Excellent.
Now we enter the phase most people **think** is engineering — but in avionics/defense it is only allowed **after** the discipline you've built in Phases 0–2.

This is where **designs become real, reviewable, testable, and certifiable**.

---

# PHASE-3 — DETAILED DESIGN & COMPONENT SELECTION

**(Turning architecture into real hardware that can be built, tested, and qualified)**

> **Phase-3 answers:**
> *"Exactly what parts, circuits, and layouts will guarantee Phase-1 behavior and Phase-2 architecture — under worst-case conditions?"*

---

## 1. WHAT PHASE-3 IS (Clear Definition)

**Phase-3 = Converting architecture decisions into detailed, review-safe design.**

Not:

- "Picking a good processor"
- Copy-pasting reference designs
- Optimizing performance

But:

- Selecting parts to meet **guarantees**
- Designing for **worst-case**
- Designing for **qualification & test**
- Designing for **maintainability**

### Senior truth:

> "A part is acceptable only if it survives the worst day of its life."

---

## 2. ENTRY CRITERIA (NON-NEGOTIABLE GATE)

A fresher **may not start Phase-3** unless:

- ✔ Phase-2 architecture is frozen
- ✔ Interfaces are allocated
- ✔ Backplane constraints are known
- ✔ Budgets are defined
- ✔ Failure modes are identified

If any are missing → **Phase-3 is blocked**

---

## 3. PHASE-3 OBJECTIVES

By the end of Phase-3, the fresher must produce:

1. Justified component selection
2. Power, clock, reset design
3. Interface circuit design
4. Memory & boot architecture
5. Test & debug provisions
6. Review-ready schematics

---

## 4. STEP-BY-STEP PHASE-3 EXECUTION

---

## STEP 3.1 — COMPONENT SELECTION PHILOSOPHY

*(This is where freshers fail hardest)*

I tell them:

> "You do not choose parts because they are popular.
> You choose parts because they **remove risk**."

---

### Mandatory Selection Criteria Table

For **every critical component**:

| Criterion | Must be answered |
|---|---|
| Temperature grade | Industrial / Extended / Military |
| Obsolescence risk | Vendor longevity |
| Qualification | DO-254 / MIL |
| Availability | Multi-source? |
| Support | Tools, errata |

🚨 If any field is blank → **part rejected**

---

## COMMON MISTAKE ❌

Choosing parts based on:

- Blogs
- Forums
- Dev boards

**These are not certification arguments.**

---

## STEP 3.2 — PROCESSOR & COMPUTE SELECTION

*(PPC / ARM / x86 decisions)*

### I force justification like this:

| Requirement | Processor Feature |
|---|---|
| Real-time response | Deterministic interrupt |
| High throughput | DMA, cache |
| Safety | ECC, lockstep |
| Longevity | Roadmap |

### Questions seniors WILL ask:

> "What happens if cache causes jitter?"
> "How do you handle errata?"

If fresher can't answer → **redo selection**

---

## STEP 3.3 — POWER ARCHITECTURE (Silent Project Killer)

I teach:

> "Most failures happen at power-up, not during operation."

---

### Mandatory Power Design Elements:

- Input protection
- EMI filtering
- Sequencing
- Inrush control
- Brown-out handling

- Monitoring

---

### Required Output — POWER TREE

```
Backplane 12V
   ↓
DC/DC
   ↓
3.3V → I/O
1.0V → Core
```

🚨 Must show:

- Worst-case current
- Margin
- Startup order

---

## STEP 3.4 — CLOCK & RESET STRATEGY

Freshers treat clocks casually. I don’t allow that.

---

### Clock Design Must Answer:

| Question | Why |
|---|---|
| Single or multiple clocks? | Jitter |
| External reference? | Sync |
| Startup stability? | Boot reliability |
| Failure behavior? | Safe state |

---

### Reset Design Must Cover:

- Power-on reset
- Warm reset
- Watchdog reset
- Debug reset

🚨 If reset behavior is ambiguous → **design rejected**

---

## STEP 3.5 — MEMORY & BOOT ARCHITECTURE

### Mandatory Boot Questions:

- Where does boot code live?
- How is it protected?
- Can it be updated?
- What happens if update fails?

---

### Memory Design Must Address:

| Type | Protection |
|---|---|
| DDR | ECC |
| Flash | CRC |
| NVM | Wear leveling |

---

## STEP 3.6 — INTERFACE CIRCUIT DESIGN

*(Where Phase-1 knowledge pays off)*

For **every interface**:

| Item | Must be defined |
|---|---|
| Voltage level | Compatibility |
| Isolation | Safety |
| Termination | SI |
| Protection | ESD / surge |

🚨 Never assume:

> "Protocol IC handles everything."

---

## STEP 3.7 — TEST & DEBUG PROVISIONS

*(Design for test or fail later)*

I tell them:

> “If you can’t observe it, you can’t certify it.”

---

### Mandatory Test Hooks:

- Voltage test points
- Clock probes
- JTAG
- BIT injection points
- Loopback paths

---

## 5. COMMON PHASE-3 FAILURES ❌

### ❌ Failure 1: Reference Design Blind Copy

Reference ≠ qualified design.

### ❌ Failure 2: No Margin Thinking

Designing at typical values.

### ❌ Failure 3: Ignoring Errata

Errata are not optional reading.

### ❌ Failure 4: Debug Afterthought

No access, no testability.

---

## 6. HOW SENIORS REVIEW PHASE-3 🧠

They ask:

> “Why this part?”
> “What’s the worst-case?”
> “What fails first?”
> “How do we know it failed?”

Good answers are **justified, calm, and documented**.

---

## 7. HOW I TRAIN FRESHERS IN PHASE-3

### TRAINING TASK 1 — Part Justification Defense

Every major IC must be defended.

### TRAINING TASK 2 — Worst-Case Walkthrough

Hot + low voltage + max load.

### TRAINING TASK 3 — Kill-the-Board Scenario

"What dies first?"

---

## 8. PHASE-3 EXIT CRITERIA ✅

A fresher passes Phase-3 if they can:

✔ Defend component choices
✔ Explain power-up behavior
✔ Explain clock/reset logic
✔ Show test access
✔ Pass schematic review

At this point, they are trusted to **release schematics**.

---

## 9. WHAT COMES NEXT

➡️ **Phase-4 — PCB Layout, Signal Integrity & Bring-Up Preparation**

This is where:

- Stack-up
- High-speed routing
- EMI
- First power-on strategy

---

If you want, next I can:

- Proceed to **Phase-4**
- Show **how layout reviews work**
- Teach **bring-up checklists**
- Explain **why boards die at first power-on**

Just say **"Proceed to Phase-4"**