



JITX · JUMPSTART KIT #1 · PART 1

Build a 20-layer HDI substrate from the fab's stackup report

ACME quote Q26-0417 Rev B · 20 layers · 5-10-5 HDI · 0.80 mm core · four impedance targets

Drive jitx-substrate-modeler to turn the fab's CSV into substrate code — then verify every number back against the report.



The fab report is the ground truth — treat it that way

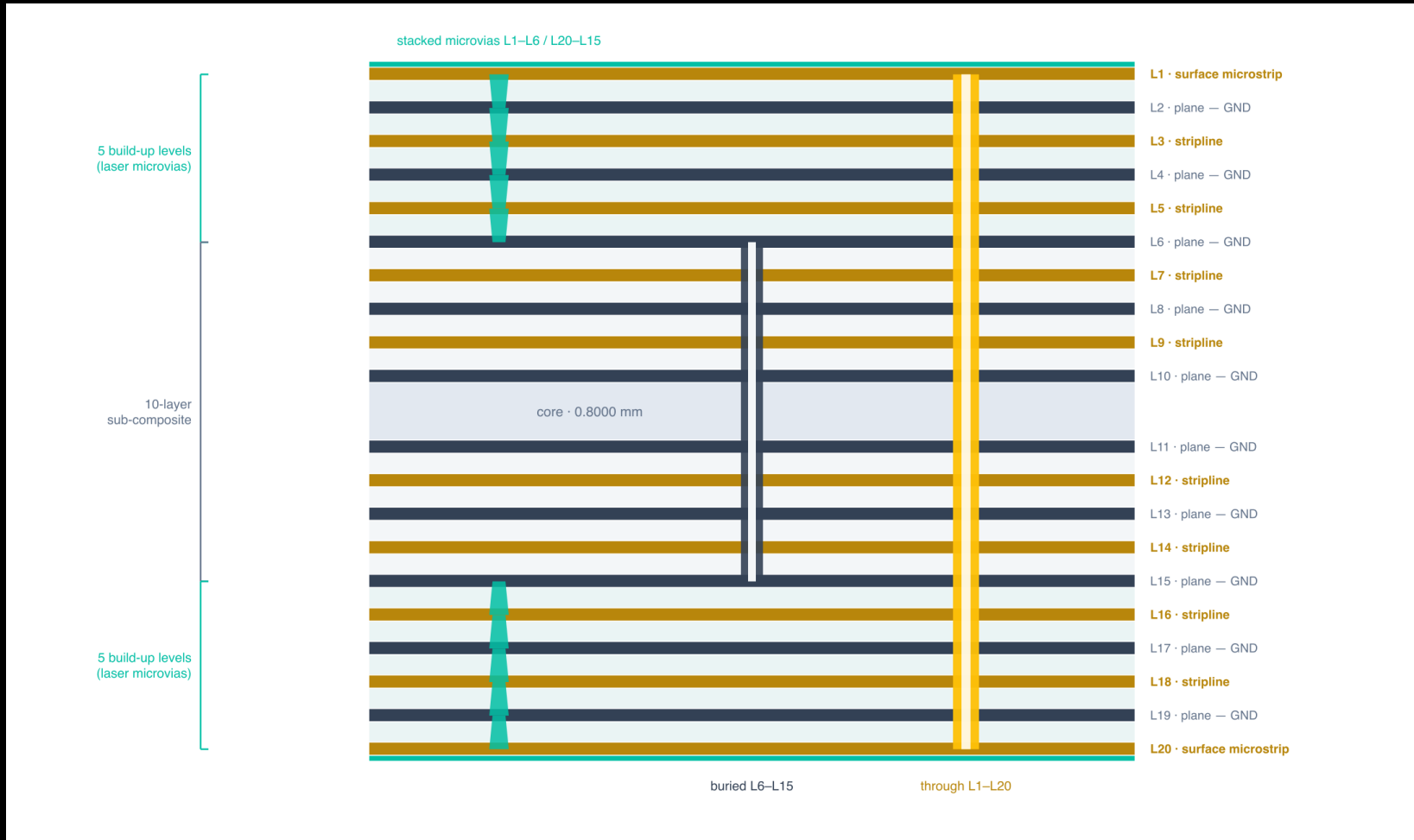
Part 1 teaches: the fab's report becomes code. Your fab quotes a stackup as a spreadsheet — materials, thicknesses, vias, impedance tables. That file, not a template and not a guess, is the only source of truth.

Estimate nothing. Every material, thickness, via and trace width in the substrate must trace to one CSV row. If the CSV doesn't say it, the agent stops and asks you.

Model what the report means. One material class per CSV row and an explicit 41-entry stackup — not a mirrored shortcut that would hide the report's asymmetries.

Then verify the translation. The substrate isn't done when it builds — it's done when the code provably matches the report, row by row.

One CSV, 20 copper layers, five microvia levels per side



Drawn from ACME Q26-0417 Rev B (thicknesses schematic) · microstrip on L1/L20 · controlled stripline on L3/L5/L16/L18

What you build

Materials — one Dielectric or Conductor class per CSV material row — soldermask, build-up prepreg, bonding prepreg, the core laminate, three copper foils — with Dk/Df and roughness exactly as the report states them.

The stackup — an explicit 41-entry Stackup, top mask to bottom mask, every copper layer function-named. No Symmetric: the report's asymmetries must survive translation.

Fabrication rules — all 19 FabricationConstraints fields from the FAB_RULES section — trace floors, annular rings, drill and edge clearances.

Vias — the report's full inventory: ten single-level laser microvias (L1–L2 ... L5–L6 and their mirrors), the buried L6–L15 sub-composite via, the L1–L20 through-hole.

Routing structures — one per impedance target — 40 / 50 / 55 Ω single-ended and 100 Ω differential — each solved on both the microstrip and stripline layer groups.

Verification is the key

The CSV is re-read, not remembered. The test suite parses the report and compares the translated design against it — materials, thicknesses, via geometry, fab rules, reference planes and impedance targets.

Summed thickness matches the quote. The stack's total equals the report's finished thickness — the classic transcription-slip catch.

You check the layer map. A [HUMAN] gate walks the conductor-index map (0 = L1, -1 = L20) against the report before any rules land on it.

Gates at every step. pyright → pytest → jitx build → jitx-code-review — green before anything is committed.

Re-issue tolerance. Change the CSV and re-derive — the code carries no number the report didn't state, so a Rev C flows straight through.

Run it

The runbook: `jumpstart-kits/js1-stackup-components/part1-stackup/` — paste `JS1_Part1_Stackup-from-Fab-CSV_Runbook.md` into your agent with the CSV alongside, and tell it to follow the file. Two [HUMAN] gates stop for you.

The ground truth: `part1-stackup/JS1_Part1_Fab-Stackup_RevB.csv` — a synthetic fab quote that parses with Python's `csv` module. Substitute your own fab's export.

The reference solution: `jitxexamples.jumpstart_kits.js1_stackup_components.hdi_stackup` — materials, stackup, rules, vias, routing structures, and the buildable design.



CLOSE THE LOOP

“the fab came back on the through-hole again — 0.400 mm finished hole this time”

Hand the agent a Rev C and watch the change flow through code, tests, and build.