Detailed Explanation – EDT Flow (Level2)

This document explains the step-by-step flow followed in Level2 (EDT insertion).

1. Dofile Script (atpg_dofile.png)

- Context set: `set_context dft -edt`
- Loaded scan-inserted netlist + standard cell library
- Ran ATPG/EDT configuration commands
- Purpose: prepare Tessent for EDT compression

2. Log File (edt_log.png)

- 40 scan cells traced into scan chains
- EDT wrapper added (decompressor + compactor)
- Warnings observed:
  - X-state propagation handling (E5-1)
  - Scan init patterns increased by 25%
- Outputs: EDT netlist, reports, wrapper

3. Synthesis Log (syn_log.png)

- Reports generated: `report_area`, `report_timing`, `report_constraint`
- Area overhead: ~1361 units
- Timing: No violations (slack = 0)

4. EDT Top Netlist (case1_edt_top_gate.png)

- Gate-level Verilog netlist after EDT
- Scan flops with SI (scan in), SO (scan out), SE (scan enable)
- Example:
  ```verilog
  SDFFX4 DmaSel (.Q(DMSel_r), .D(n290), .SI(net253576), .SE(scan_en), .CK(FastClk), .SN(Reset));

**5. Tessent Core Description (case1_DmaWr_edt_tcd.png)** Defines EDT decompressor + compactor Interfaces:

- edt_channels_in / edt_channels_out
- edt_scan_in / edt_scan_out
- Used later in ATPG flow

6. **Terminal Log (terminal_log.png)**

- Execution trace across directories: atpg/, edt/, edt_pipeline/
- Shows dofile runs and generation of outputs
- Confirms EDT insertion completed successfully

**Summary**

- Scan chains compressed using EDT wrapper
- All reports, logs, netlists generated cleanly
- Design now ready for ATPG pattern generation (next step)
  ```