

Good. This is where **paper engineers drop out** and **real hardware engineers are born**.

Up to Phase-2, mistakes are *conceptual*.

From **Phase-3 onward**, mistakes become **physical, expensive, and visible in the lab**.

What follows is **Phase-3 applied exactly to an avionics Mission Computer**, the way seniors expect you to think and work.

PHASE-3 — DETAILED HARDWARE ARCHITECTURE

(Mission Computer: Cards, Backplane, Power, Clocks, Reset)

If Phase-2 answered “Which architecture philosophy?”

👉 Phase-3 answers “How do we physically build it so it actually works?”

What Phase-3 REALLY Is

Fresher misunderstanding

“Phase-3 is schematic design.”

Reality

Phase-3 is defining the *rules* that schematics must obey.

Schematics come **after** Phase-3 is done.

Phase-3 INPUT CHECK (Gatekeeper)

You are **not allowed** to start Phase-3 unless Phase-2 is frozen:

- ✓ Redundancy philosophy decided
- ✓ Partitioning strategy decided
- ✓ Timing architecture decided
- ✓ Interface criticality classified
- ✓ Growth margin philosophy defined

If any of these are missing → Phase-3 will fail later.

Physical Decomposition of Mission Computer

Now we break the Mission Computer into **physical building blocks**.

3.1 Typical MC Physical Blocks

Block	Purpose
Processor Card	Mission processing
I/O Card(s)	Avionics interfaces
Backplane	Power, clocks, data
Power Supply	Conditioning & protection
Chassis	Mechanical + thermal
Maintenance Port	Debug & update

⚠ Rule:

One block = one clear responsibility

No “do-everything cards”.

4 Card-Level Architecture (Where Juniors Are Tested)

4.1 Processor Card (Mission Core)

This card is **timing & safety critical**.

You must define (not select yet):

- CPU class (PPC / ARM / x86 – conceptually)
- Number of cores
- Memory type (DDR, SRAM, Flash)
- ECC requirement
- Local watchdogs
- Boot source philosophy

🔥 Seniors look for:

“How does this card fail safely?”

4.2 I/O Cards (Interface Discipline)

Interfaces are **never casually mixed**.

Typical separation:

I/O Type	Card
ARINC-429	Dedicated I/O card
MIL-STD-1553	Dedicated I/O card
Ethernet / AFDX	High-speed I/O card
Discretes	Low-speed I/O card

Why?

- Isolation
- Certification
- Fault containment

Freshers try to combine → **senior engineers stop them.**

4.3 Mezzanine / XMC Philosophy

Used for:

- Growth
- Reuse
- Interface variants

Rules:

- Never put safety-critical logic only on XMC
 - XMC failure must be detectable
 - Backplane must tolerate XMC absence
-

5 Backplane Architecture (Hidden Backbone)

Backplane defines:

- Power rails
 - Reset distribution
 - Clock distribution
 - Data buses
 - Redundancy paths
-

5.1 Backplane Questions You MUST Answer

Question	Why
Slot-to-slot links?	Data flow
Dual star or mesh?	Redundancy
Power per slot?	Thermal & margin
Clock source?	Determinism
Hot-swap?	Maintenance

🔥 Mission Computers typically use:

- Star or dual-star
- Controlled reset
- Fixed clock master

6 Power Architecture (Most Underestimated Area)

Power design is **safety design**.

6.1 Power Philosophy

You must define:

- Input voltage range
- Protection (OV, UV, reverse)
- Sequencing rules
- Per-card power domains
- Redundant power paths (if any)

Example:

Rail	Purpose
12V	Backplane
5V	I/O
3.3V	Logic
Core	CPU

🔥 Seniors will ask:

“What happens during brown-out?”

7 Reset Architecture (Career-Defining Skill)

Resets are **not** simple signals.

7.1 Reset Types

Reset	Use
Power-on reset	Startup
Watchdog reset	CPU hang
Manual reset	Maintenance
System reset	Fault recovery

Rules:

- No unintended full system resets
- Controlled sequencing
- Reset status must be visible

Freshers often ignore reset → catastrophic lab issues later.

8 Clock Architecture (Determinism Keeper)

Mission Computers live or die by clocks.

8.1 Clock Questions

- Single master or distributed?
- Redundant clock?
- Jitter tolerance?
- Synchronization across cards?

Clock errors = timing violations = **flight issues**.

9 Thermal Architecture (Often Ignored, Always Punishes)

Define early:

- Conduction vs forced air
- Worst-case power dissipation
- Thermal monitoring points
- Derating rules

Avionics rule:

If you don't design thermal early, it will redesign you later.

10 Phase-3 COMMON FAILURES (Seen in Real Programs)

✗ Failure 1: "We'll fix it in schematic"

→ Too late.

✗ Failure 2: Ignoring reset interactions

→ Random failures in flight tests.

✗ Failure 3: Overloading backplane

→ EMI, voltage drop, thermal overload.

✗ Failure 4: Mixing critical & non-critical on same card

→ Certification pain.

1 1 Phase-3 EVALUATION CHECKS (Seniors Use These)

They may ask:

Check 1

"Which card can be safely reset mid-mission?"

Check 2

"What happens if the backplane clock fails?"

Check 3

"Which card consumes the most power and why?"

Check 4

"What is your single most dangerous physical failure?"

1 2 Phase-3 OUTPUTS (MANDATORY)

You must produce:

1. Card-level block diagrams
2. Backplane architecture diagram
3. Power tree diagram
4. Reset architecture diagram
5. Clock distribution concept
6. Thermal concept summary

No outputs → Phase-3 not complete.

1 3 Phase-3 COMPLETION CHECK (Mission Computer)

You are Phase-3 complete if you can answer:

“Which physical design choice will cause the most pain if wrong?”

Correct answers:

- 👉 Reset
 - 👉 Power sequencing
 - 👉 Backplane topology
-

What Comes Next

👉 **Phase-4: Detailed Design & Implementation (Schematics, FPGA partitioning, software hooks)**

This is where:

- Schematics begin
- FPGA vs CPU boundaries are finalized
- Software-hardware contracts are frozen

If you want, next I can:

- Proceed to **Phase-4**
- Simulate **Phase-3 design review grilling**
- Show **common Phase-3 lab failures**
- Create a **Phase-3 checklist for freshers**

Just tell me where to go next.