys documentation, and Leon and others, ICECS 2024 (Impulse as NanoXplore's tool).*

**WHAT THIS STEP REQUIRES**

The bench step of this series for FPGAs is stated as requirements, because no result exists yet. It requires: a commercial board with a part that an open flow supports end to end (for example an iCE40 or ECP5 part); the design, the tool versions and the build script kept in a versioned repository, so that the same bitstream can be rebuilt; a soft processor core in the design that emits telemetry as space packets; and a test that checks the packets on a ground computer. A later step adds the voting and the fault injection of chapter 4. Results, when they exist, will be reported with their dates; until then nothing in this book says this step was done.

**TRADE-OFF: MORE BITS TO PROTECT, AND A DIFFERENT REPAIRER**

No FPGA anomaly in flight could be traced to a primary record for this chapter, so the lead case is a trade-off in one maker's own figures, with its conditions. AMD gives the configuration memory of its XQRKU060 as 193 Mb and that of its XQRVC1902 as 363 Mb, 1.88 times as many bits that a particle can upset (363 divided by 193; AMD's slide rounds it to about 80 percent more). With its older soft error manager, which runs in the user's logic, AMD estimates one functional interrupt of the XQRKU060 in about 6 years; with XilSEM, running on triplicated hardwired processors, one of the XQRVC1902 in about 200 years. Both figures are "with mitigation", for geostationary orbit at solar minimum, and from "AMD internal radiation test data". On the maker's own estimates, the newer part with its different repairer comes out ahead, although the estimates do not apportion the gain between the part, its process and the repairer: the capacity added bits to protect, and the newer product moved the protection into fixed, triplicated silicon. The figures are the maker's, with no independent check found; they speak to configuration interrupts only, not to the user's own state or to the processor system, which has its own interrupt rate.

SECURITY NOTE

**Who can interfere or command.** A bitstream is the design. Whoever can load one into an FPGA in orbit can change what the satellite's hardware does, and an SRAM FPGA that is reconfigured or scrubbed from a stored copy trusts that copy. The parties with that power are whoever can command a reconfiguration, whoever can write the stored image, and whoever built the bitstream and the tools that made it. **How it is seen.** By authenticating the bitstream before it is loaded and keeping a record of every load. Frontgrade lists "AES-256 bitstream encryption and ECDSA authentication" for its UT24CP1008; AMD's product table gives the Versal parts' platform management controller the roles of "Boot, Security, Safety, Monitoring, High-Speed Debug, SEU Mitigation". The same AMD deck says its beam tests exercised that controller "except security block", so the security block's own radiation behavior is not covered by that test. For the other families, the documents cited here do not describe bitstream protection, and a buyer asks. Authentication shows where a bitstream came from and that it was not changed; it does not show that the design is correct. A scrubber that repairs from a stored copy cannot tell a particle from a deliberate change, so the copy itself needs protection. **How it is recovered.** By a protected first-stage image that a bad upload cannot overwrite, from which the device can be reconfigured and the ground can upload again; and by a command path for reconfiguration that is separate from the one used by hosted users. Detecting a wrong configuration, diagnosing whether a particle, a fault or a person caused it, and attributing it are separate steps; the first two can be designed for in advance, the third usually needs an investigation.

**Monday rule:** for any FPGA in a proposal, ask three things first: where the configuration lives, what protects the user's state, and what process it is made on; then who checks the configuration, and how often.

CHAPTER EIGHT

# Software-Defined Payloads: What Can Be Changed from the Ground




At 23:00 Central European Summer Time on 30 July 2021, a rocket left French Guiana with Eutelsat Quantum on board, a geostationary telecommunications satellite that the European Space Agency described as one that "can be completely repurposed while in space." A conventional fixed-beam, fixed-channel payload was largely set at the factory: its beams, its frequency plan and its power split were decided years before launch, and the operator lived with them, apart from the switching it had been built with, for the satellite's life. The claim behind the word software-defined is that some of those decisions move from the factory to the control room. This chapter asks which ones, and what computing on board makes that possible. The answer changes one decision for a reader: before paying for flexibility, ask for the list of what can be changed, by whom, and how long a change takes.

## A definition before the cases

In this book a **software-defined payload** is a payload whose coverage, frequency plan, power allocation, routing or applications can be changed after launch by a command or an upload from the ground. Some of that can be done by commanding radio hardware alone (switched filters, steered antennas); this chapter follows the payloads in which a digital processor sits in the signal path, and says so where a case does not show which kind it is. The first book of this series, *The Bit's Path*, gives the short form: a payload whose beams, frequencies and power can be changed from the ground, transparent or regenerative. This chapter adds what is inside.

Three things that an operator might change are kept apart, because they differ in risk, in who may do them and in how a mistake is undone:

- **Parameters.** Values that existing software reads: which channel goes to which beam, how much power a beam gets, where a steerable beam points. A change of this kind is a command.
- **Firmware and configuration.** The program of a processor or the bitstream of a programmable logic device (chapter 7). A change of this kind replaces part of the payload's design, and a bad one can stop the payload.
- **Applications.** Programs added to run beside the payload's own functions, as on the reprogrammable missions described below. A change of this kind brings new code, possibly written by someone other than the satellite's builder.

The word "software-defined" in an operator's description rarely says which of the three it means. The cases below say it where their sources do.

### Digital depth, and a separate question: can new code run?

3GPP, the body that writes the mobile network standards, defines two kinds of satellite payload for its non-terrestrial networks. Its overview page describes a **transparent** payload as an analog radio frequency repeater: it shifts the frequency, filters and amplifies, and does not process the signal's content. A **regenerative** payload demodulates, decodes, switches or routes, and then codes and modulates again, so that it can carry functions of a base station. Release 17, whose protocol specifications were frozen in June 2022, was the first 3GPP release with normative requirements for non-terrestrial networks; Release 19 work adds support for regenerative payloads.

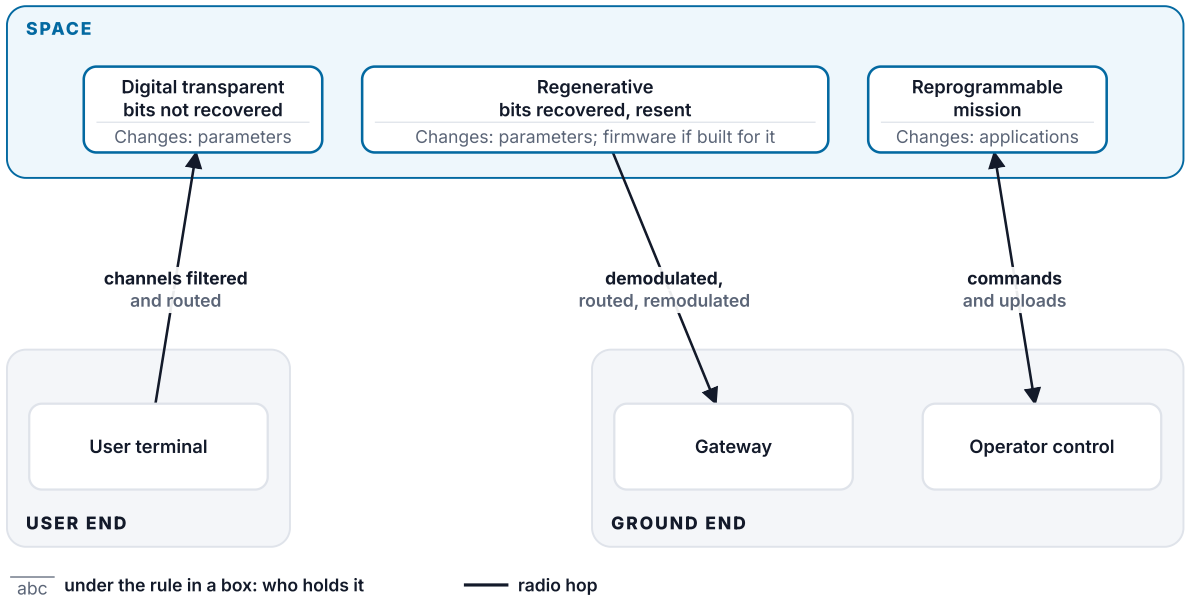
In practice the line is drawn at one question: are the bits recovered on board? A payload can filter, channelize and route digitally and still be transparent, because it passes the waveform on without regenerating its data. Much of what is sold as software-defined is transparent in that sense. This chapter marks three reference points; figure 8.1 draws them.

- **Digital transparent processing.** The payload converts the received band to samples, filters it digitally into channels, and routes each channel to an output beam, then converts back to radio. It never recovers the bits. The routing and the filter shapes are parameters that can be changed by command. In 3GPP's terms the payload is still transparent, because nothing is demodulated.
- **Regenerative.** The payload recovers the bits, decides where each piece goes, and transmits it again, possibly with a different code and modulation on each side. The second book of this series, *From Telstar to the Phone in Your Pocket*, tells how NASA's ACTS tested this kind of switching in orbit from 1993; it is not retold here.

- **A reprogrammable mission.** The satellite carries a general-purpose computing platform on which new applications can be loaded after launch, so that what the satellite does with its instruments and links can change within its hardware. This point answers a different question from the first two: whether new code can run, not whether bits are recovered. A satellite that hosts applications need not be regenerative, and a regenerative payload need not accept applications.

FIGURE 8.1

**Three reference points: two on whether bits are recovered, one on whether new code can run, and what an upload can change**




The first two points differ in whether bits are recovered; the third, whether new code can run, is a separate question; the line under each satellite says what a change from the ground reaches. The changes listed are typical, not required: what a payload can change is a design choice separate from whether it recovers bits; a digital transparent processor can also take new firmware if it was built for it.

*Author's diagram based on the 3GPP definitions of transparent and regenerative payloads (3GPP, Non-Terrestrial Networks overview) and the cases of this chapter.*

The points are reference marks, not product classes. The same hardware can sit at different points with different firmware: a processor that today channelizes and routes may, with a new design loaded, demodulate one carrier, if the hardware has the capacity and the operator has the rights. That is why a reader asks for the list of changeable functions, not the label.

### The cases, each with its status fields

Each case below carries what its own source says, with the date, kept in the status fields of chapter 1: launched, demonstrated function, current health, service access, announced. A field that no source read speaks to is left empty. Figure 8.2 sets them side by side, and Appendix D dates each row.

- **Eutelsat Quantum (geostationary).** Launched on 30 July 2021, at 23:00 Central European Summer Time; ESA's press release says the satellite "can be completely repurposed while in space." The press release, as read for this chapter, describes the satellite's flexibility in general terms and gives no list of parameters with their limits; other partner pages could not be read when checked, so this chapter prints no list.
- **O3b mPOWER (medium orbit).** Boeing, the builder, reported that the last three satellites, F11 to F13, launched on a Falcon 9 on 13 September 2026, bringing the number launched to 13. Boeing states that the software-defined payload lets the satellites reroute bandwidth, concentrate beam power and adjust capacity "in real time", a phrase it does not define with a time. SES, the operator, calls O3b mPOWER its "software defined Medium Earth Orbit (MEO) system." SES's results release of 5 November 2026 says the final three are expected to be operational in mid 2027: launched, not yet in service, on the operator's own word.
- **Iridium (low orbit).** In a release of 21 January 2026, Iridium described trials of its standards-based narrowband service as using "new 5G waveform algorithms implemented on Iridium's software-defined satellites," and said the service was "activated via software update to the Iridium constellation." On the operator's account, then, an existing constellation gained a service by upload; the release does not say which elements on board or on the ground the update changed, so this case does not give the list that this chapter's Monday rule asks for.
- **Space INSPIRE (geostationary, Thales Alenia Space).** ESA, which supported the product line, describes a "fully software-defined payload architecture that enables complete in-orbit reconfiguration," and named SES, Arabsat, Intelsat, Eutelsat and SKY Perfect JSAT as customers. Its first satellite, ASTRA 1Q, was "scheduled for launch in 2026" in an ESA article of 28 May 2025. No source read for this chapter confirms a launch, so the status field is "announced."
- **SmartSat (Lockheed Martin).** Announced on 20 March 2019 as a software-defined satellite architecture that lets satellites change missions in orbit. The first missions named were Linus, two 12U CubeSats, and Pony Express, a 6U CubeSat, carrying a radiation-hardened computer developed by the NSF SHREC center, a hypervisor and multi-core processing. The announcement is the source; what the missions demonstrated in orbit is not in it.

- **TianZhi-1 (China).** A conference paper of 2019 calls TianZhi-1 "China's first software-defined satellite" and describes an application uploaded to it, running on an Android smartphone used as the computing platform, that reduces image redundancy before downlink. The source is a research paper by the application's authors, labeled as such; no agency page was read.

Three cases the plan for this chapter named could not be confirmed from a primary source when checked: Airbus's OneSat, the reconfigurability of Hughes's JUPITER 3 and of ViaSat-3, and the digital payload of Japan's Engineering Test Satellite 9. No Indian case was found. Their absence here is a gap in the sources reached, not a judgment of the systems; Appendix D lists them as sought.

FIGURE 8.2

## Software-defined payloads by status field: what each source says can change, and how far each case has gone

| Case | What the source says can change | Launched | Service or demonstration | Source, date |
|---|---|---|---|---|
| Eutelsat Quantum, geostationary | "Completely repurposed while in space"; parameters not listed in the release read | 30 July 2021 | Not in the source read | ESA, 2021 |
| O3b mPOWER, medium orbit | Reroute bandwidth, concentrate beam power, adjust capacity 'in real time' (no time stated) | 13 satellites after 13 September 2026 | Last three expected operational mid 2027 | Boeing, SES, 2026 |
| Iridium constellation, low orbit | New waveform algorithms; a service activated by software update | In orbit (launch dates not in this source) | On-air trials of a narrowband service, January 2026 | Iridium, 2026 |
| Space INSPIRE (first: ASTRA 1Q), geostationary | "Complete in-orbit reconfiguration" | Not confirmed; scheduled for 2026 in May 2025 | Announced | ESA, 2025 |
| SmartSat, small satellites | Missions changed in orbit by software upload; hypervisor | Not in the source read | Announced 2019 with two demonstration missions | Lockheed Martin, 2019 |
| TianZhi-1, China | Applications uploaded to a smartphone-based platform | Not in the source read | An image application run in orbit, per its authors | Conference paper, 2019 |

Every cell reports what its cited source establishes or claims on its date (a paper's in-orbit run, an operator's release, a maker's statement); an empty field means no source read speaks to it, not that the answer is no.

*Sources: ESA press release 25-2021; Boeing feature of September 2026; SES O3b mPOWER page and results release of 5 November 2026; Iridium release of 21 January 2026; ESA resilience article of 28 May 2025; Lockheed Martin release of 20 March 2019; ISKE 2019 conference paper on TianZhi-1. All read November 2026.*

## What does the work on board

Three kinds of part do the processing, and each was described in the chapters before this one.

- **Application-specific integrated circuits (ASICs).** A chip designed for one payload's processor. Its circuitry is fixed when it is made, though it may implement much programmable behavior: routing tables, channel maps, beam weights, embedded processors. Its hardwired circuits cannot be rewired after manufacture; new behavior has to fit the programmable resources built into it, such as an embedded processor whose memory and upload path allow new code.
- **Programmable logic (FPGAs, chapter 7).** In a reprogrammable FPGA the design is a stored configuration, so the function itself can change, if the spacecraft was built to accept a new configuration safely; an antifuse part is programmed once. With the change come the risks of chapter 7: a bad bitstream for any reprogrammable part, and for SRAM configuration also configuration upsets and a scrubber to trust.
- **Processors (chapter 5).** General-purpose cores run the control software and, on reprogrammable missions, the applications. BAE Systems describes its RAD5545 SpaceVPX single board computer, a radiation-hardened multi-core card, as able to host software-defined radio payloads, one card replacing several earlier ones.

A payload usually combines them: fixed-function silicon where the data rate is highest, programmable logic where the function may change, a processor to manage both. Which part sits where decides what can be changed later. A function that the fixed silicon was not built to vary cannot be changed later, whatever the brochure calls the satellite.

## A change, step by step

A change to a payload in orbit is an operation, not a click, and a buyer should expect these steps whatever the operator calls it. No operator's own timings were found, so the steps are stated without durations; asking for the durations is part of the Monday rule.

- **Define the change.** A new beam plan, a new channel map, a new firmware image or a new application, with the service it is meant to deliver and the spectrum rights it uses.
- **Check it on the ground.** Against a model of the payload or an engineering model of the hardware, including what the change does to power and heat on board, because moving power between beams moves heat between amplifiers.
- **Send it up.** Through the command link: a parameter set is small, a firmware image much larger, and the command link's rate decides how many contacts an upload takes. The upload should be checked on board before it is used.

- **Switch over.** At a planned time, with the users of the affected beams told in advance, since a beam that moves or changes frequency may interrupt the terminals in it unless the change is made before the old beam is dropped.
- **Verify, or roll back.** Measure the new beams from the ground and through the operator's telemetry; if the result is wrong, return to the previous configuration, which must still be on board or able to be sent again.

A buyer asks for evidence that an update is safe, not only that it is authenticated: a recovery path and watchdog that the update cannot touch; a validated recovery image kept on board; staged activation with health checks; a rollback that also restores stored state; limits on radio output, power and heat while the new version runs; and a defined outcome if power or the contact is lost during the upload or the switch. The strongest evidence is a recovery from a deliberately bad update on representative hardware.

Iridium's account of its 2026 trials shows the last two steps at the scale of a constellation: a service "activated via software update," then tested by sending messages on air. What a buyer cannot see from outside is how long each step took and what would have happened had it failed.

> **WHAT HASN'T CHANGED**
>
> Unless a servicing mission reaches it, the satellite cannot be repaired or extended in hardware. A software-defined payload moves the decisions about coverage and capacity from the factory to the control room, but every choice it can make was paid for at launch: in gates and cores, in watts and radiator area, in antenna elements and in spectrum filings. Flexibility in orbit is a reserve bought before launch and spent afterward; software can use it in new ways, but without servicing no hardware is added to it.

## What stays fixed

Software can move what the satellite has; it cannot give it what it does not have. Four limits are set by the time of launch, and only the third can change afterward, and then not by software.

- **The aperture.** Antenna size and the number of elements set the gain and the narrowest beam. A digital beamformer can point and shape beams only within what the antenna allows.
- **The power system.** The solar arrays, the batteries and the amplifiers set the radio power available. At the limit of the available power, moving power to one beam takes it from another.

- **Spectrum rights.** A payload that can tune anywhere may transmit only where its operator holds rights. Those can change after launch, but through a regulator, not a command; the fourth book of this series, *Who Owns the Sky*, owns that subject.
- **Processing capacity on launch day.** The number of gates, cores and memory flown is fixed. New firmware can use them differently, but it cannot add any; a function that needs more than the processor has cannot be uploaded.

For links, the regenerative payload has a consequence in decibels: each hop is decided on its own, instead of the noise of both hops adding up in one circuit. The first book of this series works that out on its ledgers, LL-1998-01 for a handset call through Iridium's first satellites of 1998, which switched calls on board, and LL-2026-04 for a low orbit broadband path. This book prints no filled hop budget.

## The trade: flexibility against power, mass and heat

A digital processor in the signal path draws power that an analog payload of the same service would not draw for that function; it may also replace analog filters and switches, and change how the amplifiers are run. Its converters between radio and samples also add their own noise and distortion, a further term in the comparison. So the honest comparison is between two complete payloads delivering the same service, not between a processor and nothing. No published comparison of complete payloads at equal service was found, so this chapter states no such number. What the peer-reviewed literature does show is that the processor's power is managed as a resource: one paper on digital payloads for broadband satellites in non-geostationary orbits treats the payload as several processors and sets out to switch off as many as the traffic allows (IEEE Transactions on Aerospace and Electronic Systems, 2023), and another benchmarks a bent pipe, a beam hopping payload and a hybrid beam steering payload for a low orbit constellation, as a trade between flexibility and complexity (International Journal of Satellite Communications and Networking, 2020). Both are studies, not flight records.

### FAILURE, BOUNDED: INSURANCE ON THE FIRST O3B MPOWER SATELLITES, CAUSE NOT STATED

The case here is bounded by what its source shows. SES reported, in a results release of November 2026, that it has collected 229 million dollars of insurance on O3b mPOWER satellites 1 to 4. The SES documents read for this chapter do not state the cause, and nothing in them says that the software-defined payload failed; this chapter does not say so either. The lesson it supports is narrower and holds whatever the cause: a payload that can be reconfigured by command can redistribute only the capacity that the satellite's hardware still delivers. Flexibility in the payload does not replace a working power system, amplifier or antenna, and an operator who buys flexibility still carries the risk of the hardware.

### SECURITY NOTE

**Who can interfere or command.** A payload that can be redefined by command can be redefined by whoever holds the command path. Five questions apply to every case above: who is authorized to send a reconfiguration, a firmware image or an application; who can read the configuration the payload holds; who can interfere, through the command link, through an application loaded beside the payload, or through the supply chain of a bitstream; how a command or an upload is authenticated; and how the payload is recovered. None of the operator sources read describes its authentication; a buyer asks for it in writing. **How it is seen.** Detection, diagnosis and attribution are separate steps: a telemetry change shows that a beam moved, not who moved it or why. Ask for a log of every reconfiguration with its origin and its check. **How it is recovered.** Ask how the previous configuration is restored, from where, by which path, and whether that path still works when the payload's own processor runs a bad image: a protected first image (chapter 7) and a separate command route are what make a rollback possible.

**Monday rule:** ask for the list of what can be changed in orbit, by whom, how long a change takes, and what cannot be changed.

CHAPTER NINE

# Processing the Data On Board: Compress, Filter, Decide




On 11 April 1991, far enough from the Sun at last, the Galileo spacecraft ran the stored commands that were to open its 4.8 meter high-gain antenna. Telemetry received minutes later showed that the motors had stalled and the antenna had only partly opened. It never opened further. The antenna had been designed to send data from Jupiter at up to 134,400 bits per second; the small antenna left to the mission could manage 8 to 16. By JPL's plan of December 1995, new software on the spacecraft and new equipment on the ground were to raise that to as much as 160 bits per second and make each bit carry far more science. Galileo is the plainest case of a sensor making more data than the link can carry, and of what computing on board can and cannot do about it. This chapter asks that question for every satellite: what has been done on board to the data before the downlink, and with what result.

## The data volume budget

A link budget in decibels asks whether one hop closes at one rate. The question of this chapter is one level up, and its unit is the bit, not the decibel: how much data the sensor produces per orbit, how much the satellite can store, and how much each contact with the ground can carry away. In words: what is stored at the end of an orbit is what was stored at its start, plus what was written to the store (after any compression, and counting raw and processed copies held together), less what was deleted on board, less what the contacts delivered and released. If that keeps growing, the store fills and data is lost, or the sensor is switched off.

An illustrative case shows why a daily average is not enough (inputs assumed, results calculated). A satellite writes 12 Gbit to its store each orbit, after compression, 180 Gbit over 15 orbits a day; four contacts a day each deliver 48 Gbit, 192 Gbit a day, so on average the downlink keeps up. But the four contacts fall in four orbits in a row, and eleven orbits pass without one: the store climbs to 132 Gbit before the first contact and is drained over the next

four orbits (132, then 96, 60, 24 and empty). If one contact is missed, 36 Gbit are still in the store after the day's last contact, and the next peak reaches 168 Gbit; with only 12 Gbit a day of spare downlink, that backlog takes three days to clear. The store must hold the peak after a missed contact, with a margin, not the daily average.

The European radar satellite Sentinel-1 shows the shape at full scale. Its mission description gives 1,410 gigabits of storage on board and a payload data downlink of 520 megabits per second, two channels of 260, sent either by X band to a ground station or by optical link through a relay satellite. The store holds what the satellite acquires between opportunities to send it; a relay gives it more such opportunities. Every Earth observation satellite has the same three numbers, smaller or larger: produced, stored, sent per contact.

Three kinds of on-board processing reduce what has to be sent, and they save different kinds of data:

- **Compress.** Fewer bits for the same content. Lossless compression gives back every bit on the ground; lossy compression gives back an approximation, with a bounded or unbounded error.
- **Filter.** Whole items are judged on board and some are never sent, such as images under cloud. A discarded item cannot be recovered later; only what the satellite reports about it, a count, a mask or a thumbnail, reaches the ground.
- **Decide.** The satellite changes what it acquires next, so that some data is never produced: it looks again at a volcano that its own analysis found active.

Compression changes how much each item costs to send; filtering changes which items are sent; deciding changes what exists to send. Figure 9.1 shows the first two side by side on one illustrative day, with the source of each number. Their reductions in volume can be combined, as figure 9.1 does, but their effects on what information is kept have to be reported separately.

FIGURE 9.1

### One illustrative day: lossless compression shrinks every image, filtering removes some images, and of the two only filtering loses data for good

| Step, per illustrative day | Send everything | Lossless only | Cloud screening, then lossless | Kind of number |
|---|---|---|---|---|
| Produced by the sensor | 100 | 100 | 100 | assumption (units) |
| Discarded on board as cloudy | 0 | 0 | 30 | assumption, from an operator claim |
| Lossless compression factor (bits before over bits after) | none | 2.5 | 2.5 | assumption, inside a preprint's range |
| Volume to send | 100 | 40 | 28 | calculated |
| What the ground can no longer recover | nothing | nothing | the 30 discarded, including any clear image marked cloudy | follows from the method |

An illustrative model, not a mission's record: 100 units produced, equal-size images, a filter that removes 30 of them (an assumption suggested by the Phi-Sat-1 chip maker's bandwidth figure), and lossless compression at 2.5, inside a preprint's range of about 2.4 to 3.

*Author's illustrative model. Inputs: about 30 percent of bandwidth saved (Intel Newsroom, 2020, operator claim), converted to a share of images by assumption; lossless image coding at about 4 to 5 bits per pixel, a ratio of about 2.4 to 3 for 12-bit data (preprint arXiv 2610.12182, 2026, not peer reviewed). Arithmetic calculated.*

## Moving the result: compression standards and file delivery

Space agencies standardize compression through the Consultative Committee for Space Data Systems (CCSDS), so that a compressor built into one mission can be decoded by any ground system that follows the standard. Three issues are current in the papers that cite them. CCSDS 121.0-B-3 (2020), Lossless Data Compression, describes a block-adaptive coder

based on the Rice algorithm. CCSDS 122.0-B-2 (2017), Image Data Compression, describes a technique that can be run lossless or lossy. CCSDS 123.0-B-2 (2019), for multispectral and hyperspectral images, adds a near-lossless mode to the lossless one of its first issue (2012): the user sets the largest error allowed in any reconstructed value. These identifications come from the reference lists and descriptions of peer-reviewed papers that implement the standards; the CCSDS documents themselves could not be opened when this chapter was checked.

How much lossless compression saves depends on the data. One recent preprint, not peer reviewed, puts lossless image coding to CCSDS 122.0 at about 4 to 5 bits per pixel on typical Earth observation test images; for 12-bit data that is a ratio of about 2.4 to 3. Lossy coding goes further and costs fidelity, which the mission has to accept in writing.

Compressed data still has to reach the ground whole. The CCSDS File Delivery Protocol, issue 727.0-B-5 of 2020 in the citations read, moves files between a spacecraft's storage and the ground, and NASA missions have used it for flight to ground file transfers, among them the Global Precipitation Measurement mission, per a NASA review of thirty years of flight software. Storage on board, a file protocol and a compressor form one system: the compressor decides how big each file is, the store how long it can wait, the protocol whether it arrives.

## The case that made the argument: Galileo, 1991 to 1997

Galileo was launched from the Space Shuttle Atlantis on mission STS-34, with an Inertial Upper Stage, on 18 October 1989. Its computers are described in chapter 2; this section follows its link. JPL's press kit for the arrival at Jupiter, dated December 1995, tells the antenna story and the recovery plan, and is the main source here.

**The antenna.** The high-gain antenna was an umbrella of wire mesh on 18 ribs, designed to send X band data at up to 134,400 bits per second, "about one imaging frame each minute." Because the flight path swung past Venus and twice past Earth, the heat-sensitive antenna had to stay folded behind a sun shade until April 1991. When the motors stalled, more than 100 engineers from JPL and industry analyzed the telemetry and tested an identical spare. They concluded that a few ribs were most likely held by friction between their standoff pins and sockets, after a dry lubricant was lost, probably from vibration during cross-country truck shipments of the antenna in the 1980s. Their analysis showed that the problem existed at launch and went undetected. The team turned the spacecraft toward and away from the Sun to work the ribs free by heating and cooling, ran the deployment motors on and off more than 13,000 times between December 1992 and January 1993, and spun the spacecraft to 10 revolutions per minute. None of it freed the antenna.

**What was left.** Two low-gain antennas in S band, designed for the cruise. JPL put the difference plainly: the received signal "is 10,000 times fainter," so the data must be sent more slowly. At Jupiter, without changes, that meant 8 to 16 bits per second.

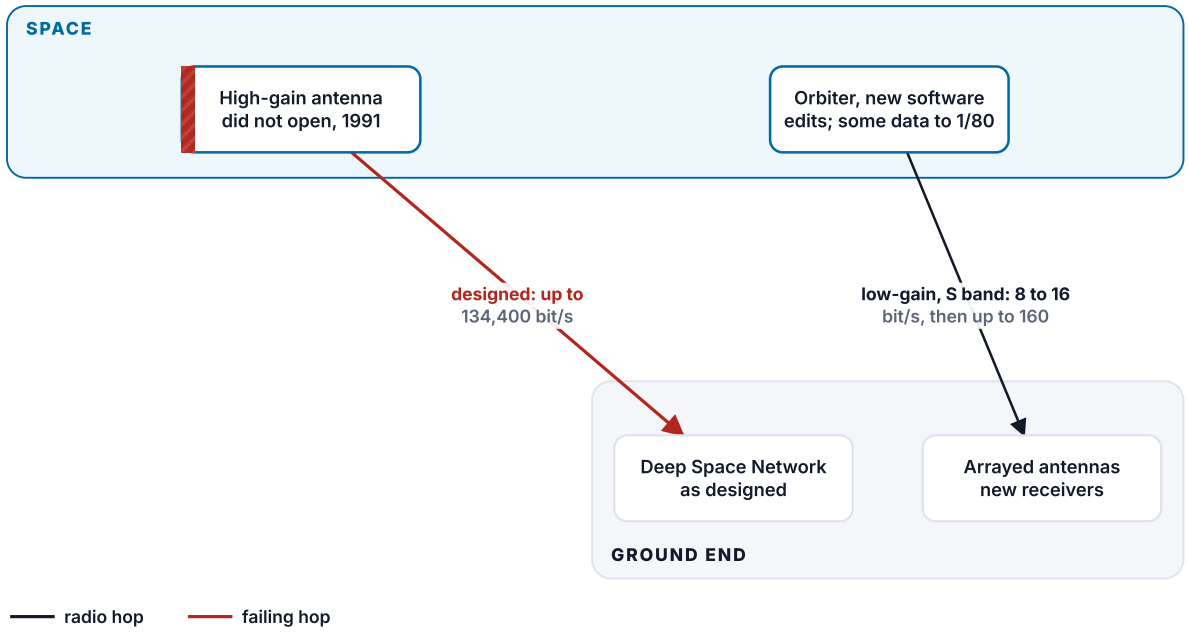
**What was changed on board.** Two sets of new flight software were written and loaded in flight. Phase 1 began operating in March 1995, designed to back up the most important data of the atmospheric probe. Phase 2 was loaded during the first orbit of Jupiter (the press kit planned March 1996) and gave the spacecraft "extensive new onboard data-processing capabilities," in the words of JPL's later fact sheet. It edited and compressed the data on the tape recorder before playback: lossless compression for data that must come back exact, and lossy compression that shrank "imaging and plasma wave data down to as little as 1/80th of its original volume." Papers in NASA's technical series of 1993 describe the choices as proposed: an integer cosine transform for images, chosen because it needs no multiplications, and a dictionary coder for other data, both to be written for processors of the Command and Data Subsystem and the attitude control computer "which have very limited computational and memory resources." The new link design also relied on stronger coding: a convolutional code of constraint length 14 and rate 1/4 with a Reed-Solomon code of variable redundancy, per a NASA paper of 1994.

**What was changed on the ground.** S band capabilities of the 1980s were restored at the 70 meter antenna at Canberra, which saw most of Galileo's tour because Jupiter was in the southern sky. New Block V receivers made it possible to transmit with the carrier suppressed, so that, in JPL's words, all of the signal power could be "dedicated to the data stream." The 70 meter and two 34 meter antennas at Canberra were arrayed and their signals combined, with the 64 meter Parkes radio telescope added, and the 70 meter antenna at Goldstone when its view overlapped.

**The plan, and what was reported.** JPL's plan of December 1995 said the changes together would raise the rate from Jupiter "by as much as 10 times, to 160 bps," increase the information returned by "at least 100 times," and achieve at least 70 percent of the original science objectives. The imaging team reported afterward, in a peer-reviewed paper of 1999, that the camera returned 1,645 unique images of Jupiter and its satellites in the two-year primary mission, that the lossy compressor "proved essential," and that its artifacts had to be held to acceptable levels by the choice of its settings, together with multiple playback passes of the tape recorder. Figure 9.2 draws the link before and after.

FIGURE 9.2

## Galileo before and after 1991: the designed link failed, and the recovery split the work between new software on board and new equipment on the ground




Rates are JPL's: 134,400 bit/s designed, 8 to 16 bit/s through the low-gain antenna without changes, up to 160 bit/s with the changes (plan of December 1995); the arrayed antennas were Canberra's 70 m and two 34 m, with Parkes and Goldstone when in view.

*Author's diagram based on JPL, Galileo Jupiter Arrival press kit, December 1995, pp. 22 to 27; and Optimizing the Galileo Space Communication Link, NASA technical report, 1994.*

Two lessons follow, and each stays within what the sources show. First, the rate in bits per second came back through the link: arraying, new receivers and stronger codes, which JPL credits together with the new software for the factor of ten. Compression did not add a decibel to the link; it reduced how much information had to cross it, so that each bit carried more science. Second, all of it depended on a computer that could be reprogrammed in flight, by a team that knew its limits, and on a store, the tape recorder, that itself failed for a time on 11 October 1995 and was then used with care. LL-1995-01 records the link as planned in 1995, with the imaging team's later count.

**LINK LEDGER · LL-1995-01 · SERVICE LEDGER**

1995 to 1997 (prime tour)

System: Galileo: science return through the low-gain antenna after on-board compression **Retired**

**Class: service only**

| | |
|---|---|
| **1 Requirement** | Return the orbiter's science, imaging included, from the tour that began on 7 December 1995, without the high-gain antenna. JPL's aim (1995): at least 70 percent of the original objectives. |
| **2 Ends and path** | Orbiter instruments, tape recorder and computers. On Earth: the Deep Space Network, mainly Canberra's 70 m antenna arrayed with two 34 m antennas and the 64 m Parkes telescope.<br>**Path:** Store and forward, as planned: record during encounter periods; edit and compress on board; play back through the S band low-gain antenna, direct to Earth, in the months between. |
| **3 Orbit and time** | Eleven elliptical orbits of Jupiter in the prime tour; Earth about 52 light minutes away.<br>**Time:** Weeks to months from observation to ground, by design. |
| **4 Band and permission** | **Downlink.** S band 2295 MHz, low-gain antenna; NASA Deep Space Network; authorization not in the sources read |
| **5 Capacity** | **Up.** Commands and new flight software: Phase 1 in operation from March 1995, Phase 2 loaded in the first orbit of Jupiter.<br>**Down.** Designed: up to 134,400 bit/s (high-gain). Low-gain at Jupiter: 8 to 16 bit/s unchanged; up to 160 bit/s with the changes (JPL plan, 1995). Signal about 10,000 times fainter (JPL). |
| **6 Who owns each end** | **Space segment.** NASA; built and operated by JPL<br>**Ground segment.** NASA Deep Space Network, with the Parkes radio telescope<br>**Customer relationship.** The science teams<br>**The exit.** Ended 21 September 2003 by a planned plunge into Jupiter |
| **7 Trust** | The mission's operations team commanded the spacecraft and loaded software, through the Deep Space Network. Upload authentication: not described in the sources read. |
| **8 How it fails, and how we would see it** | Antenna ribs most likely held by friction after a lubricant was lost in transport; present at launch. Tape recorder stuck on 11 October 1995. Lossy compression leaves artifacts.<br>**How we would see it:** Telemetry minutes after the 11 April 1991 command: stalled motors. Image artifacts watched by the imaging team and bounded by the compressor's settings. |
| **9 Cost shape** | To December 1997: 1.354 billion dollars, 892 million for the spacecraft (JPL, 1995). Recovery cost: not in the sources read. |
| **10 Evidence** | 134,400; 8 to 16; up to 160 bit/s; compression to 1/80 operator claim JPL press kit, December 1995, pp. 22 to 28, 2026-11-23<br>S band 2295 MHz operator claim NASA technical report, 1994, 2026-11-23<br>1,645 unique images measured Imaging team, Optical Engineering, 1999 (peer reviewed), 2026-11-23 |

**Service only, no hop budget:** This book prints no filled hop budget; the sources read give rates and ratios, not budget lines. The data volume side is in the chapter text.

### Filter: deciding on board what is worth sending

Filtering discards whole items before the downlink. Clouds make the case: an optical image of the ground taken through thick cloud carries little for most users, and the satellite can see that before it spends a contact on it.

**Phi-Sat-1**, an ESA experiment on one of the two FSSCat CubeSats, launched in September 2020 and ran a neural network on a commercial vision processor to mark cloud in each image of its hyperspectral camera. An image with more than 70 percent cloudy pixels counted as cloudy and was not to be sent. Its team set a strict requirement on the costly error: clear images wrongly marked cloudy, which are lost for good, under 1 percent on the test set. In flight, 612 tiles cut from 17 image cubes were checked tile by tile (the discard rule and the requirement apply to whole images), with 96 percent accuracy and no such false positives, on scenes that the team notes were planned for applications, not chosen to be hard. The chip maker said the screening saved about 30 percent of bandwidth, without saying of what or over what period; that is its claim, not a published measurement. Chapter 10 examines the network and the processor; this chapter keeps the data volume. LL-2020-01 records the service as far as its sources go, and its emptiest fields, downlink rate and ground stations, are the ones a data volume budget most needs.

**LINK LEDGER · LL-2020-01 · SERVICE LEDGER**

2020 System: Phi-Sat-1 on FSSCat: an Earth observation CubeSat that screens clouds on board

Status on the book's date: not checked | Class: service only

| | |
|---|---|
| **1 Requirement** | Show in flight that a neural network on a commercial accelerator finds cloud well enough to discard, before the downlink, images more than 70 percent cloudy, with clear images wrongly marked cloudy under 1 percent of the test set. |
| **2 Ends and path** | On one of the two FSSCat CubeSats: the HyperScout-2 hyperspectral imager, its processing unit, and an accelerator board with an Intel Movidius Myriad 2 vision processor. Ground stations: not named in the sources read.<br>**Path:** Store and forward, with a decision on board: the image is tiled, each tile classified, and an image judged cloudy is not downlinked. |
| **3 Orbit and time** | Low Earth orbit; altitude and inclination not in the sources read. Launched in September 2020 (2 September by the chip maker's date).<br>**Time:** The decision takes about 102 ms per tile of 192 by 192 pixels in three bands, by the design figure; what it saves is downlink time at the next contacts. |
| **4 Band and permission** | **Downlink and command.** Frequencies and authorization not in the sources read |
| **5 Capacity** | **Up.** Not in the sources read. The cloud threshold is set on board and can be changed without retraining the network.<br>**Down.** Not published in the sources read. Saving: about 30 percent of bandwidth, by the chip maker's claim. |
| **6 Who owns each end** | **Space segment.** An ESA technology mission; partners named by the chip maker: Ubotica, cosine, University of Pisa, Sinergise, Tyvak International<br>**Ground segment.** Not named in the sources read<br>**Customer relationship.** None; a demonstration<br>**The exit.** Status on the book's date not checked |
| **7 Trust** | The network was built by the University of Pisa and trained on synthetic data made from Sentinel-2 images by Sinergise. Who may change the model or the threshold in orbit, and how an upload is authenticated, are not described in the sources read. |
| **8 How it fails, and how we would see it** | A clear image marked cloudy is discarded and lost for good; a cloudy one marked clear wastes downlink. An upset in the accelerator could give a wrong mask with no error (chapter 10).<br>**How we would see it:** By sending discarded images, or their masks and thumbnails, down for checking. In flight, 17 image cubes cut into 612 tiles were checked: 96 percent accuracy and no false positives, on scenes not chosen to be hard. |
| **9 Cost shape** | Not published in the sources read. |
| **10 Evidence** | Cloudy above 70 percent; false positives under 1 percent (operator claim) Giuffrida and others, IEEE TGRS 60, 2022 (peer reviewed), sections IV, V, 2026-11-23<br>612 in-flight tiles: 96 percent accuracy, 0 false positives (measured) Same paper, section V (mission team), 2026-11-23<br>About 102 ms per tile (operator claim) Same paper, section IV, 2026-11-23<br>About 30 percent of bandwidth saved (operator claim) Intel Newsroom, 2020, 2026-11-22 |

**Service only, no hop budget:** No downlink rate, frequency or ground station for this mission is in the sources read, so no hop budget is possible; the data volume side is shown as an illustration in figure 9.1.

Others followed with the same filter and bigger instruments. KP Labs states that **Intuition-1**, a 6U satellite with a 192-band hyperspectral sensor and its own data processing unit, launched in November 2023, runs image processing in orbit at 3 trillion operations per second and reduces data transmission by filtering cloud-covered images. The same company states that **Phi-Sat-2**, a 6U satellite with a multispectral instrument, launched in August 2024 with its cloud detection network quantized to 8 bits; another partner's mission page lists March 2024, and ESA's page could not be opened, so the month is given here as each source gives it. Neither source publishes how much downlink was saved in operation.

Hyperspectral instruments are where the volume presses hardest. A peer-reviewed paper of 2023 by the team of the Norwegian CubeSat HYPSO-1 describes its on-board pipeline, flown and demonstrated in orbit, of capture, binning and lossless compression, with robust booting and update procedures. The same paper cites a proposal for Europe's planned CHIME mission: an instrument gathering data at more than 5 gigabits per second against a proposed downlink of 3.7, which is why the proposal calls on-board cloud detection necessary. Those are figures of a mission in development, reported second hand, and the book treats them as such.

## Decide: changing what is acquired

The third step goes past the data already taken. NASA's Earth Observing-1 satellite, launched on 21 November 2000 as a one-year technology mission, later carried the Autonomous Sciencecraft Experiment, flying by 2004. Its team describes software that analyzed images on board, detected short-lived events such as floods, planned new observations without waiting for the ground, and sent down only the highest value data; they report that ground planning was greatly simplified and operations cost reduced. Those are the team's own words in peer-reviewed and conference papers of 2004 and 2005. Deciding can save data that filtering cannot, when the plan drops an uninteresting image that is then never taken (looking again at an active volcano adds acquisitions instead), and it spends what filtering does not: the trust that the satellite chose well while nobody was watching.

Figure 9.3 sets the cases side by side with their status fields.

FIGURE 9.3

## Compress, filter, decide: cases by status field, each as its own source states it

| Case | Kind | Launched | Demonstrated, per its source | Saving claimed | Source |
|---|---|---|---|---|---|
| Galileo, Jupiter | compress, edit | 18 October 1989 | New software loaded in flight; 1,645 unique images in the primary mission | information at least 100 times the unchanged link (plan) | JPL; imaging team |
| Phi-Sat-1 on FSSCat | filter | September 2020 | Cloud masks on 612 in-flight tiles, 96 percent accuracy | about 30 percent of bandwidth (chip maker) | Mission paper; Intel |
| Intuition-1 | filter | November 2023 | AI image processing in orbit, cloud filtering | not published | KP Labs |
| Phi-Sat-2 | filter | August 2024 (KP Labs); March 2024 (another partner) | Not stated in the source read (design: cloud detection network quantized to 8 bits) | not published | KP Labs |
| HYPSO-1 | compress | not in the source read | Capture, binning and lossless compression pipeline in flight | not stated as a figure here | Remote Sensing, 2023 |
| EO-1, Autonomous Sciencecraft | decide | 21 November 2000 | Event detection and replanning on board, flying by 2004 | ground planning simplified (team) | Team papers, 2004 to 2005 |

Launch and demonstration are separate fields; a blank saving means no source read publishes one.

*Sources: JPL Galileo press kit (1995) and Optical Engineering (1999); Giuffrida and others, IEEE TGRS (2022), and Intel Newsroom (2020); KP Labs missions page (read 2026-11-23); Langer and others, Remote Sensing (2023); EO-1 papers of 2004 and 2005; Copernicus SentiWiki, S1 Mission.*

## Where it does not pay

Reducing the downlink does not by itself justify processing on board. Four reasons can weaken the case, and the cases above show each of them somewhere.

- **The raw data has a value the filter cannot judge.** A scientist studying clouds wants the cloudy images; a filter tuned for land users throws away that science. Raw data sent today can be reprocessed with tomorrow's methods; data discarded in orbit cannot.
- **The ground can do it later for less.** If contacts are plentiful, or a relay is available, the cost of sending everything may be lower than the energy, mass, qualification and risk of processing on board. The trade study of chapter 11 asks exactly this.
- **The model is wrong.** A filter trained on other data, as Phi-Sat-1's was on synthetic images built from another satellite's archive, may meet scenes it never saw; its errors are silent, because what it discards is never seen.
- **The compression ratio is poor.** Noisy or already compressed data shrinks little, so lossless compression buys little and lossy compression must cost more fidelity for the same saving.

What reaches the ground, and in what processing level, is the subject of the sixth book of this series, *Eyes and Clocks*; this book stops at the on-board step.

FAILURE AND RECOVERY: GALILEO'S HIGH-GAIN ANTENNA, 1991

Galileo's high-gain antenna did not open on 11 April 1991, and the failure was in the hardware: JPL's analysis found ribs most likely held by friction, probably after a dry lubricant was lost in transport, a problem present at launch and not detected before it. The recovery was by computing and by the ground. New flight software loaded in 1995 and 1996 compressed and edited the data on board; arrayed antennas, new receivers and stronger coding were planned to raise the rate from 8 to 16 bits per second to as much as 160. JPL planned at least 70 percent of the original objectives, and the imaging team reported 1,645 unique images from the primary mission. The lesson the sources support: a reprogrammable computer with spare margin turned a hardware loss into a reduced mission, not a lost one; it did not restore the antenna's gain, and the mission paid for the rest in data never returned.

**SECURITY NOTE**

**Who can interfere or command.** On-board filtering decides what the ground never sees. The five questions apply to the filter as to any software: who may change the filter, its model or its threshold (on Phi-Sat-1 the threshold could be changed without retraining); who can read the discarded data before it is erased; who could interfere, by an altered model or by scenes crafted to be judged worthless; how a change is authenticated, which shows its origin and integrity, not that it is right; and how the previous filter is restored. Galileo's sources say only that its software went up through the Deep Space Network; how uploads were authenticated is not described in them. **How it is seen.** A filter that discards the wrong images leaves no trace in the images that arrive. The defense is to send down a sample of what was discarded, on a schedule the filter cannot see, and to compare. **How it is recovered.** Keep the previous filter and its settings on board, and a command path to fall back to sending data unprocessed while the cause is found, with acquisition cut back so that the store does not fill.

**Monday rule:** before asking for a faster downlink, ask what fraction of the data is worth sending, and whether deciding that on board costs less in power, mass and risk than sending it all.

CHAPTER TEN

# On-Board AI: Accelerators, Models and Trust




Phi-Sat-1 went to orbit in September 2020 with a vision processor that, its team wrote, runs at 600 MHz "nominally consuming only 1 W". It ran a neural network that marked each pixel of an image as cloud or not cloud, so that cloudy images could be discarded before the downlink. The network was trained before launch on images synthesized from another satellite's archive, because, as the team put it, "insufficient data volume" from the mission's own camera was available. In orbit it was tested on 612 image tiles. This chapter asks what it takes to run a learned model in orbit, and how anyone would know that its answers are still right.

## What has flown, and what each source says it did

A learned model in orbit needs three things: a processor that can run it within the satellite's power, a way to get the model on board, and a way to check its answers. Most of what has been reported concerns the first. Figure 10.1 lists the accelerators named in the sources read, with their status fields kept apart: launched, demonstrated in orbit, tested or qualified on the ground, announced. An empty field means that the sources read do not establish it.

FIGURE 10.1

## Accelerators for on-board AI, by status field: launched, demonstrated, tested, announced

| System | Device and stated power | Launched | Demonstrated in orbit; service access | Ground test, qualification or announce-ment | Source kind |
|---|---|---|---|---|---|
| Phi-Sat-1 (ESA) | Intel Movidius Myriad 2 vision processor; about 1 W nominal at 600 MHz (mission team) | September 2020 | Cloud mask on 612 tiles from 17 acquisitions, 96 percent accuracy; self-tests passed over 70 days | No latch-up observed to 8.8 MeV cm2/mg; no dose effect observed to 49 krad (material not stated) | Peer-reviewed paper |
| Kepler cluster | About 40 Nvidia Orin processors on 10 satellites; module power not stated (Orin modules listed at 7 to 40 W by CSET) | First tranche 2026-01-11 (per Axiom Space) | Function in orbit not in the source read; 18 cus-tomers on 2026-04-13 (operator claim in trade press; kind of access not stated) | | Trade press |
| Starcloud-1 | One Nvidia H100; power in orbit not stated (H100 PCIe listed at 350 W by CSET) | November 2025 | A model trained and a Google language model run, per the company (trade press) | | Trade press |
| Three-Body Com-puting Constella-tion | Not stated in the source read | First group of 12 space-craft, summer 2025 | | | Agency report (NASA) |

| System | Device and stated power | Launched | Demonstrated in orbit; service access | Ground test, qualification or announce-ment | Source kind |
|---|---|---|---|---|---|
| AMD XQRVC1902 | 7 nm adaptive chip with 400 AI engine tiles | | No public flight record of the AI engines found | Class B com-pleted 2022-11-15; Class Y under evalu-ation in March 2025 | Maker |
| Google tensor pro-cessors (announced design) | Not stated | Not launched in the 2025 announce-ment | | No hard failure up to 15 krad total dose (ground test); proto-types planned by early 2027 (announced) | Maker |

Status fields as each source gives them, read 22 to 23 November 2026; an empty cell means the sources read do not establish it, not that the answer is no.

*Source: Giuffrida and others, IEEE Transactions on Geoscience and Remote Sensing 60 (2022); Intel Newsroom; Axiom Space, Orbital Data Centers; TechCrunch (trade press), 2026-04-13 and 2026-03-30; CSET, Onboard AI: Constraints and Limitations (2023); AMD, release of 2022-11-15 and SEFUW presentation of 2025-03-25; Google Research, 2025-11-04; NASA, State-of-the-Art of Small Spacecraft Technology 2026.*

Three things stand out in the table. First, the devices span more than two orders of magnitude in stated power, from about 1 watt to hundreds, though the figures are of different things (a chip's nominal power, a module's, a card's thermal design power) and are not like for like. Second, only one row has a peer-reviewed account of what the device did in orbit, with numbers: Phi-Sat-1. The others rest on an operator's page, a press release or trade press, and report launches and, in some cases, functions the company claims, but no independently documented performance. Third, a ground test and a flight are different kinds of record. Google reported that its tensor processors showed no hard failure up to a total ionizing dose of 15 krad; that is a ground test of cumulative damage, and it says nothing about the soft errors of the section on radiation below. AMD completed Class B qualification of its 7 nm adaptive chip with 400 AI engine tiles on 15 November 2022; no public flight record of those engines was found.

The table also has a row that the frame of this book cannot fill. NASA's 2026 report on small spacecraft technology states that China's Three-Body Computing Constellation "launched the first group of 12 spacecraft in summer 2025". What those satellites have computed, whether they network with each other in orbit, and who can use them, are not in that report, and no builder or agency source for them was read for this chapter. The row records the launch and leaves the rest empty.

## Inference, not training

A neural network is used in two phases. Training adjusts its parameters, its weights, on a large set of training data, labeled in supervised training such as Phi-Sat-1's; it needs many passes over the data and a great deal of computing. Inference runs the trained network on a new input to get an answer; for an image classifier that is one pass, for a language model's answer many, and in both cases far less computing than training. Everything in figure 10.1 that has a described function in orbit does inference, with one exception that rests on a company's word: Starcloud says, through trade press, that it also trained a model on its satellite (chapter 12), and no paper or log of that work was found.

Training in orbit appears in the research literature as a proposal: papers describe satellites that would adjust a model on their own data, or share the adjustment across a constellation. The Phi-Sat-1 team wrote that improvement "by means of delta training" is feasible for such solutions, which is a statement of what the design allows, not a record of it being done. This chapter found no primary record of a model trained in orbit and then used for a mission's product, in the sources read as of 23 November 2026. A sentence that says a satellite "learns" is ambiguous, and should not be counted as training in orbit until its source says so.

## The model as an upload

A model that can be replaced from the ground is a model that has to cross the uplink, and uplinks are usually much slower than downlinks. A 2024 conference paper by researchers at KTH and partner institutes works the numbers for a science mission. On the Magnetospheric Multiscale mission (MMS), whose four spacecraft fly far out on highly elliptical orbits, they write, "the uplink data rate is only 2 kbps", and with contact windows of 15 to 80 minutes "only 225 kB to 1.2 MB can be uploaded depending on the window". Their own baseline network was 3.668 MB with 32 bit floating-point weights, about four hours of uplink at 2 kbit/s. Stored as 8 bit integers it was 0.937 MB, about one hour; a smaller network of their design was 0.193 MB in 32 bit form, about 13 minutes. Reducing the precision of the weights is a lossy compression: it changes the model, and the authors measured how much accuracy each format kept.

The upload sets a rhythm for the whole trade. A mission that expects to retrain its model as it learns about its own sensor must budget uplink time for every new version, and the number of versions it can send is then fixed by its contacts, not by its engineers. A model small enough to send in one window can be changed in a day; one that needs several windows has to be split, sent and reassembled, and checked before it is used. Phi-Sat-1's network, its authors report, had a memory footprint of 62.9 KB.

## Energy and heat per inference

The power figure printed for an accelerator is rarely the energy a mission pays for an answer. A useful figure is the energy per completed inference, and it depends on the model, the numerical precision, how many inputs are processed together (the batch), how busy the device is kept, the memory traffic, the board around the chip and the host computer that feeds it, plus the power drawn while the device waits. The thermal design power that makers quote is the heat the cooling system is designed to remove; a 2023 report on on-board AI from Georgetown University's Center for Security and Emerging Technology notes that it "is not an absolute measure of power consumption".

Phi-Sat-1 gives the only pair of inputs for a flown network and device found for this book: about 1 W nominal for the processor, and an inference time, stated in the paper's design section rather than measured in orbit, of about 102 ms per tile of 192 by 192 pixels in three bands, using 8 of its 12 vector processors. Multiplied, that is about 0.1 joule per tile, or about 102 joules per thousand tiles: about 0.03 watt-hours. Applied to the 612 tiles of the in-orbit test, those values give an estimated 62 joules of processor energy. This is an estimate calculated from the team's stated values, for the chip alone, and it assumes the nominal power is the power drawn. The board, the camera's own computer that prepared the tiles, and the idle power are not in the published figures.

Figure 10.2 sets that estimate beside two larger device classes. Their makers' published inference rates for this network are not available, so the table does not invent them. It asks the reverse question: how many of the same inferences per second a device at a given power would have to complete to match 0.1 joule each. At the 15 W thermal design power listed for one embedded module, the answer is about 147 per second; at the 350 W listed for one data-center card, about 3,400 per second. A large device that reaches those rates on the workload, while drawing no more than the listed power, uses no more energy per answer than the small one; one that does not may still match it if it draws less than the listed power, since energy per answer is the power actually drawn divided by the answers completed each second. Which it is depends on the measured rate, which a buyer should ask for, with its conditions.

FIGURE 10.2

## Energy per thousand inferences: one estimate for a flown device, and the rate a larger device must reach to match it

| Line | Vision processor (Phi-Sat-1) | Embedded module class | Data-center card class |
|---|---|---|---|
| Stated power (kind) | About 1 W nominal (mission team) | 15 W thermal design power | 350 W thermal design power |
| Time or rate for one 192 by 192 by 3 tile | About 102 ms (stated in the mission paper's design section, not measured in orbit) | Not published for this network | Not published for this network |
| Energy per thousand inferences | About 102 J (0.03 Wh), estimated | 102 J if it completes about 147 per second | 102 J if it completes about 3,400 per second |
| Not included | Board, host computer, preprocessing, idle | Measured power at that rate, memory traffic, host | Measured power at that rate, memory traffic, host |
| Idle cost: each watt held for a 94.5 minute orbit | About 5,700 J, the energy of about 55,000 of these inferences | Same per watt | Same per watt |
| Radiator at 20 degrees C, before absorbed loads | Under 0.01 square meters | About 0.04 square meters at 15 W | About 1 square meter at 350 W |

Calculated from stated values only, for the chip or module alone; the larger devices' rates for this network are not published, so the table gives the rate each would need, not a rate it reaches.

*Source: calculated by the author from Giuffrida and others, IEEE Transactions on Geoscience and Remote Sensing 60 (2022), section IV-B (1 W, about 102 ms per 192 by 192 by 3 tile on 8 of 12 vector processors), and CSET, Onboard AI: Constraints and Limitations (2023), thermal design powers (Jetson Orin Nano 8GB 15 W; H100 PCIe 350 W). Radiator: chapter 1 (emissivity 0.85, 20 degrees C, about 355 W per square meter).*

Two lines of the table matter more than the comparison. Idle power counts for every minute the device is on: one watt held for a 94.5 minute orbit is about 5,700 joules, the energy of about 55,000 of Phi-Sat-1's inferences. And nearly all the electrical energy a processor draws ends as heat inside the spacecraft. By the arithmetic of chapter 1, a radiator at 20 degrees C

emits at most about 355 W per square meter before the sunlight, albedo and Earth infrared it absorbs; a device that dissipates 350 W needs about a square meter at that temperature, and a 15 W module about 0.04 square meters. Chapter 11 works the heat balance with its signs.

## Radiation and a model

Any program can be corrupted silently by an upset, and many crash instead, which is seen. A neural network makes silent corruption harder to catch, because its answer has no exact reference to check against. A flipped bit in a stored weight or in a buffered value can move an output a little, enough to change a decision, while the program runs to completion and reports nothing. The danger is a plausible wrong answer with no error flag.

What a mission can know comes from ground tests and in-flight self-tests. Before Phi-Sat-1 flew, its processor was tested at European facilities. The team reports no latch-up at a linear energy transfer (LET) of up to 8.8 MeV cm2/mg, with further heavy-ion tests still being analyzed when they wrote; an upset cross-section for the in-package memory, given without units or test conditions and so not usable for a rate; and no dose effect observed under cobalt-60 up to 49 krad, the material not stated. In orbit, a built-in self-test ran in four phases over 70 days. Every diagnostic passed except a test of an SD card that the mission did not use, which showed errors of 2 and 3 bits in two readbacks. In one phase a sample network ran for 240 seconds, and every output matched its reference values. The team concludes that the device ran without functional upsets or degradation from radiation over that period.

That is useful, and it is bounded: one device, one low orbit, 70 days, and tests designed to catch upsets in the hardware, not errors in the network's answers. The mitigations are the ones chapter 4 sets out, applied to a model: checksums over the stored weights and a reload from a protected copy, running an input twice and comparing (which catches a transient fault that affects the two runs differently, not a corrupted weight or input both runs share), checking outputs against simple rules (a cloud mask computed on an image taken in darkness, which a reflected-light camera cannot judge, is flagged rather than used), and a supervisor that falls back to sending data unprocessed, with acquisition cut back so that the store does not fill. Each costs energy, time or memory, and each catches some faults and not others. For an accelerator, the request list of chapter 6 applies, with one question more: did the tests run the model itself, with its external memory, or only diagnostics?

## Is the answer still right?

The second threat to a model in orbit is not a particle. It is that the data it sees is not the data it was trained on. Phi-Sat-1 shows the problem at its plainest. With too little data from its own camera before launch, the team built a training set by resampling images from another satellite's archive to the new camera's resolution and noise, and wrote that this was "a potential source of inaccuracy in the in-flight inference results".

On that synthetic set, the network classified pixels with 88.4 percent accuracy and whole tiles with 95.1 percent; the mission's requirement was false positives, useful images wrongly marked cloudy, under 1 percent of images, and the synthetic test gave 0.8 percent. In orbit, 17 acquisitions gave 612 tiles, with 96 percent accuracy and no false positives. The team then states the limit of that result: the acquisitions were planned for applications, not "to challenge the NN", so no clouds over snow or salt lakes were imaged, cases that the synthetic test set contained. The threshold that decides "cloudy" is computed by the on-board processor and can be changed without retraining the network, so an operator can trade lost images against wasted downlink after launch.

A model's answers drift when the sensor ages, the season changes or the scenes differ from the training set. The way to see it is to keep sampling: send down a fraction of what the model discarded and a fraction of what it kept, compare with a ground result, and track the difference over time. The sample costs downlink, which is the resource the model was meant to save; how large a sample is enough is a line on the trade sheet.

## Two sheets

The two sheets below follow the fifteen lines of chapter 1, shortened to the lines that decide. Sheet A takes its inputs from Phi-Sat-1's published values where they exist. Sheet B is illustrative: its deadline, its paths and its device are assumptions, chosen to show which line decides when the objective is time.

**Sheet A: cloud screening on a low-power accelerator (inputs from Phi-Sat-1 where published)**

| Line | On-board case | Evidence |
|---|---|---|
| 1 Objective and payer | Less downlink spent on cloudy images; for the operator and the images' users | Operator claim (mission team) |
| 2 Workload and quality | Per-pixel cloud mask on 192 by 192 tiles in three bands; an image is "cloudy" above 70 percent cloudy pixels. Requirement: false positives under 1 percent of images | Operator claim (mission team, peer-reviewed paper) |
| 3 Data | About 30 percent of bandwidth saved, by the chip maker's statement | Operator claim (chip maker) |
| 4 Link | Saves the downlink of discarded images; adds the upload of the model and its threshold. Phi-Sat-1's uplink rate is not in the sources read | Not published |
| 5 Energy | About 1 W for about 102 ms per tile: about 0.1 J per tile, about 3.7 J per image of 36 tiles; preprocessing on the camera's computer not published | Calculated from the paper's stated values |
| 6 Heat | About 1 W while running, the chip's nominal power, assumed to be what it draws; the satellite's radiator capacity is not in the sources read | Operator claim (paper), nominal; radiator not published |
| 7 Radiation | Ground tests (latch-up, upsets, dose to 49 krad); 70 days of passing self-tests in orbit (one unused memory card excepted) and a reference network whose outputs matched; no test of whether the cloud answers stayed right under radiation | Measured by the mission team (peer-reviewed paper) |

| Line | On-board case | Evidence |
|---|---|---|
| 9 Updates and trust | A new network and a new threshold can be sent; the threshold changes without retraining | Operator claim (peer-reviewed paper) |
| 12 Uncertainty | In-orbit accuracy from 612 tiles that did not include clouds over snow or salt lakes | Measured by the mission team; limit as its authors state |
| 13 Sensitivity | The false positive rate on the scenes the satellite actually sees | |
| 15 Verdict | Advantageous for the operator, against downlink spent on cloudy images, if the false positive rate stays under the requirement on the scenes actually imaged and the downlink saved exceeds what checking adds (masks, thumbnails and discarded samples sent for audit). The published in-orbit sample (no false positives in 612 tiles from 17 acquisitions, a small and correlated sample) is consistent with that for the scenes it contained, but the tiles are not images and are not independent, so the image-level requirement is not verified by it: with no failures, showing a rate under 1 percent at 95 percent confidence takes about 300 independent clear images (3 divided by 300, calculated). It says nothing yet about the hard cases | |

**Sheet B: ship detection with a 30 minute decision deadline (illustrative; all inputs are assumptions unless sourced)**

| Line | On-board case | Ground case |
|---|---|---|
| 1 Objective and payer | An alert with position and a thumbnail must reach a coastal agency within 30 minutes of imaging; for the agency | |
| 2 Workload and quality | A detector runs over all sea imagery; missed ships are silent, false alarms cost the agency time | A detector on the ground, with more time and a larger model: a different design, with its own missed-detection rate |
| 4 Link | An alert of about 50 kB needs a path to the ground within the deadline. With only the 4 scheduled contacts a day of chapter 1, the wait can be hours. With a low-rate relay path available within 15 minutes (assumed), the alert can arrive in time if detection, transfer and delivery fit the other 15: 50 kB in 15 minutes needs at least 444 bit/s (calculated) | The image waits for the next scheduled contact, then for processing: hours at 4 contacts a day |
| 5 Energy | A module drawing an assumed 15 W (its thermal design power) running 30 minutes an orbit over the sea: about 7.5 Wh an orbit, about 114 Wh a day (15.2 orbits of 94.5 minutes), about 8 percent of chapter 1's assumed 60 W orbit average (1,440 Wh a day) | Nothing on board beyond storage and the radio |
| 7 Radiation | A commercial module; rate not cited (no environment run or device test here), so stated as a risk. Outage: a reset inside the deadline. Hazard: a ship missed in silence | Not on board |
| 9 Updates and trust | The model is uploaded and must be authenticated and reversible | Changed on the ground |

| Line | On-board case | Ground case |
|---|---|---|
| 13 Sensitivity | Whether a path for the alert exists within the deadline; then the detector's missed-detection rate, which no flown source read publishes | |
| 15 Verdict | Advantageous for the agency, against the 30 minute deadline, only if a path for the alert exists and detection, wait, transfer and delivery together fit the deadline, and the detector meets a missed-detection limit the agency sets. With only scheduled contacts, detecting on board does not remove the wait for the next contact, though the small alert then goes down in seconds where the image takes minutes; against a 30 minute deadline it buys little | |

Sheet B's verdict depends first on the path, and only then on the processor's detection time. On-board detection makes the message small; it does not make the link available. A decision deadline is met by the path for a small message, and the trade only starts when that path exists.

**FAILURE NOT ON RECORD: THE SILENT WRONG ANSWER**

The failure this chapter has to name is one that its sources do not show: a model in orbit giving plausible wrong answers that nobody saw. In the sources read as of 23 November 2026 (the source library's papers on on-board inference, the Phi-Sat-1 mission paper, operator pages and trade press on the systems of figure 10.1), no public flight record of in-orbit model error rates was found beyond Phi-Sat-1's evaluation on 612 tiles. That is not evidence that no such errors occurred. It means that, apart from Phi-Sat-1's small sample, a buyer cannot point to a published rate for a flown system in these sources. Phi-Sat-1's own record shows the shape of the gap: a hardware self-test that passed for 70 days, a network test of 240 seconds, and an accuracy figure from scenes that, by its authors' statement, did not include the hard cases. Read narrowly, the record says this much: a hardware test shows that the device computed what it was told; only a sample checked against an independent reference shows that the answer was right.

**SECURITY NOTE**

**Who can interfere or command.** The five questions, for a model: (1) who may upload a model or change its threshold; (2) who can read the model and its inputs, which matters on a platform shared by several customers; (3) who can interfere, through a model poisoned before upload, through inputs crafted in the scene itself, or through the code that loads the model; (4) how a model file is authenticated, which proves its origin and that it was not altered, and does not prove that it is correct; (5) how the previous model is restored. **How it is seen.** A poisoned model and a drifting one look alike in orbit: both give wrong answers without an error flag. The same sample of discarded and kept data that shows drift is the instrument that can show tampering, and attribution needs the upload log. **How it is recovered.** A protected copy of the last good model, a command to fall back to it or to send data unprocessed, and a command path that does not run through the model's own software.

**Monday rule:** for every on-board model, ask who checks its answers, on what sample, and what happens to the data it threw away.

CHAPTER ELEVEN

# When Processing in Orbit Is Worth It: The Trade Study




On 29 April 2026 a JPL physicist, Slava Turyshev, posted a study of orbital data centers (a preprint, not peer reviewed). For a node delivering one megawatt to its computers he found about 5,640 square meters of solar array and 2,500 square meters of radiator, and concluded that computing in orbit for users on the ground "closes only for low Earth-coupled communication intensity, high effective utilization, long delivered lifetime, and very low combined launch-plus-build cost". A CubeSat launched in 2020 had shown a different case: a cloud screen on a vision processor drawing under 2 W that chose which images to send. The two address different workloads and rest on different evidence. This chapter fills the trade sheet of chapter 1 four times, with numbers where the sources give them and the missing deciding evidence named where they do not, and asks of each workload: is processing in orbit advantageous, for whom, and against which objective?

## The method

The sheet has fifteen lines, from the objective to the verdict (chapter 1 sets it out; Appendix A has it blank). The method has four rules.

- **Fill it twice.** Once for the on-board case and once for the ground case, the same job done after the downlink. A sheet with one column is an advertisement.
- **Mark every number.** Each value carries a class: measured (by whom), calculated (from what), simulated, an operator claim, a standard, published history, or an assumption to verify. An illustrative sheet says so in its title.
- **Write the verdict in its form.** "Advantageous for [whom], against [objective], if [assumption]"; or "not advantageous"; or "cannot be decided from public evidence". The verdict names the line that would change it.

- **Say when the sheet is not enough.** A first-order sheet treats each line alone. When two lines interact (a battery sized for the radio and the computer together, a radiator that also cools the instrument, a schedule shared with other payloads), a system model is needed, and the sheet says which interaction it did not model.

The sections that follow work through the lines that carry most of the arithmetic: energy, heat, radiation, the link and money. The last fills the four sheets.

### Energy: a watt at the board costs more than a watt of array

A computer in low orbit runs on its satellite's solar arrays in sunlight and on its battery in the Earth's shadow. Three things make a watt at the board cost more than a watt of panel.

- **Eclipse.** In a circular orbit at 550 km with the Sun in the orbit plane, the worst case, the satellite is in shadow for 35.6 of its 95.6 minutes, 37 percent of the orbit (calculated from the orbit's geometry with a cylindrical shadow). A load that runs through the eclipse must be carried by the battery, which the array must recharge in the 60.0 minutes of sunlight.
- **Losses.** Power is conditioned and distributed before it reaches the board, and a battery returns less than it took. The sheet assumes 90 percent for each; a bus maker's data settles both.
- **Degradation.** Arrays lose output over the mission from "aging, radiation, and contamination", as NASA's State of the Art of Small Spacecraft Technology report puts it in its 2026 power chapter, which gives multi-junction space cells "a nominal efficiency of 30%". The sheet assumes a sun-pointed panel, 30 percent cells, 85 percent for packing, wiring and temperature, and 90 percent at end of life: about 312 W per square meter from a solar constant of 1,361 W per square meter.

Figure 11.1 applies this to an assumed 20 W computer. Run through the whole orbit, it needs 31.9 Wh per orbit at the board, 11.9 Wh of it in eclipse, which the battery must deliver as 13.2 Wh once the distribution loss is counted, and 36.9 W of array in sunlight: 1.84 W of array for each watt at the board. Run only in sunlight, it needs 22.2 W, 1.11 W per watt. Run for 10 minutes an orbit, it needs 22.2 W while it runs if the array feeds it directly, or 3.7 W averaged over the sunlit time if a lossless buffer supplies the peak, about 4.1 W if the energy passes through the 90 percent battery (calculated). The cells for the eclipse are light, 0.049 to 0.088 kg of lithium-ion cells at the 150 to 270 Wh per kg that NASA's power chapter gives, before depth-of-discharge, temperature, aging and packaging allowances multiply it for a long life.

FIGURE 11.1

## Energy over one orbit for an assumed 20 W computer at 550 km: running through eclipse costs 1.84 W of array per watt

| Line | Runs the whole orbit | Runs in sunlight only | Runs 10 minutes per orbit, in sunlight | Class |
|---|---|---|---|---|
| Orbit period; sunlit and eclipse time | 95.6 min; 60.0 min sunlit, 35.6 min eclipse (37 percent) | same | same | calculated |
| Energy at the board, per orbit | 31.9 Wh | 20.0 Wh | 3.3 Wh | calculated |
| Eclipse energy: at the board; out of the battery | 11.9 Wh; 13.2 Wh | none | none | calculated |
| Array power needed in sunlight | 36.9 W | 22.2 W | 22.2 W while it runs (direct); 3.7 W averaged through a lossless buffer, about 4.1 W through the battery | calculated from assumptions |
| Array watts per watt at the board | 1.84 | 1.11 | 1.11 while it runs | calculated from assumptions |
| Array area, at 312 W per square meter | 0.12 square meters | 0.071 square meters | 0.071 square meters (direct); 0.012 to 0.013 (buffered) | calculated from assumptions |

| Line | Runs the whole orbit | Runs in sunlight only | Runs 10 minutes per orbit, in sunlight | Class |
|---|---|---|---|---|
| Battery cells for the eclipse, at 150 to 270 Wh per kg | 0.049 to 0.088 kg, before depth-of-discharge and other allowances | none | none | calculated |

An illustrative energy balance, worst-case eclipse (Sun in the orbit plane), with 90 percent distribution and battery efficiency and a sun-pointed array delivering 312 W per square meter at end of life, all assumptions to verify.

*Source: calculated by the author from the inputs listed (orbit from IERS GM and the WGS 84 radius; solar constant from IAU 2015 Resolution B3; cell efficiency and battery energy density from NASA, State of the Art of Small Spacecraft Technology, 2026, power chapter); board power, efficiencies, packing and end-of-life factors are assumptions. Author's table.*

Two lessons hold beyond these assumptions. The first is that the duty cycle is a design choice: a job that can wait for sunlight needs 1.11 W of array for each watt at the board, against 1.84 for one that cannot. That is about 40 percent less per watt drawn, not per job: to do the same work in 60.0 sunlit minutes, assuming the same energy per completed job, the board must draw about 31.9 W while it runs, which needs about 35.4 W of array, only about 4 percent below 36.9 W, with more heat to reject while it runs (calculated). Waiting for sunlight mainly saves the battery and its cycles. The second is that "solar power in orbit is free" is wrong twice: the array has a mass and an area to launch and point, and every watt drawn in eclipse is paid for with battery mass and cycles. An orbit chosen to avoid eclipse, a dawn-dusk sun-synchronous orbit for example, removes most of the battery line; the 2026 studies of large nodes read for this chapter assume orbits with long sunlight for that reason.

## Heat: the balance, with its signs

"In a vacuum, heat is transferred only by radiation and conduction with no convection," NASA's 2026 thermal chapter states. A computer's electrical input becomes heat on board almost entirely; unlike a transmitter, it sends little of its energy out as a signal. That heat is conducted from the chip through its board, its housing and the structure to a radiator, and the radiator emits it to space. "Space is cold, so cooling is free" fails on both steps.

The radiator's balance, per square meter, is written with its signs:

- **Emitted (positive):** $\varepsilon\sigma T^4$, where $\varepsilon$ is the coating's emissivity, $\sigma$ is the Stefan-Boltzmann constant and $T$ the radiator temperature in kelvin.

- **Absorbed (negative):** sunlight, sunlight reflected by the Earth (albedo) and the Earth's own infrared, each weighted by the coating's absorptivity and by the view factor, how much of the radiator sees each source. A radiator turned away from the Sun still sees the Earth from low orbit.
- **Net:** what the radiator can carry away. The area needed is the heat load divided by the net.

Figure 11.2 works the balance for a radiator with emissivity 0.85 and an assumed 50 W per square meter absorbed, at three radiator temperatures. At 300 K (27 degrees C) one square meter emits 390 W and rejects 340 W net, so the 20 W computer needs 0.059 square meters, a 100 W board 0.29 and a kilowatt 2.9. A cooler radiator needs more area: at 273 K the kilowatt needs 4.6 square meters.

FIGURE 11.2

## The heat balance of a radiator, sunlit steady state: a kilowatt needs about 2.9 square meters at 300 K

| Line, with its sign | 273 K (0 degrees C) | 300 K (27 degrees C) | 330 K (57 degrees C) | Class |
|---|---|---|---|---|
| Emitted, plus: emissivity times the Stefan-Boltzmann constant times T to the fourth | 268 W per square meter | 390 W per square meter | 572 W per square meter | calculated |
| Absorbed, minus: Earth infrared, albedo and stray sunlight | 50 | 50 | 50 | assumption, to verify |
| Net rejected | 218 W per square meter | 340 W per square meter | 522 W per square meter | calculated |
| Area for 20 W | 0.092 square meters | 0.059 square meters | 0.038 square meters | calculated |
| Area for 100 W | 0.46 square meters | 0.29 square meters | 0.19 square meters | calculated |
| Area for 1 kW | 4.6 square meters | 2.9 square meters | 1.9 square meters | calculated |

| Line, with its sign | 273 K (0 degrees C) | 300 K (27 degrees C) | 330 K (57 degrees C) | Class |
|---|---|---|---|---|
| Check: 350 K, emissivity 0.92, nothing absorbed | 783 W per square meter emitted: 224 to 351 square meters for 175 to 275 kW, against "at least" 225 to 350 in the preprint | | | calculated; inputs from a preprint (not peer reviewed) |

Emitted minus absorbed, per square meter of radiator emitting from one face, at emissivity 0.85 with 50 W per square meter absorbed (both assumptions to verify), and the area each heat load needs; eclipse transients and the conduction path need a thermal model.

*Source: calculated by the author with the CODATA 2018 Stefan-Boltzmann constant; emissivity and absorbed heat are assumptions; the check row uses Hu, Alhassan and Iqbal, arXiv 2610.32260 (preprint, not peer reviewed). Author's table.*

Three things this sunlit steady state does not show must be stated beside it. The radiator must be colder than the chip, and the conduction path from chip to radiator takes some of the temperature difference: a hotter radiator is smaller, but only if the chip can run hotter still. The balance changes with attitude and season. In eclipse, direct sunlight is gone and the sunlight reflected by the Earth (albedo) falls as less of the sunlit Earth is in the radiator's view, while the Earth's infrared remains and the computer may keep running; those transients need a thermal model. And emitting area is not footprint: a radiator that emits from both faces needs half the panel area, if neither face sees the Sun or the Earth much.

The same arithmetic scaled up explains the studies. Rejecting one megawatt at 300 K with these assumptions needs about 2,940 square meters of radiator; Turyshev's base case for a one megawatt node gives 2,500 square meters, on his own assumptions. Hu, Alhassan and Iqbal (a 2026 preprint) state that rejecting 175 to 275 kW at 350 K with emissivity 0.92 needs at least 225 to 350 square meters; emission from one face with nothing absorbed gives 783 W per square meter and 224 to 351 square meters, so their figure is that lower bound, and they add 25 to 67 percent for absorbed heat. These are research figures for designs not built. What they share is the scaling: at the same net rejection per square meter (the same

temperature, coating, absorbed heat and geometry), radiator area grows in proportion to the heat, and a node that rejects a thousand times more heat needs a thousand times the radiator.

### Radiation: line 7, carried from chapters 3 and 4

Line 7 of the sheet takes what chapters 3 and 4 establish: the dose and single-event environment of the orbit, from a model run with its inputs; the part's test data, with its lot, its conditions and its failure criterion; and the mitigation, with what it leaves. Raw upsets, corrected errors, outages and hazardous failures are counted apart. Where those inputs are not cited, the line is qualitative, with its reasons.

For the workloads of this chapter the line asks three things. How does the computer fail: a cloud screen designed to forward everything when it stops avoids deletions by the classifier, but only if storage and contacts can absorb the extra data, while a screen that upsets silently may discard good images. Which fallback a mission has must be read in its documents, not assumed. Does the result carry its own check: a detection can be confirmed on the ground from a small cropped image (an image chip), while a computed result for a customer on the ground may have no independent check. And how long must the part last: one public dose test of a data center processor, Google's test of its Trillium chip in a 67 MeV proton beam, found "No hard failures were attributable to TID up to the maximum tested dose of 15 krad(Si) on a single chip". That is one chip and one kind of effect; chapter 10 sets out what it does not say about upsets or lifetime.

### The link: what processing removes, adds and delays

Processing on board changes the link in four ways, and only the first is the one usually quoted.

- **It removes data from the downlink** when it discards, compresses or replaces raw data with a result. Chapter 9 keeps the three apart: lossless compression keeps all the information, lossy compression keeps an approximation, and filtering throws data away for good.
- **It adds data to the uplink:** new software, new models and the jobs themselves. A model upload competes with commands on a command link that is often much narrower than the downlink (chapter 10).
- **It changes the delay.** What decides the delay is usually the wait for the next contact, not the transfer. A result ready on board still waits for a link.
- **It saves radio energy** when it shortens transmission, and spends processor energy to do so. Chapter 1's example found the processor costing more energy than the radio saved.

The budgets of the hops themselves belong to the first book of this series, *The Bit's Path*, and the prices of contact minutes to the fifth, *Rent Before You Build*. The sheet takes both as inputs.

## Mass, money and lifecycle

A launch price enters the sheet only as a dated booking minimum, with what it leaves out. NASA's 2026 chapter on integration and launch gives SpaceX's Transporter rideshare as "starting at $350k for ~50 kg"; that is 7,000 dollars per kg at 50 kg (calculated), and the fifth book of this series records what it leaves out: deployment hardware, insurance and integration. It is not the cost of a satellite.

Turyshev's economic test makes the scale plain. With about 40 kg of node for each kilowatt delivered to the computers (his base case gives 34 to 59 kg per kW with the fixed spacecraft mass) and a terrestrial benchmark of 10,000 to 40,000 dollars per kW of data center, he finds that only 250 to 1,000 dollars per kg can be spent on launch and spacecraft build together before communications, operations and lifetime are counted. Against the rideshare minimum above, a booking for a small satellite and not a price for launching large nodes, the launch alone is 7 to 28 times that allowance, before building anything. Van Berkel (Eindhoven, 2026 preprint) reaches a related conclusion from the network side: inference in low orbit "may be feasible", while training frontier-scale language models in orbit "is unlikely to be competitive" with data centers on the ground. Both are studies with stated assumptions, and both change if launch prices fall far; neither is a verdict on any company.

The rest of the lifecycle is mostly "not published": the engineering and qualification of a flight computer, operations staff and ground stations, and replacement when the hardware wears out or becomes obsolete. On the ground a failed or outdated server is replaced in place; on an unserviced satellite in orbit it is replaced by launching another.

## Four sheets

The four workloads below were chosen because they differ on the line that decides. Each sheet fills the on-board and the ground case; numbers are sourced or marked as assumptions. Figure 11.3 sets the four side by side.

**Sheet (a): cloud screening of optical imagery, with Phi-Sat-1's published inputs.** Phi-Sat-1 flew on one of ESA's two FSSCat CubeSats from September 2020. Its team's paper in IEEE Transactions on Geoscience and Remote Sensing (2022) reports a network running on an Intel Movidius Myriad 2 vision processor, a device the paper describes as "maintaining the power consumption well below 2 W", that labels an image cloudy when more than 70

percent of its pixels are cloud, and 96 percent accuracy "when performing cloud detection on live images on-satellite". Intel, the chip maker, said the screen saved about 30 percent of bandwidth.

**Sheet (a): cloud screening, on board and on the ground (published inputs where they exist; the rest marked)**

| Line | On board | On the ground |
|---|---|---|
| 1 Objective and payer | Useful images delivered per unit of downlink, for the operator and its users | |
| 2 Workload and quality | A network classifies each image; 96 percent accuracy on live images (paper); 0.8 percent of images wrongly called cloudy on the synthetic test set (paper); in flight, no false positives in 612 tiles checked tile by tile (chapter 10), a small sample and not an image-level rate | The same or a better classifier on the full data, after downlink |
| 3 Data | Images judged cloudy are not sent; the chip maker claims about 30 percent of bandwidth saved (the share of images and of contact time is not stated) | Every image is sent |
| 4 Link | Less downlink time: about 30 percent if the claimed bandwidth saving maps to contact time (assumption, to verify); a model upload when it changes | All images through the contacts available |
| 5 Energy | The vision processor "well below 2 W" while it runs (paper); run time per image not in the sources read | Radio energy to send the images the screen would discard; transmitter power not in the sources read. The energy advantage cannot be decided from these inputs |

| Line | On board | On the ground |
|---|---|---|
| 6 Heat | The processor under 2 W while running (paper); board and conversion losses not published; the satellite's thermal margin was not in the sources read | None added in orbit |
| 7 Radiation | Rate not cited for this orbit (chapter 3). Ground tests of the processor: no latch-up observed to LET 8.8 MeV cm2/mg, no dose effect observed to 49 krad (chapter 10). Outage: a reset stops the screen. Hazard: a silent upset that discards a good image | None in orbit |
| 8 Mass and volume | One small board, in the camera's processing chain | None |
| 9 Updates and trust | Who uploads a new model, how it is authenticated, how the old one is restored | Ground software, changed on the ground |
| 10 Lifecycle cost | Not published | Not published |
| 11 Ground alternative | Feasible whenever the contacts can carry every image. When they cannot, the ground has other tools: storage and later delivery, acquisition scheduling, priorities commanded from the ground, compression. What the mission does with images it cannot send is an assumption, to verify in its operations documents | |
| 12 Uncertainty | The 30 percent is an operator claim and depends on season and region; the 96 percent is the paper's in-flight figure for its test set | |
| 13 Sensitivity | The cloudy fraction, and the rate at which the screen discards an image a user wanted | |
| 14 Assumptions and limits | The downlink's real constraint and the satellite's power budget were not read; a mission model would combine them | |

| Line | On board | On the ground |
|---|---|---|
| 15 Verdict | Advantageous for the operator, against useful images per unit of downlink, if the downlink is the binding constraint, the ground's other tools do not already meet the need, and the false discard rate is acceptable to the users. Not advantageous for an archive that must keep every raw image | |

**Sheet (b): detection with a decision deadline.** The objective here is not volume but time: a user must learn of a ship or a fire within a deadline. All inputs are assumptions, chosen to show which line decides. A 4 Gbit scene takes 40 seconds at 100 Mbit/s; a 10 kbit alert (position, time, class and a small image chip) takes 4.2 seconds on an assumed 2.4 kbit/s messaging link, counting the transfer alone, with access, headers and retries on top. Both are short. What decides is the time from capture to the user: detection run time, the wait for a link, transfer, ground processing and delivery.

**Sheet (b): detection with a decision deadline (illustrative; inputs assumed, times calculated from them)**

| Line | On board | On the ground |
|---|---|---|
| 1 Objective and payer | Time from capture to the user's decision, against a stated deadline; for the user who must act | |
| 2 Workload and quality | A detector runs on each scene; its misses and false alarms are the quality line | The same detector, or an analyst, on the full scene |
| 3 Data | An alert of about 10 kbit per detection | The 4 Gbit scene |
| 4 Link | 4.2 s on a 2.4 kbit/s messaging link, if one is available within the deadline | 40 s at 100 Mbit/s, at the next ground contact |
| 5 Energy | The detector's power for each scene | Radio time for the scene |
| 6 Heat | Small for a low-power detector | None added |
| 7 Radiation | Rate not cited. Outage: a reset during the deadline delays the alert. Hazard: a missed detection from an upset is silent, while a false alarm is visible and checkable from the chip | None in orbit |

| Line | On board | On the ground |
|---|---|---|
| 8 Mass and volume | A processor and, if not already on board, a messaging terminal | None |
| 9 Updates and trust | Detector updates; who may send alerts on the user's behalf | Ground software |
| 10 Lifecycle cost | The messaging service, if bought | Contact time |
| 11 Ground alternative | Meets the deadline when the next contact, the scene's transfer, ground detection and delivery all fit inside it | |
| 12 Uncertainty | The wait for a contact varies from zero to many minutes with orbit and ground network; every input here is an assumption and every time is calculated from them | |
| 13 Sensitivity | The availability of a low-rate link within the deadline; ground processing and delivery time; the detector's miss rate | |
| 14 Assumptions and limits | The distribution of contact gaps needs an orbit and ground network model | |
| 15 Verdict | Advantageous for the user, against capture-to-user time, if a low-rate link is available inside the deadline when ground contacts are not, and on-board detection, transfer and delivery fit the deadline. Without such a link both wait for the same contact; the remaining advantage is the scene's transfer and ground processing time, and whether it matters depends on the deadline | |

**Sheet (c): regeneration in a communication payload.** A regenerative payload demodulates and decodes on board, routes, then codes and modulates again (chapter 8); a transparent one filters, shifts and amplifies. The comparison must be of complete payloads delivering the same service, not of a processor against nothing. The link consequence is taken from the first book of this series: on a transparent path the uplink's and downlink's noise add, and the circuit's verdict needs the combined line, while on a regenerative path each hop closes or fails on its own at the decoder (*The Bit's Path*, chapters 4 and 8). That is a gain in how the budget is divided, not "bits removed".

**Sheet (c): regenerative against transparent, at equal service (qualitative where no public comparison was read)**

| Line | Regenerative | Transparent |
|---|---|---|
| 1 Objective and payer | Service delivered per unit of satellite power and mass, over the satellite's life, for the operator | |
| 2 Workload and quality | Demodulate, decode, route, code and modulate each carrier | Filter, shift and amplify; switching of whole channels possible |
| 3 Data | The same traffic | |
| 4 Link | Each hop closes alone; on-board routing between beams | The combined line of two hops decides; switching of channels or beams on board is possible, routing of decoded packets is not |
| 5 Energy | Processor power, against amplifier power that may change with the architecture | No regenerative processor; amplifier power, plus the digital processor of a digital transparent payload |
| 6 Heat | Amplifier heat plus the processor's; the amplifiers' operating points may differ | Amplifier heat, plus the digital processor's if there is one |
| 7 Radiation | Rate not cited. Corrected upsets cause no visible error, though correction takes time and power; an outage drops traffic until recovery; a silent routing error is the hazard. Mitigation needed (chapter 4) | Analog transparent parts with fewer state elements; dose still applies (a digital transparent processor has its own upsets) |
| 8 Mass and volume | Processor; possibly less analog hardware | Analog hardware |
| 9 Updates and trust | Waveform and routing may be changeable by command, within what was built, so the command path must be protected | Need not decode the waveform; passes it within the payload's bandwidth and linearity limits |
| 10 Lifecycle cost | Not published for a comparison at equal service | |

| Line | Regenerative | Transparent |
|---|---|---|
| 11 Ground alternative | The transparent column is the ground alternative: decoding and routing in the gateways | |
| 12 Uncertainty | No public comparison of complete payloads at equal service was read for this book | |
| 13 Sensitivity | Complete payload power and mass at equal service; whether the service needs routing between user beams without a gateway | |
| 14 Assumptions and limits | A full payload design comparison is needed | |
| 15 Verdict | Cannot be decided from public evidence. Deciding it needs a comparison of complete payloads at equal service: mass, amplifier operating points, processor power, heat, availability and the gateway alternative | |

**Sheet (d): general-purpose batch computing for customers on the ground.** Here the input starts on the ground, crosses the uplink, is processed in orbit and comes back down. The figures are the cited studies', with their assumptions, and the launch line from the dated minimum above.

**Sheet (d): a batch job for a customer on the ground (inputs from the cited studies and the dated launch minimum)**

| Line | In orbit | In a data center on the ground |
|---|---|---|
| 1 Objective and payer | Cost per unit of computing delivered, for the customer and the provider | |
| 2 Workload and quality | The same job; the same result if the hardware is the same and no upset corrupts it | |
| 3 Data | Input up, result down | Input and result over terrestrial networks |
| 4 Link | Both directions cross the space link; uplink usually the narrower | Terrestrial fiber |
| 5 Energy | About 1.84 W of array per W in continuous running in this low orbit example (Figure 11.1), less in an orbit with long sunlight | Grid power |

| Line | In orbit | In a data center on the ground |
|---|---|---|
| 6 Heat | About 2.9 square meters of radiator per kW at 300 K (Figure 11.2) | Chillers and water or air |
| 7 Radiation | Rate not cited. Dose over the mission and upsets in commercial processors; outages stop jobs; the hazard is a wrong result returned to a customer who may have no independent check | A terrestrial soft-error environment, handled with error correction and retries |
| 8 Mass and volume | 34 to 59 kg per kW delivered, base case (Turyshev) | None launched |
| 9 Updates and trust | All software and keys in a free-flying node that nobody can reach physically | Physical access, replacement |
| 10 Lifecycle cost | For large nodes: the rideshare minimum is 7 to 28 times the 250 to 1,000 dollars per kg the study allows for launch and build (calculated); a booking minimum is not a price for launching large nodes | The terrestrial benchmark of 10,000 to 40,000 dollars per kW (Turyshev) |
| 11 Ground alternative | Generally feasible for such a job; site power, land, permits, jurisdiction and connectivity are among the constraints that can rule a site out | |
| 12 Uncertainty | Launch price and node mass per kW dominate; both are study inputs, not quotes | |
| 13 Sensitivity | Launch plus build cost per kg; utilization; delivered life | |
| 14 Assumptions and limits | The studies model large nodes; a small node shares the scaling but not every number | |
| 15 Verdict | Cannot be decided from this sheet for a particular offer: it compares capital allowances, not cost per completed job. The two cited studies, on their own assumptions for large nodes, find computing for users on the ground not competitive except under the conditions they name | |

FIGURE 11.3

## Four sheets side by side: the line that decides is different for each workload

| Line | (a) Cloud screening | (b) Detection with a deadline | (c) Regeneration | (d) Batch computing for the ground |
|---|---|---|---|---|
| Objective, for whom | Useful images per unit of downlink, for the operator | Delay to a decision, for the user | Service per watt and kg, for the operator | Cost per unit of computing, for the customer |
| Where the input starts | In orbit (the sensor) | In orbit (the sensor) | On the ground (the users' carriers) | On the ground (the customer) |
| What crosses the link | Fewer images | An alert, if a link is there in time | The same traffic | Input up, result down |
| The line that decides | Cloudy fraction and false discards | A low-rate link inside the deadline, and the time from capture to user | Complete payload power and mass at equal service | Launch and build cost per kg, utilization, life |
| Evidence | Peer-reviewed paper; operator claim | Assumptions | No public comparison read | Two preprint studies; a dated launch minimum |
| Verdict | Advantageous for the operator, if the downlink binds, ground tools do not already meet the need, and discards are acceptable | Advantageous for the user, if a low-rate link is available in time and the whole chain fits the deadline | Cannot be decided from public evidence | Cannot be decided from this sheet; the large-node studies find it not competitive on their assumptions |

The four worked sheets of this chapter, reduced to the lines that carry the argument; every verdict is conditional, and none is a verdict on a company.

*Source: sheets (a) to (d) of this chapter and the sources cited there. Author's table.*

Two published studies on the large-node question disagree with the announcers more than with each other. Turyshev calls space-native preprocessing and computing integrated with communications "credible early regimes"; van Berkel finds inference possibly feasible and frontier training unlikely to compete. Both treat communication as a limit, but different communication: Turyshev through a "communication intensity", the data sent between orbit and the ground per unit of energy delivered to the computers, and van Berkel through the bandwidth between satellites that a large training job needs. Where they differ is in the variable each puts first: Turyshev's is mass and cost per kilowatt, van Berkel's is the network a large training job needs. Neither has a node in orbit to test against.

**TRADE-OFF: THE WORKLOAD CHANGES THE DECIDING LINE**

In the operator pages, trade press and research library read for this book up to 23 November 2026, no in-orbit performance record of the systems in chapter 12 was found that a trade sheet could be checked against, and only one failure report; the processor failure Starcloud reported during launch (chapter 12) has no stated cause. The lead case is therefore a trade-off, and it is the one that sheets (a) and (d) set up. Put a commercial processor on a satellite. For a job whose input is already in orbit, the images of sheet (a), the processor sits between the sensor and a narrow downlink, and it can be advantageous on the downlink line with a processor drawing under 2 W. For a job whose input starts on the ground, sheet (d), the same processor must receive its input over the uplink and be powered by an array and battery and cooled by a radiator that a data center on the ground does not need; the published studies of large nodes find that case uncompetitive on their assumptions, and the sheet cannot decide it for a small one. What fails is not the hardware. It is the general claim, "compute in orbit is advantageous" or its opposite: on these two sheets the deciding line differs, so a single answer for both jobs does not follow.

**SECURITY NOTE**

**Who can interfere or command.** Line 9 of the sheet asks the five questions of chapter 1. Moving a computation onto an unserviced free flyer moves its code, its model and its keys to a place where nobody can reach them physically: a key can be revoked or replaced from the ground, and erased by command if the design allows, but a compromised board cannot be trusted to obey that command, and on an unserviced free flyer a board cannot be replaced. Who is authorized to upload software or a model, and who can read what the computer holds and sends, must be answered for the platform and for each customer's job. Interference can come through the command link, through hosted code, or through the supply chain of a commercial part or a model. **How it is seen.** An upload is authenticated by signature; a signature proves origin and integrity, not that the model or the code is correct. A screen that discards data on board hides its own errors unless a sample of what it discarded is sent down. Detection of an anomaly, its diagnosis, attribution to a cause and recovery are separate steps. **How it is recovered.** A protected known-good image, a rollback to the previous model, and a command path that hosted code cannot reach. Where an offer does not state these, the sheet writes "not published" on line 9 and the verdict carries that condition.

**Monday rule:** never accept "compute in orbit" or "bandwidth is the scarce resource" as a general statement; ask for the workload, the objective and the sheet.

CHAPTER TWELVE

# Computing Platforms in Orbit: Launched, Demonstrated, Offered, Announced




Axiom Space states that it launched its first two orbital data center nodes on 11 January 2026 with Kepler's first satellites; its page rates their optical links at 2.5 gigabytes per second in its text (2.5 Gbps in a linked title) and says they support software and model updates over the air. Three months later, on 13 April 2026, trade press reported Kepler's cluster at about 40 Nvidia Orin processors on 10 satellites, with 18 customers. Ten months after the launch, no public record of what those nodes have computed in orbit could be found. This chapter asks one question of every computing platform in orbit: what does the public record show can be used, by whom, and what is still a plan? It answers with status fields, not status words. For most systems no source speaking for November 2026 could be read, and each such field says so; the chapter is a method for writing the fields as much as a record of them.

## What counts, and how it is written

A computing platform in orbit, in this chapter, is a computer in orbit that is shared, scheduled from the ground and offered as a service to people who load their own software. Experiments, prototypes and announcements that do not yet meet that definition are included as comparison cases, and labeled so. The second book of this series, *From Telstar to the Phone in Your Pocket*, tells in its chapter 13 how processing moved on board; this chapter does not retell it. It reads the platforms of November 2026 the way a buyer or an assessor must read them.

One system can be in several states at once, so each state gets its own field, with a date and a source:

- **Launched:** on a named vehicle, on a date.
- **Demonstrated function:** what a source says the platform did in orbit, and who the source is.

- **Current health:** a source that speaks for the book's date. An announcement from January does not.
- **Service access:** who can apply or buy, in which country, since when.
- **Announced:** a plan, in the announcer's words, with its date.
- **Retired, failed or deorbited.**

"No public record found" is written with its scope and its date, because it says something about the search, not about the system. A launched platform with no demonstrated function on record may be working well; the reader simply cannot know it from public sources. Figure 12.1 sets every system of this chapter against these fields.

FIGURE 12.1

## Computing platforms in orbit, field by field, as of 23 November 2026

| System and who runs it | Launched | Demonstrated in orbit, and who says so | Health on the book's date | Access on the book's date | Announced; retired or failed | Source, date read |
|---|---|---|---|---|---|---|
| OPS-SAT (ESA) | December 2019 (ESA) | The NanoSat MO Framework executed in space in 2020 (ESA) | Not checked | | End date not read | ESA, NanoSat MO Framework documentation, 2026-11-23 |
| OPS-SAT PRETTY, Space Lab (ESA) | Not stated on the page read | | Not checked | "Available now"; registration, no charge; 50 percent of mission time to the lab (ESA) | | ESA, OPS-SAT Space Lab page, 2026-11-22 |
| OPS-SAT VOLT and ORIOLE (ESA) | | | | | "Launch '27" (ESA) | ESA, OPS-SAT Space Lab page, 2026-11-22 |
| Spaceborne Computer-2 (HPE), on the space station | 20 February 2021 (survey) | Edge computing and AI experiments (survey) | Not checked | | | Hu and others, arXiv 2610.32260, preprint, 2026-11-23 |
| Software in Space (Spire) | | | Not checked | Sold for a flat monthly fee, amount not stated (Spire) | | Spire, Space Services page, 2026-11-22 |
| On-Orbit AI (Loft Orbital) | Operator's fleet count: 16 satellites launched; how many run On-Orbit AI not stated | | Not checked | Offered; no price (Loft Orbital) | | Loft Orbital, home page, 2026-11-22 |
| Kepler cluster | About 40 Nvidia Orin processors on 10 satellites, January 2026 (trade press) | | Not checked | 18 customers on 2026-04-13 (trade press); access on the book's date not checked | | TechCrunch (trade press), 2026-11-23 |

| System and who runs it | Launched | Demonstrated in orbit, and who says so | Health on the book's date | Access on the book's date | Announced; retired or failed | Source, date read |
|---|---|---|---|---|---|---|
| Orbital data center nodes (Axiom Space) | Two nodes, 2026-01-11 (Axiom Space) | No public record found in the sources read | Not checked | By enquiry (Axiom Space) | | Axiom Space, Orbital Data Centers page, 2026-11-22 |
| Data Center Unit-1 (Axiom Space), on the space station | Fall 2025 (Axiom Space) | Purpose stated: "to demonstrate initial capabilities" (Axiom Space); no result found | Not checked | | | Axiom Space, Orbital Data Centers page, 2026-11-24 |
| Starcloud-1 (Starcloud) | November 2025, with an Nvidia H100 (trade press) | Trained a model and ran a Google language model, "according to Starcloud" (trade press) | Not checked | Not a public service | A second processor, an Nvidia A6000, failed during launch (trade press); Starcloud 2 "later this year" (March 2026), not checked | TechCrunch (trade press), 2026-11-23 |
| Project Suncatcher prototype (Google, with Planet) | One prototype on Transporter-18, 2026-10-01 (trade press) | No public record found in the sources read | Not checked | Nothing to rent | Two prototypes by early 2027 (2025-11-04); a New Glenn launch targeted for the first quarter of 2029 (2026-11-19) (Google) | Google Research blog and Google blog, 2026-11-23; Space.com (trade press), 2026-11-22 |
| Three-Body Computing Constellation (China) | First group of 12 spacecraft, summer 2025 (NASA report); May 2025 (a 2026 preprint) | | Not checked | | | NASA, State-of-the-Art of Small Spacecraft Technology 2026; builder's page sought, not read; Hu and others, arXiv 2610.32260, and Sun and others, arXiv 2605.12681 (the May date), preprints |
| Data center satellites (SpaceX) | | | | | An FCC application of February 2026, as reported; the filing was not read | van Berkel, arXiv 2607.14172, preprint |

Each system against the status fields, with who states each entry; an empty field means no source read for this book speaks for it, and "not checked" means no source speaking for November 2026 was read.

*Source: the pages named in the last column, read on the dates given; operator statements are the operators' words, trade press is labeled, preprints are not peer reviewed. Author's table.*

## Experiments: a station, and a laboratory in orbit

**A server on a crewed station.** Hewlett Packard Enterprise's Spaceborne Computer-2, commercial servers with a graphics processor, is listed by a 2026 research survey (a preprint, not peer reviewed) as launched to the International Space Station on 20 February 2021 and as having hosted edge computing and AI experiments there since. The company's own pages could not be opened for this book, so no result is printed here. Whatever it showed, the setting matters more than the result for this chapter: a pressurized station supplies power from its own arrays, removes heat through its own cooling loops and has a crew that can swap a failed drive. A free-flying node has none of the three. A result from the station is a result about software and parts in that environment, not about a node that must generate and radiate every watt itself (chapter 11).

**A laboratory in orbit.** ESA's OPS-SAT was a CubeSat flown to let outside teams run software in orbit. ESA's documentation of its NanoSat MO Framework states that the mission launched in December 2019 and that the framework was first executed in space in 2020. Its successors are run as the OPS-SAT Space Lab. ESA's page, read on 22 November 2026, offers it with "No charge, contract or paperwork", describes OPS-SAT PRETTY as "a reconfigurable Linux based processing platform, a configurable Software-defined Radio, and a GNSS receiver", gives the lab 50 percent of PRETTY's mission time, and marks PRETTY "Available now". Two further missions, OPS-SAT VOLT and OPS-SAT ORIOLE, are listed "Launch '27": announced, not launched.

The Space Lab is the documented example of the link that this book and the fifth book of this series, *Rent Before You Build*, share: a customer's program on a hosted computer in low orbit. Its ledger follows. In ESA's description of OPS-SAT, the customer never commands the satellite: the experimenter packages the program with an identifier ESA assigns and submits it to ESA's Flight Control Team, which handles it under its operational procedures; it goes up on ESA's command link, and results come back on ESA's downlink. Other hosts' workflows were not published in that detail on the pages read.

**LINK LEDGER · LL-2026-18 · SERVICE LEDGER**

2026-11

System: A customer's program on a hosted computer in low orbit: upload, run and results through the host (ESA's OPS-SAT Space Lab as the documented example)

**In service** | **Class: service only**

| | |
|---|---|
| **1 Requirement** | Run a customer's program on a computer in orbit and get its results back, without owning, licensing or commanding a satellite. |
| **2 Ends and path** | The developer's computer and the host's test tools; the host's mission control, satellite and platform; the customer's results store.<br>**Path:** Store and forward, in OPS-SAT's documented workflow (PRETTY's not read): the customer packages the program; the host's control team builds and sends the upload; results come down on the host's downlink. |
| **3 Orbit and time** | Low orbit, the host's satellite; OPS-SAT PRETTY's orbit was not read.<br>**Time:** Upload, run and return wait for the host's passes and plan; no turnaround is published. |
| **4 Band and permission** | **Upload.** The host's command link, band and authorization; the guest holds none<br>**Results.** The host's downlink, band, authorization; the guest gets a share<br>**Guest's output.** Earth imaging or a radio adds duties: a counsel item |
| **5 Capacity** | **Up.** Size and memory limits: in the host's interface document, not read.<br>**Down.** A share of the host's downlink, set by the host's plan.<br>PRETTY gives the lab 50 percent of its mission time (ESA's page). |
| **6 Who owns each end** | **Space segment.** The host; ESA runs PRETTY's lab; owner, operator not stated<br>**Ground segment.** The host's mission control and stations<br>**Customer relationship.** Registration (ESA, no charge) or a subscription (Spire, amount not stated)<br>**The exit.** Switching cost high: porting and acceptance again on a new host |
| **7 Trust** | The host authenticates the upload through its own command chain and can stop the program (OPS-SAT's workflow). Keeping results from the host needs a key it cannot extract and isolation from its software; the pages read do not address either. |
| **8 How it fails, and how we would see it** | The program fails acceptance or exceeds its resources; the host stops it; passes delay the results; a host anomaly stops every guest at once.<br>**How we would see it:** The host's execution logs and telemetry, returned with the results (OPS-SAT's workflow); the guest's own checksums and timestamps on its outputs. |
| **9 Cost shape** | No charge after registration (ESA) or a subscription (Spire, amount not stated); others by quote, 2026-11-22. |
| **10 Evidence** | No charge, contract or paperwork; 50 percent of PRETTY's time operator claim ESA, OPS-SAT Space Lab page, 2026-11-22<br>Flat monthly fee, amount not stated operator claim Spire, Space Services page, 2026-11-22<br>About 40 processors on 10 satellites, 18 customers operator claim TechCrunch, 2026-04-13 (operator claim, reported by trade press), 2026-11-22 |

**Service only, no hop budget:** the program rides the host's links, whose budgets no page read gives.

*Status is the offer's: ESA's page lists PRETTY "Available now" (2026-11-22).*

### Offered: running a customer's code on an operator's fleet

Three kinds of offer were on public pages in November 2026. None of the commercial offers publishes a price on the pages read; ESA's laboratory is offered without charge.

- **Software on a fleet built for other work.** Spire's Space Services page, read on 22 November 2026, sells "Software in Space", an application of the customer's running on Spire's satellites, for a "flat monthly fee" whose amount is not stated. Loft Orbital's home page, read the same day, offers "On-Orbit AI" and counts 16 satellites launched and more than 25 missions deployed.
- **A cluster sold for applications in space.** Kepler's cluster, launched in January 2026 according to trade press, had about 40 Nvidia Orin processors on 10 operational satellites and 18 customers on 13 April 2026 (TechCrunch, trade press). The same report quotes Kepler as seeing itself not as a data center company "but as infrastructure for applications in space".
- **Nodes sold as orbital data centers.** Axiom Space states that its first two nodes launched on 11 January 2026 with Kepler's first satellites, reached by optical links rated at 2.5 (gigabytes per second in the page text, Gbps in a linked title), with software and model updates over the air; access is by enquiry.

Each offer is an operator claim, dated, and that is all the book can say about it. A buyer has more questions than these pages answer: the interface document, the resources a job may use, how jobs are scheduled against passes, and what the operator does when a job misbehaves. The fifth book of this series covers the buying. The node reached by an optical link has its own ledger, because its link and its trust field differ from the hosted program's.

**LINK LEDGER · LL-2026-19 · SERVICE LEDGER**

2026-11

System: An orbital data center node reached by an optical link (Axiom Space's first two nodes, launched on 2026-01-11 with Kepler's first satellites, as the documented example)

**Status on the book's date: not checked** **Class: service only**

| | |
|---|---|
| **1 Requirement** | Run a customer's workload on a computer in orbit that other satellites reach by laser, return the result, and update the node's software and models over the air. |
| **2 Ends and path** | The customer's systems on the ground; the ground stations of the relay network; the relay satellites; the computing node in orbit.<br>**Path:** Crosslinked: optical links between the node and the relay satellites; how traffic reaches the ground is not described on the operator's page. |
| **3 Orbit and time** | Low orbit, launched with Kepler's first satellites; the altitude is not on the page read.<br>**Time:** Access by enquiry; no turnaround, contact pattern or delay figure is published. |
| **4 Band and permission** | **Optical links.** Laser links; the page gives 2.5 gigabytes per second (text) and 2.5 Gbps (a title)<br>**Ground links.** The relay network's feeder links: band and license holder not on the page read |
| **5 Capacity** | **Up.** Not published for a customer.<br>**Down.** Not published for a customer.<br>2.5 is the operator's link rate, in a unit its page leaves unsettled; it is not a customer's throughput. |
| **6 Who owns each end** | **Space segment.** Axiom Space for the nodes; the relay satellites are Kepler's<br>**Ground segment.** Not stated on the page read<br>**Customer relationship.** Axiom Space, by enquiry<br>**The exit.** Not published |
| **7 Trust** | The operator states that software and model updates over the air are supported. How an upload is authenticated, who may send one, and whether one customer's code is kept from another's are not published. |
| **8 How it fails, and how we would see it** | A node, relay or ground link fails; an update fails or is malicious; a job overruns the node's energy or heat (handling not published).<br>**How we would see it:** Only through the operator: no public telemetry, health status or job record found in the pages read, as of 2026-11-23. |
| **9 Cost shape** | By enquiry only; no price published on the page checked 2026-11-22. |
| **10 Evidence** | Two nodes launched 2026-01-11 with Kepler's first satellites operator claim Axiom Space, Orbital Data Centers page, 2026-11-22<br>Optical links at 2.5 (GB/s in the text, Gbps in a title) operator claim Axiom Space, Orbital Data Centers page, 2026-11-24<br>Software and model updates over the air supported operator claim Axiom Space, Orbital Data Centers page, 2026-11-22 |

**Service only, no hop budget:** the node's optical and feeder link budgets are not published, and a stated link rate is not a hop budget.

*Status "not checked": the operator's page states the launch but no source speaking for November 2026 states the node's health or service.*

Two cautions follow from that page. A link rate is not a throughput, and here even the unit is unsettled: the operator's page gives 2.5 gigabytes per second in its text and 2.5 Gbps in a linked title, so the same number differs by a factor of eight with the unit. Either way it is a link rate, and a customer's share, after framing, coding, other customers and the time the links are available, is not published. Optical links between satellites also do not remove the last hop: a result still reaches a user through some link to the ground, and the first book of this series, *The Bit's Path*, works the budget of a hop in its chapter 4 and of optical links in its chapter 9.

## Launched prototypes

**A data center processor in orbit.** Starcloud launched its first satellite with an Nvidia H100 graphics processor in November 2025, as TechCrunch (trade press) reported on 30 March 2026. The same report says it was used to train an AI model in orbit and to run a Google language model, "according to Starcloud"; which model is not settled here, because the report's wording could not be checked against the company's own account. That is a company statement relayed by a journalist, not a published result: no paper or log of the work was found. The report also says that a second processor on board, an Nvidia A6000, failed during launch. It said Starcloud would launch a larger satellite, Starcloud 2, "later this year" with several processors including an Nvidia Blackwell chip; whether that happened by November 2026 was not checked.

**Google's Project Suncatcher: a prototype launched before its planned date.** Google announced Project Suncatcher on 4 November 2025 as a research design for computing in orbit with its own tensor processors, with "two prototype satellites by early 2027" in a partnership with Planet. Its test of a Trillium processor in a 67 MeV proton beam found "No hard failures were attributable to TID up to the maximum tested dose of 15 krad(Si) on a single chip", with irregularities in the high bandwidth memory from 2 krad(Si); a dose endpoint on one chip is not a qualification and does not give a single-event rate (chapter 3). Trade press reported one prototype launched on Transporter-18 on 1 October 2026, earlier than the planned date, which is not a conflict. On 19 November 2026 Google posted that it is partnering with Blue Origin on a Suncatcher launch "Targeted for Q1 2029" on New Glenn: an announcement, not a launch.

**A constellation sought and not read.** China's Three-Body Computing Constellation is cited in a 2026 research survey (a preprint) as pursuing a comparable architecture. NASA's 2026 small spacecraft report records its first group of 12 spacecraft as launched in summer 2025 (chapter 10); a 2026 preprint, not peer reviewed, gives May 2025. The builder's and the state news agency's pages could not be opened for this book, so its processing and its links between satellites are not printed here, and their absence is a limit of the sources read, not a finding about the system.

## Announced: plans in the announcer's words

Larger plans exist, and they are not capacity. A 2026 research paper by van Berkel (a preprint) reports that SpaceX applied to the FCC in February 2026 for a very large number of data center satellites; the filing itself was not read, so its numbers are not printed here. Google's 2029 Suncatcher launch is an announcement of 19 November 2026. Starcloud's second satellite was announced for 2026. Figure 12.1 lists announced plans with no column for capacity, on purpose: a capacity not in orbit is a forecast, and this book prints none.

The published studies of such plans, Turyshev's and van Berkel's preprints, are read in chapter 11: on their own assumptions, each, for the workloads it studies, finds computing in orbit for users on the ground competitive only under narrow conditions.

## What a reader can do today

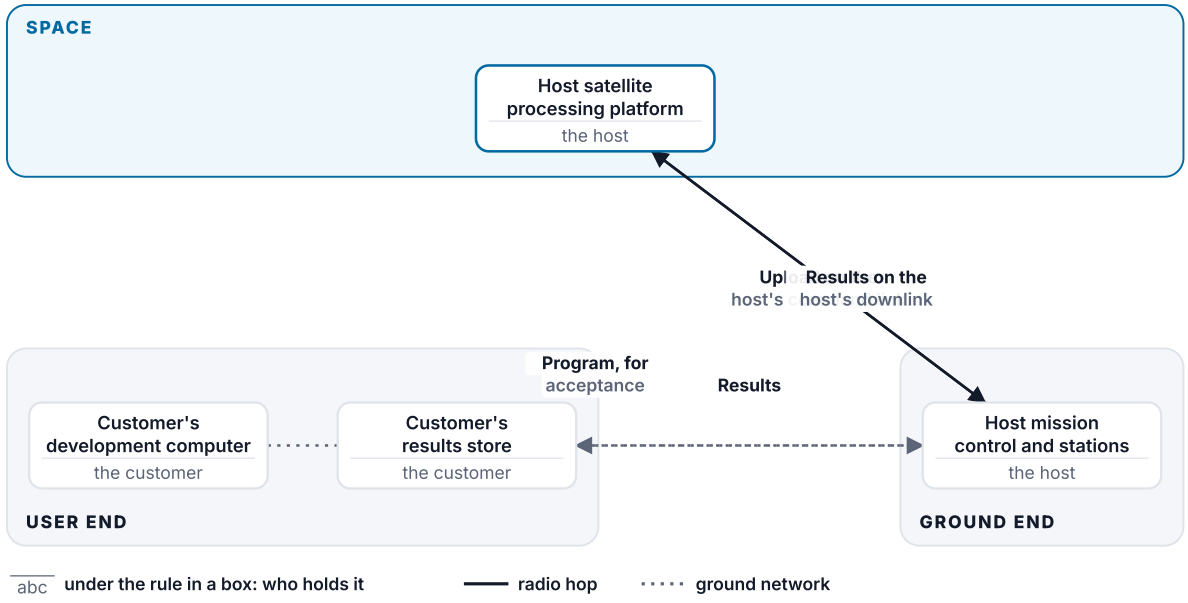
Three steps are open in November 2026, written as what each requires. None is a result.

- **Apply to the free laboratory.** Registration with ESA's OPS-SAT Space Lab requires an experiment that fits the platform (Linux, Python, an FPGA, a software-defined radio) and acceptance of ESA's process: in ESA's documentation for OPS-SAT, the Flight Control Team handles and uploads the experimenter's package; PRETTY's own procedure was not read. What it offers, in ESA's words, is the chance to test an experiment in space, subject to ESA's acceptance, the platform's resources and its operations; it does not offer control of the schedule.
- **Ask a hosting operator for its interface document.** The questions are the processor and memory available, the maximum job size and run time, the upload and result volume per day, how a job is accepted, how it is stopped, and what is logged. A written answer to these is worth more than a page of marketing.
- **Write the status fields before any meeting.** For each platform under consideration, fill the columns of Figure 12.1 from sources dated within the last few months, and leave a cell empty when no source speaks for it.

Figure 12.2 draws the hosted program's path as ESA describes it for OPS-SAT. Each arrow is a step the customer does not control.

FIGURE 12.2

## A hosted program, as ESA describes it for OPS-SAT, travels on links the customer does not hold




The path of a customer's program on ESA's hosted computer: the experimenter submits the package to ESA, which uploads it, runs it and returns the results on its own links; other hosts' workflows are not published in this detail.

*Author's diagram based on ESA's OPS-SAT Space Lab page and ESA's NanoSat MO Framework documentation for OPS-SAT, read on 2026-11-22 and 2026-11-23.*

### FAILURE: A PROCESSOR FAILURE REPORTED AT LAUNCH, AND A GAP THAT STAYS OPEN

TechCrunch (trade press, 30 March 2026) reported that one of the two data center class processors on Starcloud's first satellite, an Nvidia A6000, failed during launch; the H100 went on to the work the company describes. The report gives no cause, and none is inferred here. What the case shows is only what its source supports: a commercial processor that works on the ground can fail before the mission starts, and a platform's capacity in orbit is what survives launch, not what was integrated. The wider gap of this chapter is of the same kind. For the launched platforms in Figure 12.1, the record of demonstrated function found in the operator pages, research library and trade press read for this book up to 23 November 2026 is of four kinds: an agency's own statement (ESA for OPS-SAT), a company statement relayed by trade press (Starcloud), a research survey's listing (Spaceborne Computer-2), or nothing found (Axiom Space's nodes, the Suncatcher prototype). That is not a failure of the systems. It is a limit on what a buyer can know, and it is the reason the fields are kept apart.

**SECURITY NOTE**

**Who can interfere or command.** A shared computer in orbit has at least two kinds of user: the operator, who commands the satellite and updates the platform, and customers, who load jobs. The operator schedules work and can stop any job. Whether one customer's code can read another's memory, storage or results depends on the isolation the platform provides; none of the pages read describes it. Hosted code is itself a path of interference: a job that exhausts memory, power or the downlink share affects every other job. **How it is seen.** Ask how an upload is authenticated (for OPS-SAT, packages pass through ESA's Flight Control Team; Axiom states that over-the-air updates are supported, not how they are signed), whether results can be kept from the operator itself, which takes more than encryption on board (a key the operator cannot extract, isolation from the host's software, and a way to prove both), and what log the customer receives of its own job and of any access to it. Detection, diagnosis and attribution are separate questions: a job that returns wrong results may have met a particle, a bug or another customer. **How it is recovered.** A platform needs a known-good image that a bad update cannot overwrite, a command path that customers cannot use, and a stated way to restore a customer's previous job. Where the operator does not publish these, the buyer writes "not published" and asks.

**Monday rule:** for any orbital computing offer, write what was launched, what has been demonstrated, its health, who can get access since when, and who says so.

# Epilogue: Reading the Next Announcement




Sooner or later a reader of this book will meet an announcement: a company, an agency or a laboratory will say that it is putting computing in orbit, larger than before. The announcement will have a capacity figure, a date and a reason. This epilogue shows how to read it with the tools of the book, without a forecast of whether it will succeed.

## Five questions for any announcement

- **Which status fields does it fill?** Announced, launched, demonstrated, offered, qualified: write down each one the announcement claims and the date it gives. A capacity figure for something not in orbit belongs in the "announced" field, beside any ground test result with its conditions. If the announcement says "will", it is a plan.
- **Which kind of computing is it?** A platform computer, communication processing, payload data processing or hosted computing for customers. Each has its own budgets and its own buyers, and an announcement that moves between them in one paragraph has not yet said what it is.
- **What are the four budgets?** The power drawn and where it comes from in sunlight and in eclipse; the heat to be radiated and the radiator that does it; the orbit, the mission length and the parts, with their test conditions; and what crosses the link, in both directions. A figure without its unit or its conditions goes on the sheet as published without conditions, and decides nothing.
- **For whom, against which objective?** The verdict form of chapter 1. "Computing in orbit is cheaper" or "bandwidth is the scarce resource" is not a verdict until it names a workload, a reader and an objective. A workload whose input starts on the ground and whose result returns to the ground crosses the link twice; one whose input is already in orbit may not.
- **Who says so, and what kind of proof is it?** A maker's statement, a ground test, flight data or a certification. Trade press repeating a press release is still the press release.

None of the five needs inside knowledge. Each needs only the discipline to write the answer down, and to leave a field empty when the announcement does not fill it.

**THE AUTHOR'S VIEW, AS OF NOVEMBER 2026**

Four constraints look likely to decide the larger plans, and they are offered as the author's reading of the chapters, not as a forecast. Heat rejection at scale: radiator area grows with the power to be rejected, and the published studies cited in chapter 11 find it among the costs that a node must carry and a data center on the ground does not. Radiation in very small transistors: the record for the newest processes in orbit is short, and a ground test of dose says little about years of single events. The uplink: a node that sells computing to customers on the ground must receive their code, their models and often their data. And repair: a computer that nobody can reach is not fixed by hand; unless a spare on board takes over, replacing its hardware takes a launch. Each of these is a budget line on the sheet, and each can be filled with a number once a source publishes one.

## Where to go next

Renting computing, payloads and ground time in orbit, with quotes, contracts and a dated price register, is the subject of the fifth book of this series, *Rent Before You Build*. How flight software is structured, how a watchdog and a safe mode are written, and what software assurance asks for are covered in depth later in this series.

The Apollo computer that opened this book was sized before its workload was written down, and its programs were woven into rope months before it flew. The newest, by their operators' account, accept new programs by radio. Between them, the question did not change: what must be computed, where, and which budget pays for it.

APPENDICES

APPENDIX A

# The Trade Sheet and the Link Ledger: Blank Pages and Index

The blank forms this book uses, and an index of every worked example in it. The trade sheet is this book's own page, for the question "should this computation happen in orbit?". The Link Ledger is the form of the whole series, used here only where computing changes a link; its data volume page is this book's variant of page 2. Copy the blanks, fill one sheet per workload, and write every number with one of the seven classes of the Reader's Brief, its source and its date.

## The trade sheet, blank

One sheet per workload and per place: fill it once for the computation on board and once for the same computation on the ground, then compare. Chapter 1 teaches it line by line; chapter 11 fills four.

**The trade sheet: workload _____ · place: on board / on the ground · date _____**

| Line | What to write | Value and range | Class, source, date |
|---|---|---|---|
| 1 Objective and payer | Mission return, delay, lifecycle cost or commercial margin; for whom | | |
| 2 Workload and result quality | What is computed, on what input, how often; who needs the result by when; accuracy of the on-board result against the ground result | | |
| 3 Data | Produced per orbit; fraction worth sending; size of the result | | |
| 4 Link | Contact minutes per day, rate, cost shape; delay; radio energy saved; storage; retransmission; added uplink (software, models) | | |
| 5 Energy | Sunlit and eclipse power apart; duty cycle; dynamic, leakage, idle and support power; array degradation over life; share of the power budget | | |

| Line | What to write | Value and range | Class, source, date |
|---|---|---|---|
| 6 Heat | Emitted less absorbed (solar, albedo, Earth infrared) with view factors; conduction path to the radiator; allowed temperatures; labeled sunlit steady state; eclipse transients flagged for a thermal model | | |
| 7 Radiation | From an environment model run, device test data, sensitive bits, mode and mitigation; raw events, corrected errors, outages and hazardous failures kept apart; or "high, medium, low" with reasons | | |
| 8 Mass and volume | Added to the satellite; launch cost only as a dated booking figure with what it leaves out | | |
| 9 Updates and trust | Who can change the software or the model; how an upload is authenticated; how a bad update is rolled back | | |
| 10 Lifecycle cost | Engineering, qualification, operations, ground infrastructure, replenishment; or "not published" | | |
| 11 The ground alternative | The same job on the ground: what it needs from the link and what it costs; when it is infeasible | | |
| 12 Uncertainty | The range on each line and its class | | |
| 13 Sensitivity | The one or two lines that change the verdict | | |
| 14 Assumptions and limits | What the sheet does not model; when a system model is needed | | |
| 15 Verdict | "Advantageous for ____, against ____, if ____", or "not advantageous", or "cannot be decided from public sources" | | |

The verdict is never a single word. It names who gains, against which objective, under which assumption, and the line in 13 that would change it. A sheet whose inputs are assumed or solved is an illustrative model and says so in its title; it shows how the arithmetic goes, not that a real service works.

## The part choice matrix, blank

Chapter 6 fills one row per way of buying a function. Fill every cell from a document and date it, or write "not established in the sources read"; an empty cell is not a weakness of the part.

**Part choice: function _____ · orbit _____ · mission length _____ · criticality _____**

| Approach | Part and lot | Dose, and who says so | Single events, with conditions | Package and flow | Kind of support | What the design adds | Who accepts the residual risk |
|---|---|---|---|---|---|---|---|
| Hardened | | | | | | | |
| Tolerant | | | | | | | |
| Commercial | | | | | | | |

## The data volume page, blank

Where computing changes a link, this book uses page 2 of the ledger in a data volume form instead of a hop budget in decibels; the hop budgets are in the first book of this series, *The Bit's Path*. Each line is in bits, per orbit or per day, with its class and source. When the volume to send per day exceeds the volume the contacts can carry, the shortfall adds to the backlog in storage; otherwise the backlog can drain, as far as contact timing allows.

**Data volume: link _____ · operating point (orbit, contacts, rate, date) _____**

| Line | How it is found | Value | Class, source, date |
|---|---|---|---|
| A Produced per orbit | Sensor rate times active time per orbit | | |
| B Orbits per day | From the orbit | | |
| C Produced per day | A times B | | |
| D1 Fraction kept after filtering | Share of the produced bits that the on-board filter keeps (bits, not a count of observations); the rest are discarded | | |
| D2 Compression ratio | Bits after compression over bits before, lossless or lossy, said which; for a result that replaces the data, the result's bits over the data's bits (the inverse of a compression factor such as 2.5 to 1: a D2 of 0.4) | | |

| Line | How it is found | Value | Class, source, date |
|---|---|---|---|
| E To send per day | C times D1 times D2 | | |
| F Contacts per day, and usable minutes per contact | From the ground stations and the orbit | | |
| G Downlink information rate | From the link's own budget, at its operating point | | |
| H Deliverable per day (capacity) | Usable contact seconds per day (from F) times G, less any protocol overhead and retransmission not already taken out of G | | |
| I Backlog change per day | E minus H while a backlog remains (actual traffic sent cannot exceed what is stored); a positive value fills storage | | |
| J Storage on board | Capacity, and the days of backlog it holds | | |
| K Added uplink | Software and model updates, per update and per year | | |
| L Discarded and never seen | C times (1 minus D1): observations filtered out, not compressed ones; and how a sample of them reaches the ground for checking | | |

# The blank Link Ledger

**LINK LEDGER · BLANK · SERVICE LEDGER**

Link id (LL-year-nn) Date System

Status: Planned | In service | Suspended | Retired | Failed | Not checked

Class: full | service only | budget only

| | |
|---|---|
| **1 Requirement**<br>What must be delivered: size, deadline, coverage, availability. | |
| **2 Ends and path**<br>Who talks to whom; the path as it really is (transparent, regenerative, crosslinked, store and forward). | |
| **3 Orbit and time**<br>Altitude, pass or continuous; delay one way and round trip; scheduling and queueing; Doppler. | |
| **4 Band and permission**<br>Frequencies per hop; who holds the license, under which rule, in which country. | |
| **5 Capacity**<br>Up and down separately; peak against sustained; contention. | |
| **6 Who owns each end**<br>Space segment, ground segment, customer relationship, the exit. | |
| **7 Trust**<br>Who can command it; whose keys; how it is updated and recovered; jamming and spoofing. | |
| **8 How it fails, and how we would see it**<br>Silent failure named; the evidence that shows it. | |
| **9 Cost shape**<br>Per minute, per megabyte, per month or per terminal; per kilogram for launch; a price only with its date and what it leaves out. | |
| **10 Evidence**<br>Each number with its evidence class, source and date; an assumption is marked so. | |

### The six questions

1. What must be delivered, and by when?
2. How is the work divided between user, space and ground?
3. Does each hop close, with what margin, under which assumptions?
4. Who owns each end, and who gave permission?
5. How does it fail, who can interfere, and what would show us?
6. What can I use or rent before I build?

### The evidence classes

**measured** From an instrument or a log; who and when.
**calculated** From stated inputs by a formula the text gives.
**simulated** From a named model and its settings.
**operator claim** Stated by the operator; not independently checked.
**standard** From a named standard or rule, clause or table.
***assumption*** To verify: no primary source read; name what settles it.
**published history** Published history: an agency history or labeled scholarship.

## LINK LEDGER · BLANK · HOP BUDGET

**Hop name: _____ (one block per radio hop; a variant of it is a case: Case A, Case B)**

From ____ to ____ · frequency · distance · reference plane · information rate · symbol rate · stated error rate

| Part | Line | Value, in its own unit | dB, signed | Evidence | Source and date |
|---|---|---|---|---|---|
| TRANSMIT POWER AND ANTENNA GAIN | Transmitter output (dBW) | | | | |
| | Feeder loss | | | | |
| | Transmit antenna gain (dBi) | | | | |
| LOSSES | Free space path loss | | | | |
| | Atmosphere and rain | | | | |
| | Polarization and pointing | | | | |
| RECEIVE SENSITIVITY | Receive G/T: antenna gain less system noise temperature (dB/K) | | | | |
| | Boltzmann's constant: add 228.6 dB (it is -228.6 dBW/K/Hz) | | | | |
| BANDWIDTH | Information rate (bit/s): for Eb/N0, subtract 10 log10 of it | | | | |
| | Symbol rate (baud): for Es/N0, subtract 10 log10 of it instead | | | | |
| INTERFERENCE | Interference, as the step from C/N0 to C/(N0+I0) | | | | |
| CODING | Modulation, code and implementation loss | | | | |
| REQUIRED SIGNAL QUALITY | Required Eb/N0, Es/N0 or C/N0 (choose one; the same quantity as above) for the stated error rate | | | | |
| RESULT | **Achieved Eb/N0, Es/N0 or C/N0: the sum of every line above the required one** | | | calculated | |
| | **Required: the Eb/N0, Es/N0 or C/N0 for the stated error rate** | | | source: | |
| | **Margin: achieved less required** | | | design minimum: ___ dB | |

Verdict (circle one): CLOSES · DOES NOT CLOSE, with a margin of _____ dB

**Operating point** (rate, direction, elevation, weather, availability):

**Source of the required value** (a standard or a published curve):

**If it does not close, the line that fails, and by how many dB:**

**Transparent circuit: the combined line.** Combine C/N0 (or C/(N0+I0)) of each hop, before the rate and the implementation loss. Uplink C/N0: _____ dBHz. Downlink C/N0: _____ dBHz. Combined (the inverse ratios add): _____ dBHz. Less the rate and the implementation loss, once: _____ dB. Required: _____ dB. Margin: _____ dB. Two passing hops are not a passing circuit.

**How to add it up.** Write every line in decibels with its sign: gains positive, losses negative. Beside each number you may write the level after that line (dBW after the transmitter, dBW/K after G/T, dBHz after Boltzmann's constant, dB after the rate). After Boltzmann's constant the level is C/N0 in dBHz. Es/N0 = C/N0 minus 10 log10(symbol rate in baud); Eb/N0 = C/N0 minus 10 log10(information rate in bit/s). Compare like with like: the required value is the same quantity as the achieved one. Write a dB value on one rate line only, and on neither for C/N0. The achieved value is the sum of every line above the required one. The margin is the achieved value less the required value. The hop closes when the margin is at or above the design minimum you wrote down before you started.

## Index of the worked examples

The worked sheets and tables of this book, by chapter. Every sheet built on solved or assumed inputs is an illustrative model and is labeled so where it is printed.

| Chapter | Worked example | Kind |
|---|---|---|
| 1 | The trade sheet taught line by line on a neutral example | Trade sheet, illustrative |
| 3 | Line 7 of the trade sheet for three orbits, from published runs (table 3.1) | Worked table |
| 4 | Line 7 reworked: where each kind of event goes, before and after mitigation (table 4.1) | Worked table |
| 5 | Flight processors side by side with identical test-condition columns (figure 5.2) | Worked table |
| 6 | One function bought four ways: the decision matrix | Worked table |
| 7 | Space FPGA families with identical fields (figure 7.2) | Worked table |
| 9 | One day of data volume: send everything, compress, or screen and compress (figure 9.1) | Data volume table, illustrative |
| 10 | Cloud screening on a low-power accelerator; detection with a stated decision deadline; energy per thousand inferences (figure 10.2) | Two trade sheets; a calculated table |
| 11 | Cloud screening, detection with a deadline, regeneration in a communication payload, batch computing for customers on the ground; the energy and heat balances (figures 11.1 and 11.2) | Four trade sheets; two illustrative balances |
| 12 | Computing platforms in orbit by status fields (figure 12.1) | Worked table |

Each ledger id names one link across the whole series, so a link another book also treats keeps that book's id and numbers, and this book prints only the page it needs. A service ledger has no budget. The PDF page column gives the pages where each worked ledger is printed.

## The worked Link Ledgers in this book

| Id | Date | System | Class | Status | Budget | Requirement | PDF page |
|---|---|---|---|---|---|---|---|
| **LL-1969-01** | 1969-07-20 | Apollo: position and burn solutions computed on the ground, sent up to the guidance computer (Apollo 11) | service only | Retired | service only | Keep the computer's knowledge of position and of the next burn good enough to steer, while the ground does most of the navigation: ground computers were the prime source of position, except during maneuvers. | 32 |
| **LL-1995-01** | 1995 to 1997 (prime tour) | Galileo: science return through the low-gain antenna after on-board compression | service only | Retired | service only | Return the orbiter's science, imaging included, from the tour that began on 7 December 1995, without the high-gain antenna. JPL's aim (1995): at least 70 percent of the original objectives. | 114 |
| **LL-2020-01** | 2020 | Phi-Sat-1 on FSSCat: an Earth observation CubeSat that screens clouds on board | service only | Not checked | service only | Show in flight that a neural network on a commercial accelerator finds cloud well enough to discard, before the downlink, images more than 70 percent cloudy, with clear images wrongly marked cloudy under 1 percent of the test set. | 116 |
| **LL-2026-18** | 2026-11 | A customer's program on a hosted computer in low orbit: upload, run and results through the host (ESA's OPS-SAT Space Lab as the documented example) | service only | In service | service only | Run a customer's program on a computer in orbit and get its results back, without owning, licensing or commanding a satellite. | 157 |

| Id | Date | System | Class | Status | Budget | Requirement | PDF page |
|---|---|---|---|---|---|---|---|
| **LL-2026-19** | 2026-11 | An orbital data center node reached by an optical link (Axiom Space's first two nodes, launched on 2026-01-11 with Kepler's first satellites, as the documented example) | service only | Not checked | service only | Run a customer's workload on a computer in orbit that other satellites reach by laser, return the result, and update the node's software and models over the air. | 159 |

APPENDIX B

# Timeline of Flight Computers by National Line, 1962 to 2026, with Cases Sought

Every entry below was read in the source named in its last column, and each date is given as precisely as that source gives it. A launch vehicle or a launch vehicle module is kept apart from a satellite. A maker's statement stays a maker's statement. Facts this book sought and could not read are listed in the second part, with the source that would settle them and the date they were sought, rather than guessed. Status on the book's date, where a part or system is still offered, is in Appendix D.

**Source key.** CiS = NASA, *Computers in Spaceflight: The NASA Experience* (1988), with the chapter. ALSJ = NASA's Apollo Lunar Surface Journal. NASA 2021 = NASA History, "40 Years Ago: The Countdown Begins for STS-1; First Launch Attempt Scrubbed". GSFC 2015 = D. McComas, NASA Goddard, "Lessons from 30 Years of Flight Software". JSASS = Transactions of the Japan Society for Aeronautical and Space Sciences. TP = trade press, labeled as such. Full titles are in Appendix E.

### United States

| Date | Computer or event | Architecture as the source gives it | Where it flew | Source |
|---|---|---|---|---|
| 19 Apr 1962 | Gemini computer contract to IBM, 26.6 million dollars | Discrete components; 4,096 addresses of 39 bits; 58.98 lb | Gemini; last of 20 machines delivered by Dec 1965 | CiS ch. 1 |
| Gemini VIII (not dated in the source) | Auxiliary Tape Memory, first flown on Gemini VIII | 1,170,000 bits; each program recorded three times and voted | Gemini | CiS ch. 1 |
| 1969 | Programmable Central Computer and Sequencer, JPL and Motorola | 128 words alterable in flight; 26 lb | Mariner Mars 1969 | CiS ch. 5 |

| Date | Computer or event | Architecture as the source gives it | Where it flew | Source |
|---|---|---|---|---|
| 20 Jul 1969 | Apollo 11 landing: 1202 and 1201 alarms, restarts, landing continued | Apollo Guidance Computer: MIT design, Raytheon build; 16 bit word; 36K fixed, 2K erasable; 70.1 lb, 70 W | Command and lunar modules | CiS ch. 2; ALSJ |
| 1975 | Viking orbiter and lander computers | Orbiter: 18 bit, 4K plated wire, dual redundant. Lander: two Honeywell HDC 402, 18K each | Viking 1 and 2 | CiS ch. 5 |
| 1977 | Voyager's three computer types | Command computer as Viking's; separate attitude computer; Flight Data System in CMOS | Voyager 1 and 2 | CiS ch. 6 |
| c. 1977 | RCA 1802 chosen for Galileo | Six 1802s with 176K (unit not given) in the Command and Data Subsystem; 19 microprocessors on the spacecraft | Galileo (launch slipped beyond 1986) | CiS ch. 6 |
| 10 Apr 1981 | STS-1 first launch attempt scrubbed: backup computer could not communicate with two primaries; 40 ms timing problem | Five IBM AP-101 computers; four in a redundant set, one backup with separately developed software | Space Shuttle | CiS ch. 4; NASA 2021 |
| 1982 | Cyclotron tests: Galileo's 2901 bit-slice chips upset-sensitive; Sandia hardened 2901s chosen | 2900 series bit-slice attitude computer | Galileo | CiS ch. 6 |
| 24 Apr 1990 | Hubble launched | NSSC-I and a DF-224 named, 32K 24 bit words at a time; the split of work not stated | Hubble Space Telescope | GSFC 2015 |
| 27 Feb 2014 | GPM launched on an H-IIA | RAD750 at 132 MHz, 36 MB RAM, 4 GB recorder | GPM | GSFC 2015 |

| Date | Computer or event | Architecture as the source gives it | Where it flew | Source |
|---|---|---|---|---|
| Listed 2015 | Goddard missions' processors | 80386 at 6 MHz (SAMPEX); Mongoose V at 12 MHz (MAP); ColdFire (MMS); RAD750 at 115 to 132 MHz (SDO, GPM, Webb instrument module) | As named | GSFC 2015 |
| 15 Aug 2022 | HPSC contract to Microchip, 50 million dollars | RISC-V; target at least 100 times current spaceflight computers | No flight in the sources read | NASA JPL |
| Feb 2026 | First email from an HPSC processor; in testing, not space qualified, as of Mar 2026 | As above | Ground | NASA HPSC page |
| 27 Oct 2026 | Early radiation results for PIC64-HPSC1000-RH: 200 krad(Si), latch-up tested to 78.2 MeV cm2/mg | As above | Ground tests | Microchip release (TP) |
| Read Nov 2026 | RAD750, in BAE's words, powers more than 100 satellites, including the GPS III navigation payload | Radiation-hardened PowerPC 750 based | As stated by the maker | BAE page |

### Europe, India and Japan

| Date | Computer or event | Architecture as the source gives it | Where it flew | Source |
|---|---|---|---|---|
| Reported 2008 | ERC32 platform computer | SPARC V7 with on-chip peripherals | Proba (ESA), all platform functions | Surrey thesis 2008; RTEMS |
| Listed 2015 | GR712RC | Dual-core LEON3FT, 100 MHz | Not named | GSFC 2015 |
| Read Nov 2026 | GR740, ESA's Next Generation Microprocessor | Quad-core LEON4FT SPARC V8, 250 MHz; 300 krad(Si); latch-up immune to 125 MeV cm2/mg; QML Class V and Q | Three on NASA's Roman telescope, per the RTEMS project | Maker page; RTEMS |
| Since 2009 | VIKRAM1601 in ISRO launch vehicle avionics; made at SCL from 2016 | 16 bit, custom instruction set, Ada | Launch vehicles (not satellites) | ISRO |
| 5 Mar 2025 | VIKRAM3201 and KALPANA3201 handed over to ISRO | VIKRAM3201: 32 bit, 180 nm CMOS. KALPANA3201: 32 bit SPARC V8 | VIKRAM3201 validated on POEM-4, PSLV-C60, a launch vehicle module | ISRO |
| Reported 2021 | SuperH computers in Japanese microsatellites | SH-3 on REIMEI; SOI-SoC SH-4 on Hodoyoshi-3 and 4 | Those satellites; later PROCYON | JSASS 2021; JSASS 2014 |

## Cases sought and not sourced

Each case below was sought on the date shown in a source of the kind named. None could be read on that date, and no primary record of them was among the sources available. Nothing in the chapters depends on them, and no figure for them is printed. A case listed here is a gap in what was read, not a judgment on its importance.

### Sought, not read

| Line | Case | Source that would settle it | Sought |
|---|---|---|---|
| Soviet Union | The on-board computers of the Zond lunar spacecraft, and their family | The builder's or an institute's history | 2026-11-23 |
| Soviet Union | The computers of the Salyut and Mir stations | Builder or Roscosmos records | 2026-11-23 |
| Soviet Union | Biser-4, the computer of the Buran orbiter | Builder or institute history | 2026-11-23 |
| Russia | A current Russian flight processor | A maker's datasheet or an agency page | 2026-11-23 |
| Europe | The computers of Spacelab and of the first ESA science missions | ESA history pages or reports | 2026-11-23 |
| Europe | ERC32 and LEON dates and missions from ESA itself | ESA microelectronics pages | 2026-11-23 |
| Europe | Editions and dates of the SpaceWire and SpaceFibre standards | ECSS-E-ST-50-12C and ECSS-E-ST-50-11C | 2026-11-23 |
| Japan | Early Japanese on-board computers; SDS-1 (2009); the HR5000 family | JAXA pages and papers | 2026-11-23 |
| China | One flight processor with its architecture and flight record (for example the BM3883 a 2024 survey names) | The builder's or an institute's page; a peer-reviewed paper with flight data | 2026-11-23 |
| United States | RAD6000: clock, missions, radiation figures | A BAE or NASA page | 2026-11-23 |
| United States | RAD5545 processor dose and latch-up figures; RAD510 | BAE datasheets | 2026-11-23 |
| United States | Hubble's DF-224 replacement; the Gemini, Viking and Shuttle computers' later histories | NASA Hubble pages; CiS chapters not read | 2026-11-23 |
| United States | The mechanism of the STS-1 scrub in its designer's words | J. R. Garman, ACM SIGSOFT Software Engineering Notes, October 1981 | 2026-11-23 |

| Line | Case | Source that would settle it | Sought |
|---|---|---|---|
| United States | The Apollo uplink's frequencies, power and rates | Apollo network technical reports; the Apollo 11 Mission Report | 2026-11-23 |
| United States | The Roman telescope's GR740 count from NASA; Perseverance's and Ingenuity's computers from NASA's mission pages | NASA mission pages | 2026-11-23 |
| Radiation | A dose-depth model run for the three orbits of chapter 3, with protons and electrons separated | A SPENVIS or AE9/AP9 run with its inputs | 2026-11-24 |
| Radiation | The loss of the Phobos-Grunt probe (2011) as its commission described it; the loss of Phobos 1 (1988) | The commission's or the agency's own record | 2026-11-24 |
| Radiation | Upsets logged against the South Atlantic Anomaly by a university microsatellite | A peer-reviewed paper with the flight data | 2026-11-24 |
| Standards | ECSS-Q-ST-60-15C (radiation hardness assurance) and ECSS-Q-ST-60-13C (commercial components): the text, not only the title | ECSS publications | 2026-11-23 |
| Standards | MIL-PRF-38535 (the QML classes) and MIL-STD-883 (test methods): the definitions in the documents themselves | Defense Logistics Agency, Land and Maritime | 2026-11-23 |
| Standards | ESCC 9030 and ESCC 25100, and the qualified parts list entry for NG-ULTRA | ESCIES, the European Space Components Information Exchange System | 2026-11-23 |
| Agency | NASA's payload risk classes and the parts requirements tied to them | NPR 8705.4, NASA's directives system | 2026-11-23 |
| Agency | The QML listing of the RT PolarFire RTPF500ZT | The Defense Logistics Agency's Qualified Manufacturers List | 2026-11-23 |
| Parts | The antifuse FPGA reliability investigation of 2003 to 2005 | NASA NEPP and Office of Logic Design records | 2026-11-23 |

| Line | Case | Source that would settle it | Sought |
|---|---|---|---|
| Parts | RTG4 data; the RT PolarFire configuration cell type; XQRVE2302 shipment and Versal Class Y after March 2025 | Microchip and AMD datasheets and product pages | 2026-11-23 |
| Parts | Lead times and prices for any part or computer in this book | Dated maker or distributor pages | 2026-11-23 |
| Parts | Flight computers of other countries with published flight data, for the matrix of chapter 6 | Makers' and agencies' pages | 2026-11-23 |

Regulator pages. The one regulator record this book needed is SpaceX's application to the US Federal Communications Commission for data center satellites, reported for February 2026 (chapter 12). The FCC's filing system refused the reading on 23 November 2026, so the application is cited only as a preprint reports it, and Appendix D lists it as sought. No other regulator's page was needed by the chapters: the pages of the regulators of the United Kingdom, Australia, New Zealand, Canada and India, which other books of this series list as sought and not read, were not sought for this one. Status facts sought and not read, such as the health of a platform on the book's date, are listed in Appendix D.

APPENDIX C

# Glossary

Terms as this book uses them, with the chapter that explains each. Where a standard or a maker defines a label, its own text governs.

### Radiation and its effects

| Term | Meaning in this book | Ch. |
|---|---|---|
| Cross-section | Events observed divided by the fluence (particles per square centimeter that struck the part in the test), in cm2 per device or per bit; for heavy ions, plotted against LET. With the environment it gives a rate. | 3 |
| Cumulative effect | Damage that builds up over the mission: total ionizing dose and displacement damage. | 3 |
| Displacement damage; displacement damage dose (DDD) | Defects left when particles knock atoms out of the crystal lattice, and the dose that measures them. Cumulative; matters most for detectors, light emitters and gain. | 3 |
| Dose-depth curve | Mission dose against shielding thickness (usually mm of aluminum), computed from named models for a stated orbit, duration and confidence. | 3 |
| Dose rate; ELDRS | How fast a dose arrives. Enhanced low dose rate sensitivity (ELDRS): some bipolar parts degrade more at a given dose when it arrives slowly, as in orbit, than in a fast ground test. | 3 |
| krad(Si) | Thousands of rad of dose absorbed in silicon, the usual unit of a dose rating. | 3 |

| Term | Meaning in this book | Ch. |
|---|---|---|
| Linear energy transfer (LET) | Energy a particle deposits per unit length of its track; space-electronics tests quote it divided by the material's density, in MeV cm2/mg, for silicon. The yardstick of single-event tests. | 3 |
| Multiple-bit upset (MBU) | Several bits of one protected word changed by one particle. Can defeat a code that corrects one bit per word. Several cells hit by one particle, wherever they fall, is a multiple-cell upset (MCU). | 3, 4 |
| Radiation hardness assurance | The chain from the environment specification through transport to the part, part data, margin and rates, to mitigation and an acceptance decision. | 3 |
| Single event effect (SEE) | A disturbance caused by one particle passing through or near a sensitive node. Destructive forms include latch-up, burnout (SEB) and gate rupture (SEGR). | 3 |
| Single event functional interrupt (SEFI) | A device stops working properly until reset, reconfigured or power cycled. Not destructive, but an outage. | 3, 4 |
| Single event latch-up (SEL) | A parasitic structure switched on by one particle, drawing high current; potentially destructive unless power is removed in time. "Immune to" a LET means none was seen up to that LET in the tests run. | 3, 4 |

| Term | Meaning in this book | Ch. |
|---|---|---|
| Single event transient (SET) | A short glitch in a circuit, logic or analog; in logic it becomes an upset if a storage element captures it. | 3 |
| Single event upset (SEU) | A stored state changed by one particle; when one particle changes several bits, a multiple-bit upset. Not destructive. | 3 |
| South Atlantic Anomaly (SAA) | The region over the South Atlantic where trapped particles reach unusually low altitudes; low-orbit satellites crossing it can see enhanced flux and upset rates. | 3 |
| Total ionizing dose (TID) | Energy deposited per unit mass by ionizing radiation; it builds up and shifts transistor behavior until a part fails its limits. | 3 |

### Mitigation

| Term | Meaning in this book | Ch. |
|---|---|---|
| Cold sparing | A second unit kept off until the first fails. Restores service after a failure, with downtime for detection and switch-over. | 4 |
| Dual redundancy, lockstep | Two copies compared. Detects a fault but cannot tell which copy is right. In lockstep the copies run in exact step. | 4 |
| Error detecting and correcting code (EDAC, ECC) | Extra bits stored with each word so that changed bits within the code's strength are found and corrected (more can be missed or miscorrected); repairing the stored word needs a write back or a scrub. | 4 |
| Fault protection; safe mode | On-board software that detects faults and puts the spacecraft into a safe configuration until the ground acts. How it is implemented belongs to a later book of this series, on flight software. | 2, 4 |
| Golden image | A protected copy of software or an FPGA configuration, used to restore a known-good state. | 4, 7 |
| Interleaving | Spreading the bits of each word apart in memory, to reduce the chance that one particle changes two bits of the same word; how much it reduces depends on the memory's layout. | 4 |
| Latch-up protection | A circuit that detects the current rise of a latch-up and cuts power. Works only if it acts before damage. | 4 |

| Term | Meaning in this book | Ch. |
|---|---|---|
| Scrubbing | Reading memory, or an FPGA's configuration, on a schedule and writing back corrected or known-good contents, so that errors are removed before they accumulate. Fixing a configuration bit does not fix the state it already corrupted. | 4, 7 |
| Supervisor | A simpler, more robust device that watches a processor and can reset, reload or power cycle it. | 4, 5 |
| Triple modular redundancy (TMR) | Three copies with voting. In the terms of the presentation chapter 4 follows (others draw the lines differently): block TMR votes only at outputs; local TMR copies flip-flops; distributed TMR copies data paths; global TMR also copies clocks. In the classical model (independent copies, constant failure rate, perfect voter), without repair it fails sooner on average than one copy. | 4 |
| Watchdog timer | A timer that resets a device unless software clears it in time. | 2, 4 |

### Parts, qualification and claims

| Term | Meaning in this book | Ch. |
|---|---|---|
| Commercial off the shelf (COTS) | A standard commercial part; in NASA Goddard's radiation classification, "designed with no attempt to mitigate radiation effects". | 3, 6 |
| ESCC, ESCC 9030 | The European Space Components Coordination system and its specifications; 9030 is described by NanoXplore and STMicroelectronics as a standard for high-performance microcircuits on organic substrates or in plastic packages. | 6 |
| Failure criterion | The line a tester draws between "still working" and "failed". A test result cannot be compared without it. | 3, 6 |
| Flight proven | A maker's claim of flight heritage. Ask which part and version, which mission, how long and in which mode. | 6 |
| Hardened by design, hardened by process | Radiation resistance built into the circuit design, or into the manufacturing process. | 5, 6 |
| MIL-PRF-38535; QML classes Q, V and Y | The US performance specification behind the Qualified Manufacturers List kept by the Defense Logistics Agency, and the classes makers quote. The specification's text defines the classes. | 6 |

| Term | Meaning in this book | Ch. |
|---|---|---|
| MIL-STD-883; Class B | The US test method standard for microcircuits, source of screening and qualification tests. Ask which document defines a maker's "Class B" flow. | 6 |
| Radiation tolerant | A part designed to account for and mitigate radiation effects by process or design, with stated figures. | 3, 6 |
| Screening, qualification | Screening is applied to every delivered unit; qualification groups are run on samples. Neither is a mission's radiation analysis. | 6 |

### Status fields

| Field | What it records | Ch. |
|---|---|---|
| Announced | A plan stated by its owner, with the date and the owner's words. Never a capacity in orbit. | 12 |
| Launched | Launched, with the date and the vehicle; reaching the intended orbit is recorded with it, from a source. | 12 |
| Demonstrated function | What a source says the system did in orbit, and no more. | 12 |
| Current health | Whether it works now, by whose statement and of what date. | 12 |
| Service access | Who can buy or apply, in which country, since when. | 12 |
| Qualification | Standard, level, issuer and date. | 6, 12 |
| Retired, failed or deorbited | The end of the system, with the date and the source. | 12 |
| "No public evidence found" | Written with the search scope and cutoff date. An empty cell means not established in the sources read. | 6, 12 |

APPENDIX D

# Dated Register: Parts, Qualification and Computing in Orbit as of November 2026

The facts in this register change faster than the book: which qualification a part has reached, what a maker says it has shipped, which computers have been launched and what they have been shown to do, and what has only been announced. The chapters use these facts for their analysis; this register keeps them in one place, each with the date it describes, the source and its own date, and the date the source was read. Before relying on any row, read its source again. One row is a price, the rideshare booking minimum that chapter 11 uses, printed with its date and what it leaves out; no other current price of a service or part is used in the book (contract values and the studies' cost assumptions are printed as such).

## How to read a row

- **Event** is the date the fact describes: a qualification completed, a launch, an announcement, a test. **Source** names the document and, where it has one, its own date of publication, which is often later than the event. **Read** is the date the source was read for this book. The three are kept apart because a page published in March can describe a status of the previous year, and a page read in November can still give a status of March.
- A status row (shipping, qualified, in service, available) needs a source that speaks for its own date. A row is refreshed by reading the source again. Re-reading changes the read date, not the event date: a page that still says what it said in March still supports a status of March, not of November. If the source reports a new state, the row is rewritten with a new event date; if it can no longer be read, the row is struck. An old announcement never upgrades a status.
- **Kind** is one of the seven evidence classes of the Reader's Brief. A label in brackets says what kind of document carried the fact: trade press, a peer-reviewed paper, a preprint, a republished release. An operator claim is what a maker, operator or agency says about its own product or mission, including what trade press reports it said. "Qualified" in a maker's release is the maker's statement of a qualification, not this book's check of it; chapter 6 explains what each label certifies.
- **Ch** is where the book uses the fact. R is the Reader's Brief, EP the Epilogue.

## Processors and FPGAs: qualification and test conditions

| Fact | Event | Source | Read | Kind | Ch |
|---|---|---|---|---|---|
| GR740, quad-core LEON4FT: QML Class V and Class Q ceramic versions; total dose 300 krad(Si); latch-up immunity to 125 MeV cm2/mg (conditions on the maker's datasheet, not on the page); in the European Preferred Parts List | as listed on the page | Frontgrade Gaisler, GR740 product page (undated) | 2026-11-23 | operator claim | 5 |
| Three GR740 processors, each running RTEMS, on NASA's Nancy Grace Roman Space Telescope | as listed on the page | RTEMS project home page (undated); a NASA source was not read | 2026-11-22 | operator claim | 5 |
| RAD750 powers more than 100 satellites and is used in the navigation payload of GPS III | as listed on the page | BAE Systems, Radiation-hardened electronics (undated) | 2026-11-22 | operator claim | 5 |
| RAD5545 SpaceVPX single board computer: one card can replace several earlier cards and can host software-defined radio payloads | as listed on the page | BAE Systems, same page | 2026-11-22 | operator claim | 5, 8 |
| HPSC: contract of 50 million dollars awarded to Microchip | 2022-08-15 | NASA JPL news release, same date | 2026-11-22 | operator claim | 5 |

| Fact | Event | Source | Read | Kind | Ch |
|---|---|---|---|---|---|
| HPSC: critical design review 2024; tape-out mid 2025; first processors later in 2025; first email ("Hello Universe") February 2026; in power, performance, reliability and radiation testing and not declared space qualified | status of March 2026 | NASA, HPSC project page (latest status dated March 2026) | 2026-11-23 | operator claim | 5 |
| PIC64-HPSC1000-RH early radiation results: total dose 200 krad(Si); latch-up tested to 78.2 MeV cm2/mg; no uncorrected errors or external resets in single-event testing to date; kits to selected early access customers only | 2026-10-27 | Microchip release, same date, as republished by a news aggregator; the maker's site refused the reading | 2026-11-22 | operator claim (republished release) | 5 |
| VIKRAM3201 and KALPANA3201: first production lots handed over; VIKRAM3201 qualified for launch vehicle conditions (180 nm, made by SCL) and validated in space in the mission management computer of POEM-4 on PSLV-C60; VIKRAM1601 in launch vehicle avionics since 2009 | handover 2025-03-05 | ISRO, VIKRAM3201 page (undated) | 2026-11-23 | operator claim | 5 |

| Fact | Event | Source | Read | Kind | Ch |
|---|---|---|---|---|---|
| Virtex-5QV: hardened by design; 131,072 logic cells; 1,000 krad (material not stated); latch-up immunity above LET 125 MeV cm2/mg (no test conditions given); "V-Flow (QML-V Equivalent)"; shipping. Virtex-4QV: last time buy, with "PDN Issued 10/21" on the maker's slide | status of March 2025 | AMD presentation at ESA SEFUW, 2025-03-25 | 2026-11-23 | operator claim | 6, 7 |
| XQRKU060: radiation tolerant; 725,550 logic cells; 120 krad (material not stated); latch-up immunity at LET 80 MeV cm2/mg (no test conditions given); Class B and Class Y qualification passed (B-Flow and Y-Flow, QML-Q and QML-Y equivalent); shipping since September 2020 | status of March 2025 | AMD, same presentation | 2026-11-23 | operator claim | 6, 7 |
| XQRVC1902 (7 nm, 400 AI engine tiles): Class B qualification completed | 2022-11-15 | AMD investor relations release, same date | 2026-11-22 | operator claim | 6, 7, 10 |

| Fact | Event | Source | Read | Kind | Ch |
|---|---|---|---|---|---|
| XQRVC1902: Class Y for 15-year missions under evaluation; Class B supports missions up to 7 years. XQRVE2302: Class B qualification completed, shipping planned for the fourth quarter of 2025. Radiation tests on 20 VC1902 parts from 5 wafer lots: no latch-up at maximum supply voltage and junction temperature to LET 80 MeV cm2/mg; total dose passed at 120 krad(Si), gamma, under 18 krad/min | status of March 2025 | AMD, ESA SEFUW presentation, 2025-03-25 | 2026-11-23 | operator claim | 6, 7, 10 |
| Zynq UltraScale+ MPSoC and RFSoC (16 nm): defense grade, with "potential for proton based missions"; not offered as space grade | status of March 2025 | AMD, same presentation | 2026-11-23 | operator claim | 6 |
| RT PolarFire RTPF500ZT: DLA QML Class V achieved, called "flight proven" by the maker. RT PolarFire SoC RTPFS460ZT (64 bit RISC-V cores): QML Class Q and MIL-STD-883 Class B completed | 2026-11-19 | Microchip release, same date, as republished; the maker's site refused the reading | 2026-11-22 | operator claim (republished release) | 6, 7 |
| NG-ULTRA (28 nm FD-SOI, SRAM configuration, quad Cortex-R52): first product qualified to ESCC 9030; total dose up to 50 krad(Si); latch-up immunity tested to 65 MeV cm2/mg; supply chain in the European Union | 2026-01-28 | NanoXplore and STMicroelectronics release, same date, as republished; the ESCC qualified parts list was not read | 2026-11-22 | operator claim (republished release) | 6, 7 |

| Fact | Event | Source | Read | Kind | Ch |
|---|---|---|---|---|---|
| UT24CP1008 (28 nm FD-SOI): 96,000 logic cells; 100 krad(Si) at an effective dose rate of 1 rad(Si)/s; "LET 80 or less" with no effect named; Space PEM QD flow with L1 and L2 options; 484-ball plastic package | as listed on the page | Frontgrade, UT24CP1008 product page (undated) | 2026-11-23 | operator claim | 6, 7 |
| NOEL-V RISC-V processor: called "flight proven in Earth orbit" by its maker; the open version has no fault tolerance, the commercial version adds cache error correction and scrubbing | as listed on the page | Frontgrade Gaisler, NOEL-V product page (undated) | 2026-11-22 | operator claim | 5, 6, 7 |
| Sirius on-board computer: LEON3FT soft core at 50 MHz, triplication on all flip-flops, 1.3 W nominal, 20 krad (material not stated) | as listed on the page | AAC Clyde Space, Sirius OBC page (undated) | 2026-11-22 | operator claim | 4 |
| Xiphos Q7 (Zynq-7020): in orbit since June 2016; tested to 25 krad (material and tester not stated); about 2 W | as listed on the page | Product listing on a marketplace site (undated; the maker's own wording not shown) | 2026-11-22 | operator claim | 6 |
| Xiphos Q8: Zynq UltraScale+ MPSoC, EDAC-protected LPDDR4 memory, 4 to 25 W depending on use | as listed on the page | Product listing on a marketplace site (undated; the maker's own wording not shown) | 2026-11-22 | operator claim | 6 |
| Unibap iX10-101: AMD Ryzen V1000 processor, Microchip PolarFire FPGA, optional Intel Movidius Myriad X, 20 to 40 W | as listed on the page | Product listing on a marketplace site (undated; the maker's own wording not shown) | 2026-11-22 | operator claim | 6 |

| Fact | Event | Source | Read | Kind | Ch |
|---|---|---|---|---|---|
| AMD's list of missions using its parts: Virtex-5 for on-board computing on Artemis and Orion; Virtex-5 and Virtex-4 in instruments on Chandrayaan-3; Kintex UltraScale in a 2024 imaging constellation; Versal parts in constellations launched in 2023 and 2024, grade not stated | status of March 2025 | AMD, ESA SEFUW presentation, 2025-03-25 | 2026-11-23 | operator claim | 6, 7 |
| Open flow: nextpnr lists NanoXplore NG-Ultra among its targets, with bitstream creation still requiring the maker's Impulse tool; Xilinx 7-series marked experimental; no AMD space part or Microchip PolarFire family listed | as of November 2026 | YosysHQ, nextpnr README (library copy of November 2026) | 2026-11-23 | operator claim | 7 |

## Accelerators flown or tested for flight

| Fact | Event | Source | Read | Kind | Ch |
|---|---|---|---|---|---|
| Intel Movidius Myriad 2 on Phi-Sat-1: launched September 2020 (2 September by the chip maker's date); cloud screening run in orbit | 2020-09 | Intel Newsroom, 2020; mission paper, IEEE TGRS, 2022 | 2026-11-23 | operator claim | 9, 10 |
| Phi-Sat-1 in flight: built-in self-test in four phases over 70 days, all passed except an SD card not used by the mission; 612 tiles from 17 cubes checked, 96 percent accuracy, no false positives | during the mission | Giuffrida and others, IEEE TGRS 60, 2022 | 2026-11-23 | measured (by the mission team) | 9, 10 |
| Kepler: about 40 Nvidia Orin processors on 10 satellites, 18 customers | reported 2026-04-13 | TechCrunch, same date | 2026-11-23 | operator claim (trade press) | 10, 12 |
| Starcloud-1 launched with an Nvidia H100; per the company, used to train a model and run a Google language model (model not settled: the report's wording was not checked against the company's account); a second processor, an Nvidia A6000, failed during launch | launch November 2025 | TechCrunch, 2026-03-30 | 2026-11-22 | operator claim (trade press) | 10, 12 |
| Google tensor processor (Trillium, v6e) in a 67 MeV proton beam: no hard failures attributable to total dose up to 15 krad(Si) on a single chip; memory irregularities after 2 krad(Si). A dose test, not a qualification | published 2025-11-04 | Google Research blog, same date | 2026-11-22 | operator claim | 10, 12 |

| Fact | Event | Source | Read | Kind | Ch |
|---|---|---|---|---|---|
| Intuition-1: 6U, 192-band hyperspectral sensor, Leopard data processing unit; launched November 2023; image processing in orbit at 3 trillion operations per second, filtering cloud-covered images | launch 2023-11 | KP Labs, Missions page (undated) | 2026-11-23 | operator claim | 9 |
| Phi-Sat-2: 6U, multispectral instrument; cloud detection network quantized to 8 bits. Launch month differs by source: August 2024 (KP Labs), March 2024 (another partner's mission list) | launch 2024 | KP Labs and Ubotica mission pages (undated); ESA's page was not read | 2026-11-23 | operator claim | 9, 10 |
| Three-Body Computing Constellation (China): first group of 12 spacecraft launched | summer 2025 (NASA); May 2025 (a 2026 preprint) | NASA, State-of-the-Art of Small Spacecraft Technology 2026, Integration, Launch and Deployment; Sun and others, arXiv 2605.12681 (preprint) | 2026-11-23 | published history (agency report) | 10, 12 |

## Computing platforms in orbit: status fields

| Fact | Event | Source | Read | Kind | Ch |
|---|---|---|---|---|---|
| Axiom Space: first two orbital data center nodes launched to low Earth orbit with the first tranche of Kepler's optical relay constellation; over-the-air software and model updates supported; access by enquiry. The page gives the optical links as "2.5 gigabytes per second-capable" in its text and "2.5 Gbps" in a linked title; either is a link rate, not customer throughput | launch 2026-01-11 | Axiom Space, Orbital Data Centers page (undated) | 2026-11-24 | operator claim | P, 10, 12 |
| Axiom Space Data Center Unit-1, a data processing prototype, deployed on the International Space Station, to demonstrate initial capabilities | fall 2025 | Axiom Space, same page | 2026-11-24 | operator claim | 12 |
| Spire: "Software in Space" for a flat monthly fee; amount not stated | as listed on the page | Spire, Space Services page (undated) | 2026-11-22 | operator claim | 12 |
| Loft Orbital: On-Orbit AI offered; 16 satellites launched, more than 25 missions deployed | as listed on the page | Loft Orbital home page (undated) | 2026-11-22 | operator claim | 12 |
| OPS-SAT: launched December 2019; NanoSat MO Framework first executed in space in 2020 | 2019 to 2020 | ESA, NanoSat MO Framework documentation | 2026-11-23 | operator claim | 12 |

| Fact | Event | Source | Read | Kind | Ch |
|---|---|---|---|---|---|
| OPS-SAT Space Lab: no charge, contract or paperwork; OPS-SAT PRETTY "available now", 50 percent of mission time to the lab; OPS-SAT VOLT and ORIOLE listed "Launch '27" | as listed on the page | ESA, OPS-SAT Space Lab (undated) | 2026-11-22 | operator claim | 12 |
| Spaceborne Computer-2 on the International Space Station: launched; hosting edge computing and AI experiments since 2021 | launch 2021-02-20 | Hu, Alhassan and Iqbal, arXiv 2610.32260, 2026; the maker's and NASA's pages were not read | 2026-11-23 | published history (preprint, not peer reviewed) | 12 |
| Project Suncatcher (Google): announced, with two prototype satellites with Planet "by early 2027" | 2025-11-04 | Google Research blog, same date | 2026-11-22 | operator claim | 10, 12 |
| A Project Suncatcher prototype on the Transporter-18 rideshare: a launch earlier than the planned date, not a conflict with it | launch 2026-10-01 | Space.com, same period | 2026-11-22 | operator claim (trade press) | 12 |
| Project Suncatcher launch with Blue Origin on New Glenn, "targeted for Q1 2029" | announced 2026-11-19 | Google, The Keyword blog listing, same date | 2026-11-23 | operator claim | 12 |
| An application to the FCC for data center satellites, as reported; the filing itself was not read | reported for February 2026 | van Berkel, arXiv 2607.14172, 2026 | 2026-11-23 | operator claim (preprint, not peer reviewed) | 12 |

| Fact | Event | Source | Read | Kind | Ch |
|---|---|---|---|---|---|
| Starcloud 2, a larger satellite with several processors including an Nvidia Blackwell chip, "later this year"; whether it launched was not checked | announced 2026-03-30 | TechCrunch, same date | 2026-11-23 | operator claim (trade press) | 12 |
| Launch minimum used in chapter 11: SpaceX Transporter rideshare "starting at $350k for ~50 kg", a booking minimum that leaves out deployment hardware, insurance and integration | chapter dated 2026-05-18 | NASA, State of the Art of Small Spacecraft Technology, Integration, Launch and Deployment | 2026-11-23 | operator claim (as reported by NASA) | 11 |

## Software-defined payloads

| Fact | Event | Source | Read | Kind | Ch |
|---|---|---|---|---|---|
| Eutelsat Quantum launched; "can be completely repurposed while in space" | 2021-07-30, 23:00 CEST | ESA press release 25-2021 | 2026-11-22 | operator claim | 8 |
| O3b mPOWER: 13 satellites after F11 to F13 launched on a Falcon 9 | 2026-09-13 | Boeing feature, September 2026 | 2026-11-22 | operator claim | 8 |
| O3b mPOWER: final three expected operational mid 2027; 229 million dollars of insurance collected on satellites 1 to 4, cause not stated in the documents read | 2026-11-05 | SES, results release for nine months and third quarter 2026, same date | 2026-11-23 | operator claim | 8 |
| Iridium: on-air trials of a narrowband service using new waveform algorithms on its "software-defined satellites", activated by software update | 2026-01-21 | Iridium release, same date | 2026-11-23 | operator claim | 8 |
| Space INSPIRE: first satellite ASTRA 1Q "scheduled for launch in 2026"; no source read confirms a launch | 2025-05-28 | ESA article, same date | 2026-11-22 | operator claim | 8 |
| SmartSat announced with two demonstration missions | 2019-03-20 | Lockheed Martin release, same date | 2026-11-22 | operator claim | 8 |
| TianZhi-1, called "China's first software-defined satellite" by its application's authors: an uploaded application ran on a smartphone-based platform in orbit | paper of 2019 | ISKE 2019 conference paper by the application's authors | 2026-11-23 | operator claim (the application's authors) | 8 |

| Fact | Event | Source | Read | Kind | Ch |
|---|---|---|---|---|---|
| 3GPP Release 17, protocol specifications frozen June 2022, the first release with normative requirements for non-terrestrial networks; Release 19 work adds regenerative payloads | as listed on the page | 3GPP, Non-Terrestrial Networks overview page (undated) | 2026-11-22 | published history (the standards body's overview; the specificatio ns not read) | 8 |
| CCSDS issues cited in peer-reviewed papers (current status not checked): 121.0-B-3 (2020), 122.0-B-2 (2017), 123.0-B-2 (2019), file delivery 727.0-B-5 (2020) | as cited | Reference lists of peer-reviewed papers (Remote Sensing, 2023); the Blue Books themselves were not read | 2026-11-23 | published history (standards identified through peer-reviewed citations; their text not read) | 9 |

## Sought and not read

These were looked for on the date given; where the book prints a related fact, it comes from another source named in the text (for example a paper's citation or a partner's page), never from these. Each is a candidate for the next refresh; none is a finding that the thing does not exist.

| Sought | Where | Date |
|---|---|---|
| Airbus OneSat: what is reconfigurable, first delivery or launch | Airbus product pages; ESA program pages | 2026-11-23 |
| Hughes JUPITER 3 and ViaSat-3: the operators' own description of in-orbit reconfiguration | Hughes and Viasat pages | 2026-11-23 |
| Japan's Engineering Test Satellite 9: digital payload and launch | JAXA pages | 2026-11-23 |
| An Indian software-defined payload case | ISRO pages | 2026-11-23 |
| The cause of the O3b mPOWER 1 to 4 insurance claim | SES pages and releases | 2026-11-23 |

| Sought | Where | Date |
|---|---|---|
| Galileo's achieved telemetry rates and the decibels gained by each ground improvement | JPL's Galileo telecommunications monograph | 2026-11-23 |
| The CCSDS compression and file delivery Blue Books themselves (identified in the book through peer-reviewed citations) | CCSDS publications pages | 2026-11-23 |
| Phi-Sat-1's status on the book's date; Phi-Sat-2's launch date from ESA | ESA pages | 2026-11-23 |
| Flight data for the AI engines of a space-grade adaptive SoC | Maker and mission pages | 2026-11-23 |
| The builder's own description of the Three-Body Computing Constellation | Zhejiang Lab and Xinhua English pages | 2026-11-23 |
| The FCC filing for data center satellites reported for February 2026 | FCC filing system | 2026-11-23 |
| Spaceborne Computer-1 and -2: dates and results from the maker or NASA | HPE and NASA pages | 2026-11-23 |
| OPS-SAT's end date; OPS-SAT PRETTY's launch date and orbit | ESA pages | 2026-11-23 |
| Thales Alenia Space's ASCEND study and its conclusions | Thales Alenia Space pages | 2026-11-23 |
| Health or service of any launched platform on the book's date (Axiom Space's nodes, Kepler's cluster, the Suncatcher prototype) | Operators' pages and filings | 2026-11-23 |

## What this register leaves out

Prices, lead times and order books: apart from the one dated booking minimum above, none is printed in the book, because none was found from a dated public source that says what it includes. Capacity in orbit for any system that has only been announced: the book gives none, by rule. History that will not change, such as the Galileo antenna of 1991 or the computers of chapter 2, is cited in its chapter and in Appendix E, not here.

APPENDIX E

# Sources

Every dated fact in this book was checked against the source listed for its chapter in November 2026. Primary sources come first: original papers, standards, treaties and regulations, regulator filings and orders, official reports, operator documents and institutional histories. Trade press is labeled as such. Every number in a Link Ledger also carries one of seven evidence classes: measured, calculated, simulated, operator claim, standard, assumption (to verify) or published history. This list holds 289 chapter source entries, including repeated works and notes on the illustrative case. It is drawn from the book's claim tables, which pair each claim with its source and the date it was checked; those tables are part of the book's sources.

## Reader's Brief: Four Budgets, Three Places, Separate Status Fields

- satsearch, Xiphos Q7 Processor listing https://satsearch.co/products/xiphos-q7-processor
- Microchip press release via GlobeNewswire, republished by Finviz (trade press republication) https://finviz.com/news/425099/microchip-expands-rt-polarfire-fpga-portfolio-with-new-spaceflight-qualifications

## Prologue: Two Computers, Fifty-Seven Years Apart

- NASA, Computers in Spaceflight: The NASA Experience, chapter 2 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch2-5.html
- NASA, Computers in Spaceflight, chapter 2, "The need for an on-board computer" https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch2-1.html
- NASA, Computers in Spaceflight, chapter 2, "The Apollo computer systems" https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch2-3.html
- Axiom Space, Orbital Data Centers https://axiomspace.com/orbital-data-center
- TechCrunch (trade press), The largest orbital compute cluster is open for business https://techcrunch.com/2026/04/13/the-largest-orbital-compute-cluster-is-open-for-business/
- TechCrunch (trade press), Starcloud raises 170 million Series A https://techcrunch.com/2026/03/30/starcloud-raises-170-million-series-ato-build-data-centers-in-space
- Intel Newsroom, Intel Powers First Satellite with AI on Board https://www.intel.com/content/www/us/en/newsroom/news/first-satellite-ai.html
- NASA, Moon Fact Sheet https://nssdc.gsfc.nasa.gov/planetary/factsheet/moonfact.html; speed of light 299,792,458 m/s (NIST https://physics.nist.gov/cgi-bin/cuu/Value?c )
- Calculated
- E. C. Hall, MIT's Role in Project Apollo, Final Report, Volume III: Computer Subsystem, R-700, MIT Charles Stark Draper Laboratory, August 1972 https://www.ibiblio.org/apollo/Documents/R-700.pdf

## Chapter 1. Where Should the Computing Happen?

- NASA, Computers in Spaceflight: The NASA Experience, chapter 2, "The Apollo guidance computer: Hardware" https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch2-5.html
- NASA, Computers in Spaceflight, chapter 6 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch6-2.html
- The second book of this series, From Telstar to the Phone in Your Pocket, chapter 13
- Calculated: T = 2 pi sqrt(a^3/GM), a = 6,871 km (mean Earth radius 6,371 km plus 500 km), GM = 398,600.4418 km^3/s^2 (WGS 84, NGA.STND.0036)

- Calculated from the same inputs
- Calculated: epsilon sigma T^4 with sigma = 5.670374419e-8 W m^-2 K^-4 (NIST CODATA https://physics.nist.gov/cgi-bin/cuu/Value?sigma )
- Assumed for the illustration; not a real satellite. Settled for a real case by the operator's stated figures
- Calculated from the illustrative inputs
- Calculated from the example's stated inputs

## Chapter 2. 1962 to 1996: Computers That Had to Fly, Line by Line

- NASA, Apollo Lunar Surface Journal, Apollo 11, "The Landing" (corrected transcript and commentary, ed. E. M. Jones) https://www.nasa.gov/wp-content/uploads/static/history/alsj/a11/a11.landing.html
- Apollo Lunar Surface Journal, Apollo 11 landing, Aldrin commentary
- P. Adler, "Apollo 11 Program Alarms" (1998), in the Apollo Lunar Surface Journal https://www.nasa.gov/wp-content/uploads/static/history/alsj/a11/a11.1201-pa.html
- NASA, Computers in Spaceflight: The NASA Experience (1988), ch. 1 https://www.nasa.gov/wp-content/uploads/static/history/computers/ch1-1.html
- Computers in Spaceflight, ch. 1-1
- Computers in Spaceflight, ch. 1-2 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch1-2.html
- Computers in Spaceflight, ch. 1-2
- Computers in Spaceflight, ch. 1-5 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch1-5.html
- Computers in Spaceflight, ch. 5-3 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch5-3.html
- Computers in Spaceflight, ch. 2-1 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch2-1.html
- Computers in Spaceflight, ch. 2-5 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch2-5.html
- Computers in Spaceflight, ch. 2-5
- Computers in Spaceflight, ch. 2-6 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch2-6.html
- Computers in Spaceflight, ch. 2-7 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch2-7.html
- Computers in Spaceflight, ch. 8-2 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch8-2.html
- Computers in Spaceflight, ch. 4-3 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch4-3.html
- Computers in Spaceflight, ch. 4-4 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch4-4.html
- Computers in Spaceflight, ch. 4-5, Box 4-5 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch4-5.html
- NASA History, "40 Years Ago: The Countdown Begins for STS-1; First Launch Attempt Scrubbed" https://www.nasa.gov/history/40-years-ago-the-countdown-begins-for-sts-1-first-launch-attempt-scrubbed/
- Computers in Spaceflight, ch. 5-6 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch5-6.html
- Computers in Spaceflight, ch. 6-2 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch6-2.html
- Computers in Spaceflight, ch. 6-2
- Computers in Spaceflight, ch. 6-3 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch6-3.html
- Computers in Spaceflight, ch. 6-3
- D. McComas, NASA Goddard, "Lessons from 30 Years of Flight Software" presentation, NASA Goddard Space Flight Center, 22 September 2015
- calculated; mean Earth-Moon distance 384,400 km
- Computers in Spaceflight, ch. 2-5, Box 2-1
- E. C. Hall, MIT's Role in Project Apollo, Final Report, Volume III: Computer Subsystem, R-700, MIT Charles Stark Draper Laboratory, August 1972 https://www.ibiblio.org/apollo/Documents/R-700.pdf

## Chapter 3. Radiation: Effects, Environment and Assurance

- A. C. Dickieson, "The Telstar Experiment", Bell System Technical Journal 42 (4), July 1963, pp. 739 to 746 https://bitsavers.mirrorservice.org/magazines/Bell_System_Technical_Journal/BSTJ_V42N04_196307_Part_1.pdf
- Dickieson, BSTJ, July 1963 (as above)
- W. L. Brown and others, "The Spacecraft Radiation Experiments", BSTJ 42 (4), July 1963, p. 899 ff. (same file)
- NASA History, "Telstar Opened Era of Global Satellite Television" https://www.nasa.gov/history/telstar-opened-era-of-global-satellite-television conversion calculated
- ESA, "Radiation: satellites' unseen enemy", 4 July 2011 https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Radiation_satellites_unseen_enemy

- J. Pellish, NASA GSFC, Tutorial: Radiation Effects in Electronic Systems, ASEC, May 2017 https://ntrs.nasa.gov/api/citations/20170004674/downloads/20170004674.pdf
- Pellish 2017 (as above)
- Pellish 2017 (as above), after S. Buchner
- ESA, "Radiation: satellites' unseen enemy", 4 July 2011 (as above)
- ESA, "Radiation: a testing challenge", 4 July 2011, linked continuation of the page above
- NASA, State-of-the-Art of Small Spacecraft Technology, 8.0 Small Spacecraft Avionics, section 8.2.1 https://www.nasa.gov/smallsat-institute/sst-soa/small-spacecraft-avionics/ (page updated 2026-10-21)
- Pellish 2017, after S. Buchner, SERESSA 2011
- Pellish 2017
- A. Kazak, S. Popa, A. Bertescu, M. Ivanovici, "Simulated On-Board AI-Based Classification of Radiation-Induced SRAM Event Upsets", Electronics 15 (2026) 2814 https://doi.org/10.3390/electronics15132814
- B. Rodiger and others, DLR, "Qualification of Inter-Satellite Link Laser Communication Terminals on CubeSats: CubeISL", SmallSat Conference 2023, SSC23-IX-04 https://doi.org/10.26077/vgtd-de17
- Rodiger and others 2023 (as above); product calculated
- V. Knap, L. K. Vestergaard, D.-I. Stroe, "A Review of Battery Technology in CubeSats and Small Satellite Solutions", Energies 13 (2020) 4097 https://doi.org/10.3390/en13164097 ratio calculated
- Knap and others 2020 (as above), reference list
- S. G. Turyshev, "Orbital Data Centers: Spacecraft Constraints and Economic Viability", arXiv 2604.27197 (preprint posted 29 April 2026, not peer reviewed) https://arxiv.org/abs/2604.27197
- Frontgrade Gaisler, GR740 product page https://www.gaisler.com/products/gr740
- satsearch, Xiphos Q7 listing https://satsearch.co/products/xiphos-q7-processor (as used in chapter 6)
- M. Berg and K. LaBel, NASA GSFC, "New Developments in Error Detection and Correction Strategies for Critical Applications", SEE Symposium and MAPLD 2016 https://ntrs.nasa.gov/citations/20160007971
- Berg and LaBel 2016 (as above)
- NASA GSFC Engineering and Technology Directorate, Radiation Effects and Analysis https://etd.gsfc.nasa.gov/capabilities/capabilities-listing/radiation-effects-and-analysis/ (page updated 2026-11-18)
- NASA NTRS, Compendium of Single Event Effects, Total Ionizing Dose, and Displacement Damage for Candidate Spacecraft Electronics for NASA (2014) https://ntrs.nasa.gov/citations/20140017802
- AMD at ESA SEFUW, 25 March 2025 (as in chapter 6)
- Turyshev, arXiv 2604.27197 (as above), Table IX
- Dickieson, BSTJ, July 1963
- Arithmetic on the Poisson distribution, stated in the text

## Chapter 4. Mitigation, Layer by Layer, and What It Leaves

- A. G. Schmidt, M. French, T. Flatley, "Radiation Hardening by Software Techniques on FPGAs: Flight Experiment Evaluation and Results", IEEE Aerospace Conference, March 2017 https://doi.org/10.1109/aero.2017.7943651
- Schmidt and others 2017 (as above)
- Schmidt and others 2017
- M. Berg and K. LaBel, NASA GSFC, "New Developments in Error Detection and Correction Strategies for Critical Applications", SEE Symposium and MAPLD Workshop 2016 https://ntrs.nasa.gov/citations/20160007971
- Berg and LaBel 2016
- derived from Berg and LaBel's reliability expressions
- AMD, K. O'Neill and P. Lynch, AMD XQR Versal Adaptive SoCs, ESA SEFUW, 25 March 2025 https://indico.esa.int/event/531/contributions/10592/attachments/6538/11610/
- NASA, State-of-the-Art of Small Spacecraft Technology, 8.0 Small Spacecraft Avionics https://www.nasa.gov/smallsat-institute/sst-soa/small-spacecraft-avionics/ (page updated 2026-10-21)
- NASA SoA 2026 (as above)
- AAC Clyde Space, Smallsat Sirius OBC https://www.aac-clyde.space/what-we-do/space-products-components/command-data-handling/smallsat-sirius-obc
- Frontgrade Gaisler, GR740 product page https://www.gaisler.com/products/gr740
- as sourced in chapter 2: NASA, Computers in Spaceflight (1988), chapter 4, and NASA History, "40 Years Ago: The Countdown Begins for STS-1; First Launch Attempt Scrubbed" (2021)
- as sourced in chapter 2

- Arithmetic on the Poisson distribution, stated in the text

## Chapter 5. Processors and the Avionics Around Them, 1990 to 2026


- NASA, High Performance Spaceflight Computing (HPSC) https://www.nasa.gov/high-performance-spaceflight-computing-hpsc/
- NASA JPL, NASA Awards Next-Generation Spaceflight Computing Processor Contract https://www.jpl.nasa.gov/news/nasa-awards-next-generation-spaceflight-computing-processor-contract/
- Microchip press release via GlobeNewswire, republished by Finviz (trade press) https://finviz.com/news/408662/microchip-achieves-key-program-milestones-for-its-pic64-hpsc-spaceflight-computing-series
- K. A. Miller and A. J. Lohn, Onboard AI: Constraints and Limitations, Center for Security and Emerging Technology, August 2023 https://doi.org/10.51593/2022ca008
- NASA, spacewasm repository README https://github.com/nasa/spacewasm/blob/HEAD/README.md
- Towards On-Board SAR Processing with FPGA Accelerators and a PCIe Interface, Electronics 12 (2023) 2558 https://doi.org/10.3390/electronics12122558
- D. McComas, NASA Goddard Space Flight Center, Lessons from 30 Years of Flight Software, presentation, 22 September 2015
- McComas, Lessons from 30 Years of Flight Software (2015)
- BAE Systems, Radiation-hardened electronics https://www.baesystems.com/en-us/product/radiation-hardened-electronics
- BAE Systems, Radiation-hardened electronics
- C. M. Fuchs, Fault-tolerant satellite computing with modern semiconductors, doctoral thesis, Leiden University, 2019 https://hdl.handle.net/1887/82454
- RTEMS, bsps/sparc/erc32/README.md https://github.com/RTEMS/rtems/blob/HEAD/bsps/sparc/erc32/README.md
- S. S. Yuhaniz, Intelligent image processing on board small earth observation satellites, doctoral thesis, University of Surrey, 2008 https://openalex.org/W2787305217
- Frontgrade Gaisler, GR740 https://www.gaisler.com/products/gr740
- calculated from the GR740 page's PC100 interface
- RTEMS project home page https://www.rtems.org/
- Frontgrade Gaisler, NOEL-V https://www.gaisler.com/products/noel-v
- ISRO, VIKRAM3201 https://www.isro.gov.in/vikram3201.html
- T. Kuwahara et al., On-Board Computers for Micro-Satellites, Trans. JSASS Aerospace Tech. Japan 19 (2021) 485 to 492 https://doi.org/10.2322/tastj.19.485
- S. Kobayashi et al., Software Development Framework for Small Satellite On-board Computers, Trans. JSASS Aerospace Tech. Japan 12 (2014) Tf_1 to Tf_6 https://doi.org/10.2322/tastj.12.tf_1
- A Survey on Resource Management in Joint Communication and Computing-Embedded SAGIN, IEEE Communications Surveys and Tutorials (2024) https://doi.org/10.1109/comst.2024.3421523
- NASA, Computers in Spaceflight, ch. 6-2 and 6-3 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch6-2.html
- An Efficient Architecture and High-Throughput Implementation of CCSDS-123.0-B-2 Hybrid Entropy Coder, arXiv 2205.04123 (preprint, not peer reviewed) https://arxiv.org/abs/2205.04123


## Chapter 6. Part Choice and Qualification Evidence: A Matrix, Not a Ladder


- Microchip release, Microchip Expands RT PolarFire FPGA Portfolio with New Spaceflight Qualifications, GlobeNewswire, Chandler, 19 November 2026, as republished at https://finviz.com/news/425099/microchip-expands-rt-polarfire-fpga-portfolio-with-new-spaceflight-qualifications (microchip.com refused the request)
- AMD, K. O'Neill and P. Lynch, AMD XQR Versal Adaptive SoCs Enable Next-Generation Signal Processing and AI in Space, ESA Space FPGA Users Workshop (SEFUW), 25 March 2025 https://indico.esa.int/event/531/contributions/10592/attachments/6538/11610/
- AMD SEFUW deck, 25 March 2025 (as above)
- AMD SEFUW deck, 25 March 2025
- NanoXplore and STMicroelectronics release, NanoXplore and STMicroelectronics Deliver European FPGA for Space Missions, 28 January 2026, republished by the Edge AI and Vision Alliance https://www.edge-ai-vision.com/2026/01/nanoxplore-and-stmicroelectronics-deliver-european-fpga-for-space-missions/
- Frontgrade, UT24CP1008 product page https://www.frontgrade.com/product/ut24cp1008

- Frontgrade Gaisler, NOEL-V product page https://www.gaisler.com/products/noel-v
- satsearch, Xiphos Q7 Processor listing https://satsearch.co/products/xiphos-q7-processor
- satsearch, Xiphos Q8 Processor listing https://satsearch.co/products/xiphos-q8-processor
- satsearch, Unibap iX10-101 listing https://satsearch.co/products/unibap-ix10-101-computer
- NASA, State-of-the-Art of Small Spacecraft Technology, chapter 8, Small Spacecraft Avionics, 19 May 2026 https://www.nasa.gov/smallsat-institute/sst-soa/small-spacecraft-avionics
- NASA, Small Spacecraft Avionics, 19 May 2026 (as above)
- Calculated from AMD SEFUW deck, slide 6


## Chapter 7. Space FPGAs: Where the Configuration Lives, What Protects the Rest, and the Process Under It


- NanoXplore and STMicroelectronics release, NanoXplore and STMicroelectronics Deliver European FPGA for Space Missions, 28 January 2026, republished by the Edge AI and Vision Alliance https://www.edge-ai-vision.com/2026/01/nanoxplore-and-stmicroelectronics-deliver-european-fpga-for-space-missions/
- M. Berg and K. LaBel, NASA GSFC, New Developments in Error Detection and Correction Strategies for Critical Applications, presented at the 2016 SEE Symposium and MAPLD Workshop https://ntrs.nasa.gov/citations/20160007971
- Berg and LaBel, NASA GSFC, 2016 (as above)
- M. Stettler, M. Caffrey, P. Graham, J. Krone, Los Alamos National Laboratory, Radiation effects and mitigation strategies for modern FPGAs, 10th Workshop on Electronics for LHC and Future Experiments, 2004, CERN-2004-010, p. 112, doi:10.5170/cern-2004-010.112 (labeled scholarship)
- AMD, K. O'Neill and P. Lynch, AMD XQR Versal Adaptive SoCs Enable Next-Generation Signal Processing and AI in Space, ESA SEFUW, 25 March 2025 https://indico.esa.int/event/531/contributions/10592/attachments/6538/11610/
- AMD SEFUW deck, 25 March 2025 (as above)
- AMD, AMD Announces Completion of Class B Qualification for First Space-Grade Versal Adaptive SoCs https://ir.amd.com/news-events/press-releases/detail/1101/amd-announces-completion-of-class-b-qualification-for-first
- Microchip release, 19 November 2026, GlobeNewswire, as republished at https://finviz.com/news/425099/microchip-expands-rt-polarfire-fpga-portfolio-with-new-spaceflight-qualifications
- P. Antunes and A. Podobas, KTH, FPGA-Based Neural Network Accelerators for Space Missions: A Survey, arXiv 2504.16173, 2025 (preprint, not peer reviewed)
- V. Leon and others, Development of High-Performance DSP Algorithms on the European Rad-Hard NG-ULTRA SoC FPGA, IEEE ICECS 2024, arXiv 2409.12253
- Frontgrade, UT24CP1008 product page https://www.frontgrade.com/product/ut24cp1008
- AAC Clyde Space, Smallsat Sirius OBC https://www.aac-clyde.space/what-we-do/space-products-components/command-data-handling/smallsat-sirius-obc
- Frontgrade Gaisler, NOEL-V product page https://www.gaisler.com/products/noel-v
- YosysHQ, Yosys documentation https://yosyshq.readthedocs.io/projects/yosys/en/latest/
- YosysHQ, nextpnr README https://github.com/YosysHQ/nextpnr (copy of November 2026)
- Calculated from AMD SEFUW deck, slide 31
- Stettler and others, Los Alamos, 2004 (as above)


## Chapter 8. Software-Defined Payloads: What Can Be Changed from the Ground


- ESA, press release N 25-2021, Reprogrammable satellite deployed in space https://esa.int/Newsroom/Press_Releases/Reprogrammable_satellite_deployed_in_space
- 3GPP, Non-Terrestrial Networks (NTN) overview https://www.3gpp.org/technologies/ntn-overview
- Boeing, Leveling up service with launch of last satellites https://www.boeing.com/features/2026/09/leveling-up-service-with-launch-of-last-satellites
- SES, O3b mPOWER https://www.ses.com/network-and-technology/meo/o3b-mpower
- SES, SES Delivers Solid 9 Months and Q3 2026 Results, 2026-11-05 https://www.ses.com/news/press-release/ses-delivers-solid-9-months-q3-2026-results-and-reiterates-fullyear-outlook
- SES, SES Delivers Solid 9 Months and Q3 2026 Results (release via STT Info) https://www.sttinfo.fi/tiedote/72477880/ses-delivers-solid-9-months-and-q3-2026-results-and-reiterates-full-year-outlook?publisherId=58763726

- SES O3b mPOWER page and results release above
- Iridium, On-Air Trials Underway: Iridium NTN Direct Prepares to Enter Beta, 2026-01-21 https://investor.iridium.com/2026-01-21-On-Air-Trials-Underway-Iridium-NTN-Direct-Prepares-to-Enter-Beta-as-Testing-Continues
- ESA, ESA-supported Space INSPIRE product line makes next step to launch https://resilience.esa.int/?p=11674
- Lockheed Martin, Lockheed Martin's First Smart Satellites are Tiny with Big Missions https://news.lockheedmartin.com/2019-03-20-Lockheed-Martins-First-Smart-Satellites-are-Tiny-with-Big-Missions
- Design and Implementation of On-orbit Valuable Image Extraction for the TianZhi-1 Satellite, ISKE 2019, doi:10.1109/iske47853.2019.9170453; abstract read in the OpenAlex record https://api.openalex.org/works/W3061675527
- BAE Systems, Radiation-hardened electronics https://www.baesystems.com/en-us/product/radiation-hardened-electronics
- Demand-Aware Onboard Payload Processor Management for High Throughput NGSO Satellite Systems, IEEE Transactions on Aerospace and Electronic Systems, 2023, doi:10.1109/taes.2023.3245044; abstract read in the OpenAlex record https://api.openalex.org/works/W4321609062
- A methodology to benchmark flexible payload architectures in a megaconstellation use case, International Journal of Satellite Communications and Networking, 2020, doi:10.1002/sat.1344; abstract read in the OpenAlex record https://api.openalex.org/works/W3010137513
- The first book of this series, The Bit's Path, chapter 8
- Airbus, Hughes, Viasat, JAXA and ISRO pages
- The first book of this series, The Bit's Path, ledger LL-1998-01

## Chapter 9. Processing the Data On Board: Compress, Filter, Decide

- JPL, Galileo Jupiter Arrival press kit, December 1995 https://www.jpl.nasa.gov/news/press_kits/gllarpk.pdf
- Same press kit
- JPL, Galileo fact sheet (undated, written before the end of mission) https://d2pn8kiwq2w21t.cloudfront.net/documents/galileo_PuKN96n.pdf
- NASA JPL, Galileo mission page https://www.jpl.nasa.gov/missions/galileo
- Proposed data compression schemes for the Galileo S-band contingency mission, NASA Technical Reports Server, 1993; abstract read in the OpenAlex record https://api.openalex.org/works/W1567427456
- Optimizing the Galileo Space Communication Link, NASA Technical Reports Server, 1994; abstract read in the OpenAlex record https://api.openalex.org/works/W1655703714
- Klaasen and others, Calibration and performance of the Galileo solid-state imaging system in Jupiter orbit, Optical Engineering 38, 1999, doi:10.1117/1.602168; abstract read in the OpenAlex record https://api.openalex.org/works/W2115983945
- Copernicus SentiWiki, S1 Mission https://sentiwiki.copernicus.eu/web/s1-mission
- Langer and others, Robust and Reconfigurable On-Board Processing for a Hyperspectral Imaging Small Satellite, Remote Sensing 15(15) 3756, 2023, doi:10.3390/rs15153756, references 25, 46 to 48; Chatziantoniou and others, arXiv 2205.04123 (2022), introduction
- Chatziantoniou and others, An Efficient Architecture and High-Throughput Implementation of CCSDS-123.0-B-2 Hybrid Entropy Coder, arXiv 2205.04123 (published in Remote Sensing, 2022)
- Hyperspectral image compression onboard next-generation satellites: implementation solutions on GPU and FPGAs https://openalex.org/W2282214111
- AI-Based On-Board Maritime Object Detection for Earth Observation Payload Data Reduction on Versal Embedded Hardware, arXiv 2610.12182, 2026 (preprint, not peer reviewed)
- How Can Optical Communications Shape the Future of Deep Space Communications? A Survey, IEEE Communications Surveys and Tutorials, 2024, doi:10.1109/comst.2024.3403873, reference 138
- Lessons from 30 Years of Flight Software, NASA https://openalex.org/W2121920352
- Giuffrida and others, The Phi-Sat-1 Mission: The First On-Board Deep Neural Network Demonstrator for Satellite Earth Observation, IEEE Transactions on Geoscience and Remote Sensing 60, 2022, doi:10.1109/tgrs.2021.3125567
- Same paper
- Intel Newsroom, Intel Powers First Satellite with AI on Board https://www.intel.com/content/www/us/en/newsroom/news/first-satellite-ai.html
- ESA, First Earth observation satellite with AI ready for launch https://www.esa.int/Applications/Observing_the_Earth/Phsat/First_Earth_observation_satellite_with_AI_ready_for_launch
- KP Labs, Missions https://www.kplabs.space/missions
- Ubotica, Missions https://www.ubotica.com/missions
- Langer and others, Remote Sensing 15(15) 3756, 2023, doi:10.3390/rs15153756

- Experimenting with sensor Webs using Earth Observing 1, IEEE Aerospace Conference, 2004, doi:10.1109/aero.2004.1367603; abstract read in the OpenAlex record https://api.openalex.org/works/W2109821100
- The Autonomous Sciencecraft Experiment Onboard the EO-1 Spacecraft, AAMAS 2004 ( https://openalex.org/W3150443312; ); Intelligent Systems in Space: The EO-1 Autonomous Sciencecraft, AIAA Infotech, 2005, doi:10.2514/6.2005-6917 ( https://openalex.org/W2055287372 ); Flood detection and monitoring with the Autonomous Sciencecraft Experiment onboard EO-1, Remote Sensing of Environment, 2006, doi:10.1016/j.rse.2005.12.018
- Calculated from the example's assumed inputs

## Chapter 10. On-Board AI: Accelerators, Models and Trust

- Giuffrida, Fanucci, Meoni and others, The Phi-Sat-1 Mission: The First On-Board Deep Neural Network Demonstrator for Satellite Earth Observation, IEEE Transactions on Geoscience and Remote Sensing 60 (2022), 5517414, doi:10.1109/TGRS.2021.3125567 https://doi.org/10.1109/TGRS.2021.3125567
- Giuffrida and others (2022), as above
- Intel Newsroom, Intel Powers First Satellite with AI on Board https://www.intel.com/content/www/us/en/newsroom/news/first-satellite-ai.html
- Ekelund, Vinuesa, Khotyaintsev, Henri, Delzanno, Markidis, AI in Space for Scientific Missions: Strategies for Minimizing Neural-Network Model Upload, IEEE e-Science 2024; preprint arXiv:2406.14297 https://arxiv.org/abs/2406.14297
- Center for Security and Emerging Technology (Georgetown University), Onboard AI: Constraints and Limitations (August 2023), doi:10.51593/2022CA008 https://doi.org/10.51593/2022ca008
- TechCrunch (trade press), The largest orbital compute cluster is open for business https://techcrunch.com/2026/04/13/the-largest-orbital-compute-cluster-is-open-for-business/
- Axiom Space, Orbital Data Centers https://axiomspace.com/orbital-data-center
- TechCrunch (trade press), Starcloud raises 170 million Series A https://techcrunch.com/2026/03/30/starcloud-raises-170-million-series-ato-build-data-centers-in-space
- NASA, State-of-the-Art of Small Spacecraft Technology 2026: Integration, Launch and Deployment https://www.nasa.gov/smallsat-institute/sst-soa/integration-launch-and-deployment
- AMD Investor Relations, AMD Announces Completion of Class B Qualification for First Space-Grade Versal Adaptive SoCs https://ir.amd.com/news-events/press-releases/detail/1101/amd-announces-completion-of-class-b-qualification-for-first
- AMD at ESA SEFUW 2025, AMD XQR Versal Adaptive SoCs Enable Next-Generation Signal Processing and AI in Space (2025-03-25) https://indico.esa.int/event/531/contributions/10592/
- Google Research, Exploring a space-based, scalable AI infrastructure system design https://research.google/blog/exploring-a-space-based-scalable-ai-infrastructure-system-design/
- Calculated from the paper's stated values
- Calculated
- Calculated; orbit period from chapter 1
- Calculated as in chapter 1
- Assumed for the illustration, except the 15 W thermal design power (CSET, as above); settled for a real case by the operator's stated figures
- Search of the source library (on-board inference papers, Phi-Sat-1 paper), operator pages and trade press for the systems of table 10.1

## Chapter 11. When Processing in Orbit Is Worth It: The Trade Study

- Turyshev (JPL), Orbital Data Centers: Spacecraft Constraints and Economic Viability, arXiv 2604.27197 (preprint, not peer reviewed) https://arxiv.org/abs/2604.27197
- van Berkel (Eindhoven), The Cost and Network Limits of Space-Based AI Compute, arXiv 2607.14172 (preprint, not peer reviewed) https://arxiv.org/abs/2607.14172
- Hu, Alhassan and Iqbal, Heterogeneous Space Data Centers, arXiv 2610.32260 (preprint, not peer reviewed) https://arxiv.org/abs/2610.32260
- NASA, State of the Art of Small Spacecraft Technology, 7.0 Thermal Control https://www.nasa.gov/smallsat-institute/sst-soa/thermal-control/
- NASA, State of the Art of Small Spacecraft Technology, 3.0 Power https://www.nasa.gov/smallsat-institute/sst-soa/power-subsystems/
- NASA, State of the Art of Small Spacecraft Technology, 10.0 Integration, Launch and Deployment https://www.nasa.gov/smallsat-institute/sst-soa/integration-launch-and-deployment

- NewSpace Economy (trade press), SpaceX rideshare pricing as of February 2026 https://newspaceeconomy.ca/2026/02/27/spacex-rideshare-pricing-as-of-february-2026-what-it-costs-whats-included-and-how-buyers-budget-a-mission/
- calculated from the two rows above
- calculated: T = 2 pi sqrt(a^3/GM), eclipse fraction = asin(Re/a)/pi; GM 398,600.4418 km3/s2 (IERS Conventions 2010), Re 6,378.137 km (WGS 84)
- IAU 2015 Resolution B3, nominal total solar irradiance
- CODATA 2018 recommended values
- calculated by the author from the inputs in Figures 11.1 and 11.2; assumptions: 20 W board, 90 percent distribution and battery efficiency, 85 percent packing and losses, 90 percent end-of-life factor, emissivity 0.85, 50 W per square meter absorbed
- Giuffrida and others, The Phi-Sat-1 Mission: The First On-Board Deep Neural Network Demonstrator for Satellite Earth Observation, IEEE Transactions on Geoscience and Remote Sensing, vol. 60, 2022, article 5517414 https://doi.org/10.1109/tgrs.2021.3125567
- Intel Newsroom, Intel Powers First Satellite with AI on Board https://www.intel.com/content/www/us/en/newsroom/news/first-satellite-ai.html
- ESA, First Earth observation satellite with AI ready for launch https://www.esa.int/Applications/Observing_the_Earth/Phsat/First_Earth_observation_satellite_with_AI_ready_for_launch
- Google Research, Exploring a space-based, scalable AI infrastructure system design (2025-11-04) https://research.google/blog/exploring-a-space-based-scalable-ai-infrastructure-system-design/
- calculated from illustrative assumptions stated in the sheet
- The Bit's Path (first book of this series), chapters 4 and 8
- Calculated from the chapter's stated inputs

## Chapter 12. Computing Platforms in Orbit: Launched, Demonstrated, Offered, Announced

- Axiom Space, Orbital Data Centers https://axiomspace.com/orbital-data-center
- TechCrunch (trade press), The largest orbital compute cluster is open for business https://techcrunch.com/2026/04/13/the-largest-orbital-compute-cluster-is-open-for-business/
- Hu, Alhassan and Iqbal, Heterogeneous Space Data Centers: A Key Paradigm for Future Non-Terrestrial and In-Orbit Services, arXiv 2610.32260 (preprint, not peer reviewed) https://arxiv.org/abs/2610.32260
- ESA, NanoSat MO Framework documentation, OPS-SAT https://github.com/esa/nanosat-mo-framework/blob/HEAD/docs/source/specific-missions/ops-sat.rst
- ESA, OPS-SAT Space Lab https://opssat.esa.int/
- Spire, Space Services https://spire.com/space-services/
- Loft Orbital, home page https://loftorbital.com/
- TechCrunch (trade press), Starcloud raises 170 million Series A https://techcrunch.com/2026/03/30/starcloud-raises-170-million-series-ato-build-data-centers-in-space
- Google Research, Exploring a space-based, scalable AI infrastructure system design https://research.google/blog/exploring-a-space-based-scalable-ai-infrastructure-system-design/
- Space.com (trade press), Watch SpaceX launch Google AI satellite, 129 other payloads https://www.space.com/space-exploration/satellites/spacex-google-project-suncatcher-ai-satellite-transporter-18-mission
- Google, The Keyword blog, listing of the post "We're partnering with Blue Origin on a Project Suncatcher launch" https://blog.google/
- van Berkel, The Cost and Network Limits of Space-Based AI Compute, arXiv 2607.14172 (preprint, not peer reviewed) https://arxiv.org/abs/2607.14172
- Turyshev, Orbital Data Centers: Spacecraft Constraints and Economic Viability, arXiv 2604.27197 (preprint) https://arxiv.org/abs/2604.27197
- Rent Before You Build, ledger LL-2026-18, sources as listed on its page 1
- M. Sun, Z. Chen, J. Hou, K. Wang and others, "Toward Communication-Efficient Space Data Centers: Bottlenecks, Architectures, and New Paradigms", arXiv 2605.12681 (preprint, not peer reviewed) https://arxiv.org/abs/2605.12681

## Epilogue: Reading the Next Announcement


- NASA, Computers in Spaceflight: The NASA Experience, chapter 2, Box 2-1 https://www.nasa.gov/wp-content/uploads/static/history/computers/Ch2-5.html
- Axiom Space, Orbital Data Centers https://axiomspace.com/orbital-data-center


## Appendix B. Timeline of Flight Computers by National Line, 1962 to 2026, with Cases Sought


- NASA, Computers in Spaceflight: The NASA Experience (1988), chapters 1, 2, 4, 5 and 6 https://www.nasa.gov/wp-content/uploads/static/history/computers/contents.html
- NASA, Apollo Lunar Surface Journal, Apollo 11 landing https://www.nasa.gov/wp-content/uploads/static/history/alsj/a11/a11.landing.html
- NASA History, 40 Years Ago: The Countdown Begins for STS-1; First Launch Attempt Scrubbed https://www.nasa.gov/history/40-years-ago-the-countdown-begins-for-sts-1-first-launch-attempt-scrubbed/
- D. McComas, NASA Goddard Space Flight Center, Lessons from 30 Years of Flight Software, presentation, 22 September 2015
- BAE Systems, Radiation-hardened electronics https://www.baesystems.com/en-us/product/radiation-hardened-electronics
- NASA JPL contract release; NASA HPSC page; Microchip release republished by Finviz (trade press), as in the claim table of chapter 5
- Yuhaniz, University of Surrey doctoral thesis (2008); RTEMS ERC32 board support README
- McComas (2015); Frontgrade Gaisler, GR740; RTEMS project home page
- ISRO, VIKRAM3201 https://www.isro.gov.in/vikram3201.html
- Kuwahara et al., Trans. JSASS 19 (2021); Kobayashi et al., Trans. JSASS 12 (2014)


## Appendix C. Glossary


- J. Pellish, NASA GSFC, Tutorial: Radiation Effects in Electronic Systems, ASEC, May 2017 https://ntrs.nasa.gov/api/citations/20170004674/downloads/20170004674.pdf
- Pellish 2017 (as above)
- A. Kazak and others, Electronics 15 (2026) 2814 https://doi.org/10.3390/electronics15132814
- ESA, "Radiation: satellites' unseen enemy", 4 July 2011 https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Radiation_satellites_unseen_enemy
- M. Berg and K. LaBel, NASA GSFC, SEE Symposium and MAPLD 2016 https://ntrs.nasa.gov/citations/20160007971
- NanoXplore and STMicroelectronics release, 28 January 2026, as in chapter 6


## Appendix D. Dated Register: Parts, Qualification and Computing in Orbit as of November 2026


- Frontgrade Gaisler, GR740 https://www.gaisler.com/products/gr740
- RTEMS project home page https://www.rtems.org/
- BAE Systems, Radiation-hardened electronics https://www.baesystems.com/en-us/product/radiation-hardened-electronics
- NASA JPL, NASA Awards Next-Generation Spaceflight Computing Processor Contract https://www.jpl.nasa.gov/news/nasa-awards-next-generation-spaceflight-computing-processor-contract/
- NASA, High Performance Spaceflight Computing (HPSC) https://www.nasa.gov/high-performance-spaceflight-computing-hpsc/
- Microchip release via GlobeNewswire, republished by Finviz https://finviz.com/news/408662/microchip-achieves-key-program-milestones-for-its-pic64-hpsc-spaceflight-computing-series
- ISRO, VIKRAM3201 https://www.isro.gov.in/vikram3201.html
- AMD, AMD XQR Versal Adaptive SoCs Enable Next-Generation Signal Processing and AI in Space, ESA SEFUW, 2025-03-25 https://indico.esa.int/event/531/contributions/10592/
- AMD Investor Relations https://ir.amd.com/news-events/press-releases/detail/1101/amd-announces-completion-of-class-b-qualification-for-first

- Microchip release via GlobeNewswire, republished by Finviz https://finviz.com/news/425099/microchip-expands-rt-polarfire-fpga-portfolio-with-new-spaceflight-qualifications
- NanoXplore and STMicroelectronics release, republished https://www.edge-ai-vision.com/2026/01/nanoxplore-and-stmicroelectronics-deliver-european-fpga-for-space-missions/
- Frontgrade, UT24CP1008 https://www.frontgrade.com/product/ut24cp1008
- Frontgrade Gaisler, NOEL-V https://www.gaisler.com/products/noel-v
- AAC Clyde Space, Smallsat Sirius OBC https://www.aac-clyde.space/what-we-do/space-products-components/command-data-handling/smallsat-sirius-obc
- satsearch, Xiphos Q7 Processor https://satsearch.co/products/xiphos-q7-processor
- Giuffrida and others, IEEE TGRS 60, 2022, doi:10.1109/tgrs.2021.3125567
- TechCrunch https://techcrunch.com/2026/04/13/the-largest-orbital-compute-cluster-is-open-for-business/
- TechCrunch, 2026-03-30 https://techcrunch.com/2026/03/30/starcloud-raises-170-million-series-ato-build-data-centers-in-space
- Google Research https://research.google/blog/exploring-a-space-based-scalable-ai-infrastructure-system-design/
- KP Labs, Missions https://www.kplabs.space/missions
- Ubotica, Missions https://www.ubotica.com/missions
- NASA, State-of-the-Art of Small Spacecraft Technology 2026: Integration, Launch and Deployment https://www.nasa.gov/smallsat-institute/sst-soa/integration-launch-and-deployment
- Axiom Space, Orbital Data Centers https://www.axiomspace.com/orbital-data-center
- Spire, Space Services https://spire.com/space-services/
- Loft Orbital https://loftorbital.com/
- ESA, NanoSat MO Framework documentation https://github.com/esa/nanosat-mo-framework/blob/HEAD/docs/source/specific-missions/ops-sat.rst
- ESA, OPS-SAT Space Lab https://opssat.esa.int/
- Hu, Alhassan and Iqbal, arXiv 2610.32260 (preprint, not peer reviewed) https://arxiv.org/abs/2610.32260
- Space.com https://www.space.com/space-exploration/satellites/spacex-google-project-suncatcher-ai-satellite-transporter-18-mission
- Google, The Keyword https://blog.google/
- van Berkel, arXiv 2607.14172 (preprint, not peer reviewed) https://arxiv.org/abs/2607.14172
- As in the claim table of chapter 8
- M. Sun, Z. Chen, J. Hou, K. Wang and others, "Toward Communication-Efficient Space Data Centers: Bottlenecks, Architectures, and New Paradigms", arXiv 2605.12681 (preprint, not peer reviewed) https://arxiv.org/abs/2605.12681

THE SERIES

# Every Bit Lands

Every Bit Lands is a series about satellites, end to end: how each system divides the work between the user's end, the satellite and the ground, whether each hop closes, and what one engineer can use, rent and build. This is its seventh book. Each book stands on its own.

Also by the author:

### OTHER SERIES

**From Terminal to Agent**
Every Boundary Holds

**From Code to Glass**
Every Pixel Matters

**From Tally to Token**
Every Record Counts

**From Warehouse to Lakehouse**
Every Record Counts

**Data on Every Desk**
Every Record Counts

**Governing Rust by Contract**
GDCD in Code

**AI Foundry by Contract**
GDCD on Azure

**From Bit to Pixel**
Every Pixel Matters

**From Silicon to Screen**
Every Pixel Matters

**The Model as Reader**
Every Record Counts

**What a Value Means**
Every Record Counts

**The Contract Decides**
GDCD Field Guides

**Governing Azure by Contract**
GDCD on Azure

**From Bronze to Gold**
The Medallion Series

### FIELD GUIDES

**Ten Engineers, the Output of Thirty**
Field guide

**The Fifth Mind**
Field guide

**Every Pixel Matters**
Field guide

**The Forge and The Universe**
Field guide

**Who Is the Agent?**
Field guide

**The Agent Loop**
Field guide

**The Mesh**
Field guide

**The Crystal and The Ocean**
Field guide

**The Story of Azure**
Field guide

**The Security You Can Watch**
Field guide

**Juan Cardena**

Enterprise AI & Agentic Systems Architect · blog.jcardena.com

Download the latest edition and the blank Link Ledger and the budget sheet at blog.jcardena.com/en/ebooks/computing-in-orbit.

Every Bit Lands · Book 7 · Computing in Orbit.