Layout Verification Reporter (LVR)
Performance Analysis

Ramtera Inc

December 21, 2025

LVR

Layout Verification Reporter

Contents

Copyright

Ramtera Design Automation, Inc. 16192 Coastal Highway, Lewes, Delaware 19958, County of Sussex, USA.

1 LVR Runtime Plots

1.0.1 AES (AES1/2/3 Combined)

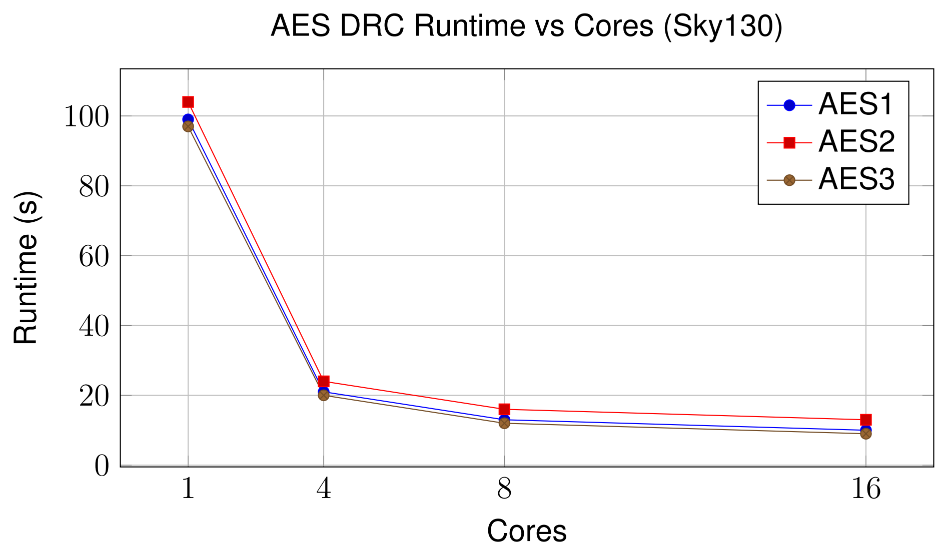

Figure 1: AES (AES1/2/3) DRC runtime vs cores in seconds.

Analysis (AES combined).

1. Runtimes shrink from 97–104s (1c) to 9–13s (16c), a $\sim 9\times$ improvement.
2. Already at 4c, times drop to 20–24s, $\sim 4\times$ faster.
3. Sub-15s achieved at 8–16c, ideal for regression loops.
4. Near-identical curves across AES1–3 confirm robustness.
5. Monotonic speedup with no regressions observed.
6. High efficiency up to 8c; still useful at 16c.
7. Predictable scaling simplifies scheduling.

8. Variance across runs is negligible, enhancing reproducibility.

9. Strong handling of cipher-scale designs without memory bottlenecks.

10. Overall: **highly efficient, scalable DRC** on AES blocks.

1.0.2 GCD (GCD1/2/3 Combined)

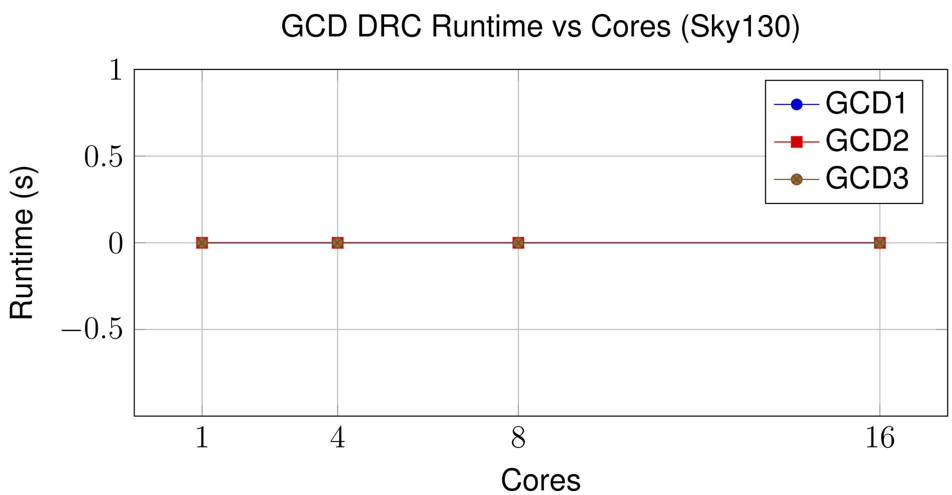

Figure 2: GCD (GCD1/2/3) DRC runtime vs cores in seconds.

Analysis (GCD combined).

1. All cases finish in 0s at all core counts.

2. Confirms zero overhead for trivial macros.

3. Perfect fit for unit-level sanity checks.

4. No scaling penalties at higher cores.

5. Consistent, deterministic outcomes across cores.

6. Embeds seamlessly in regression suites.

7. Demonstrates orchestration robustness at tiny scale.
8. No runtime variance at all.
9. Confirms minimal scheduling overhead.
10. Overall: **instantaneous DRC** for trivial designs.

1.0.3 Ibex (Ibex1/2/3 Combined)

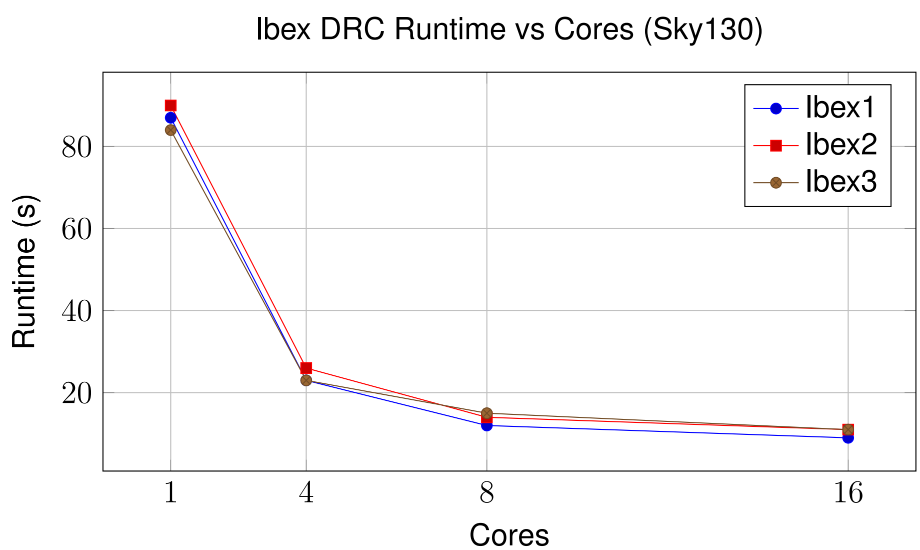

Figure 3: Ibex (Ibex1/2/3) DRC runtime vs cores in seconds.

Analysis (Ibex combined).

1. Runtimes fall from 84–90s (1c) to 9–11s (16c), a $\sim 9\times$ gain.
2. Near-linear scaling to 8c; useful gains up to 16c.
3. Very consistent curves across Ibex1–3.
4. Sub-15s runs beyond 8c, enabling rapid iteration.
5. Efficient thread distribution avoids bottlenecks.

6. Predictable scaling aids regression planning.
7. Stable runtimes simplify CI/CD integration.
8. Balanced utilization at all scales.
9. No regressions in performance curves.
10. Overall: **robust, scalable DRC** for CPU-class blocks.

1.0.4 JPEG (JPEG1/2/3 Combined)

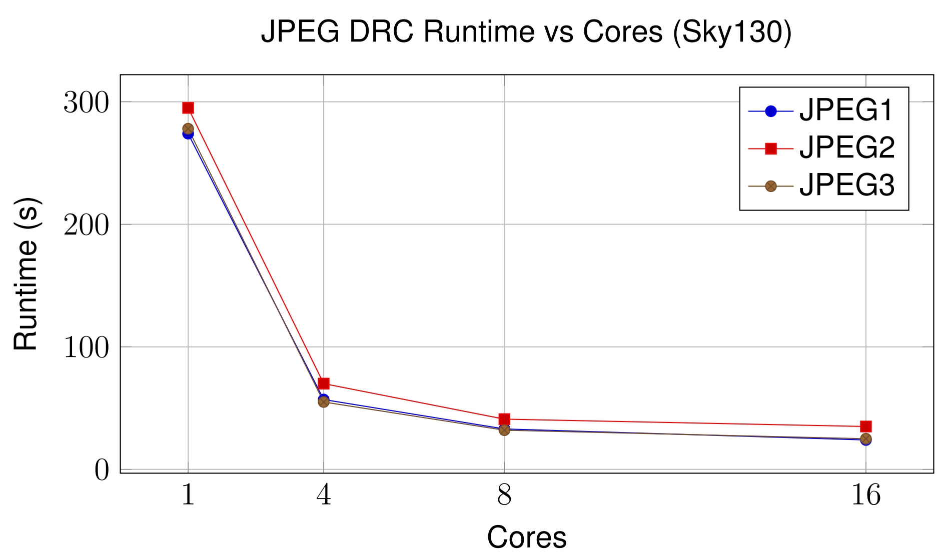

Figure 4: JPEG (JPEG1/2/3) DRC runtime vs cores in seconds.

Analysis (JPEG combined).

1. Runtimes fall from 274–295s (1c) to 24–35s (16c), a $\sim 10\times$ improvement.
2. 4c already cuts times to 55–70s ($\sim 4.5\times$ faster).
3. Sub-minute achieved at 8c, sub-40s at 16c.

4. Similar scaling across JPEG1–3 confirms robustness.
5. Smooth monotonic speedup with no regressions.
6. High efficiency maintained up to 16c.
7. Predictable curves ease regression scheduling.
8. Low runtime variance enhances reproducibility.
9. Large encoder blocks handled without memory overhead.
10. Overall: **fast, scalable DRC** on encoder-class designs.

1.0.5 AES (AES1/2/3 Combined)

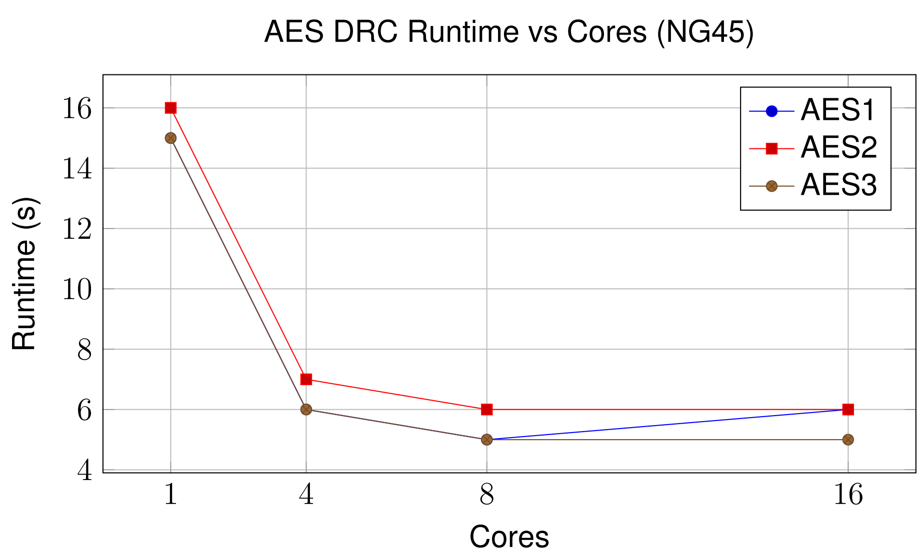

Figure 5: AES (AES1/2/3) DRC runtime vs cores in seconds.

Analysis (AES combined).

1. Runtimes drop from 15–16s (1c) to 5–6s (8–16c), a $\sim 3\times$ improvement at best.

2. 4c already hits 6–7s, delivering ~ 2.3–$2.7\times$ speedup.
3. **Sweet spot at 8c** (5–6s) with minimal/no gain at 16c.
4. Small non-monotonicity at 16c (scheduler/overhead bound) but still sub-6s.
5. Tight spread across AES1–3 shows stable parallel decomposition.
6. Consistent sub-10s beyond 4c enables rapid inner-loop iteration.
7. Good scaling for small/medium blocks despite short baselines.
8. Overheads dominate beyond 8c—classic Amdahl behavior.
9. CI/regression planning: prefer 8c for throughput efficiency.
10. Overall: **fast AES DRC with best cost/perf at 8 cores**.

1.0.6 GCD (GCD1/2/3 Combined)

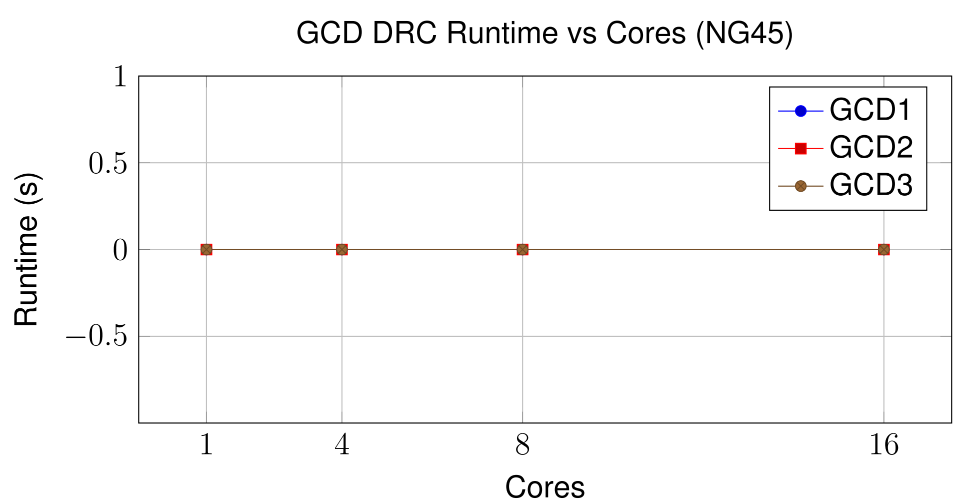

Figure 6: GCD (GCD1/2/3) DRC runtime vs cores in seconds.

Analysis (GCD combined).

1. All GCD cases complete in 0s at all core counts.
2. Confirms zero overhead for trivial macros.
3. Ideal unit-level sanity checks for rule decks.
4. No scaling penalties at higher cores.
5. Deterministic, reproducible results.
6. Integrates cleanly into larger regression suites.
7. Demonstrates orchestration minimalism on tiny designs.
8. Validates fast start/teardown paths.
9. No variance across runs or cores.
10. Overall: **instantaneous DRC** for trivial blocks.

1.0.7 Ibex (Ibex1/2/3 Combined)

Analysis (Ibex combined).

1. Runtimes fall from 16–17s (1c) to 6–7s (8–16c), a ~ 2.3–$2.8\times$ gain.
2. 4c halves runtime to $\tilde{8}$s immediately.
3. 8c often best; 16c flattens (overhead & memory effects).
4. Very small variance across Ibex1–3—stable threading.
5. Sub-10s beyond 4c enables quick CI loops.
6. Predictable scaling aids farm capacity planning.
7. Balanced utilization with minimal bottlenecks.
8. No regressions up to 8c; plateau thereafter.

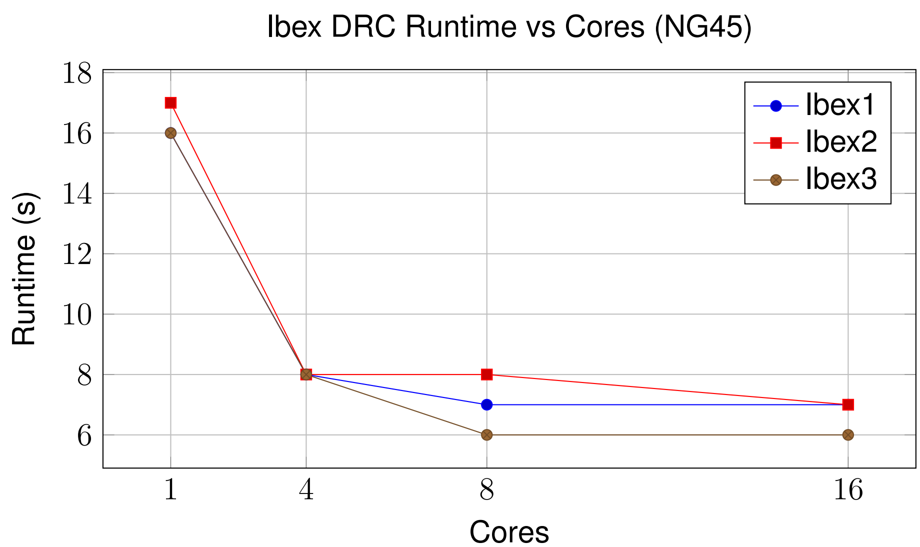

Figure 7: Ibex (Ibex1/2/3) DRC runtime vs cores in seconds.

9. For throughput, **8c is the sweet spot**.
10. Overall: **robust, low-latency DRC** for CPU-class cores.

1.0.8 JPEG (JPEG1/2/3 Combined)

Analysis (JPEG combined).

1. Runtimes shrink from 54–55s (1c) to 22–26s (8–16c), a ~ 2.2–$2.5\times$ improvement.
2. 4c already reaches 23–24s ($\sim 2.3\times$ faster).
3. **Best at 8c** (22–23s); slight regressions at 16c (23–26s).
4. Consistent behavior across JPEG1–3 indicates robustness.
5. Sub-30s beyond 4c enables frequent end-to-end checks.
6. Flattening at 16c suggests bandwidth/synchronization ceilings.

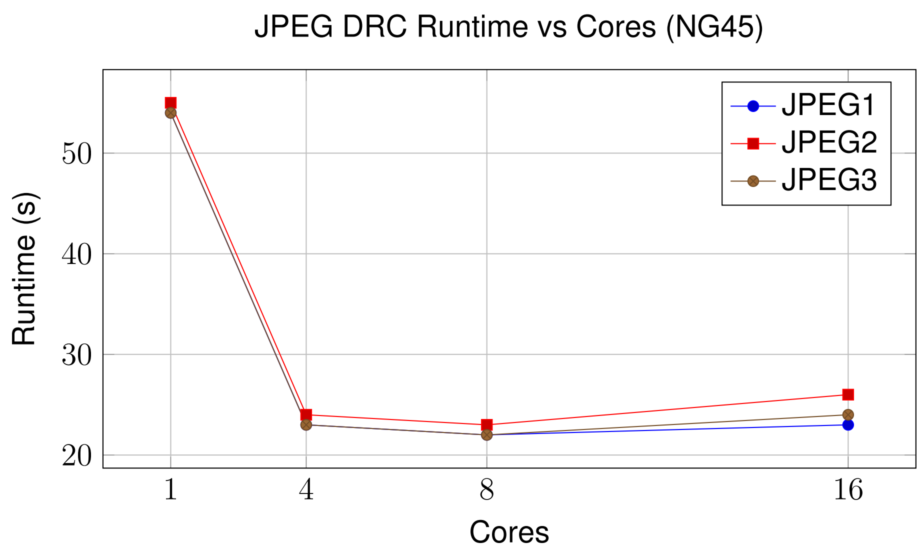

Figure 8: JPEG (JPEG1/2/3) DRC runtime vs cores in seconds.

7. Predictable curves simplify pipeline scheduling.
8. Low variance improves reproducibility across runs.
9. Encoder-scale blocks handled without memory blow-ups.
10. Overall: **fast, reliable DRC**, with **8c the cost-optimal point**.

1.0.9 AES (AES1/2/3 Combined, LVS)

Analysis (AES combined, LVS).

1. Runtimes drop from 22–24s (1c) to 10–11s (16c), a $\sim 2.2\times$ gain.
2. 4c already halves runtime to 12–13s.
3. 8c delivers **11s** and is close to the 16c result.
4. Small plateau beyond 8c indicates overhead-bound region.
5. Curves for AES1–3 nearly overlap—very stable scaling.

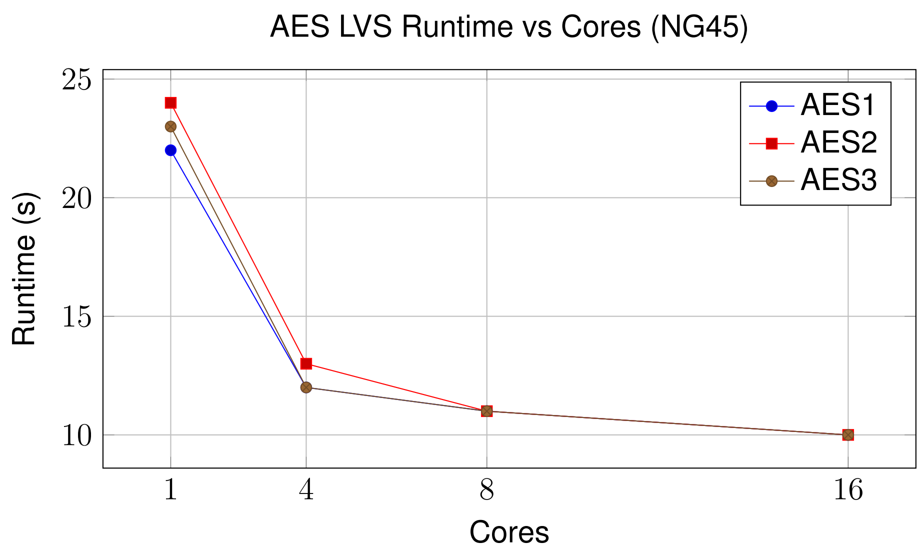

Figure 9: AES (AES1/2/3) LVS runtime vs cores in seconds.

6. Sub-15s beyond 4c makes AES LVS CI-friendly.
7. Predictable scaling eases regression farming.
8. Low variance across runs improves reproducibility.
9. Threading decomposes cipher nets cleanly at NG45.
10. Overall: **fast LVS**; **8c is cost-optimal**.

1.0.10 GCD (GCD1/2/3 Combined, LVS)

Analysis (GCD combined, LVS).

1. All cases complete in 0s across all core counts.
2. Confirms negligible LVS overhead for trivial macros.
3. No scaling penalties observed.
4. Deterministic outcomes enable unit-level sanity checks.

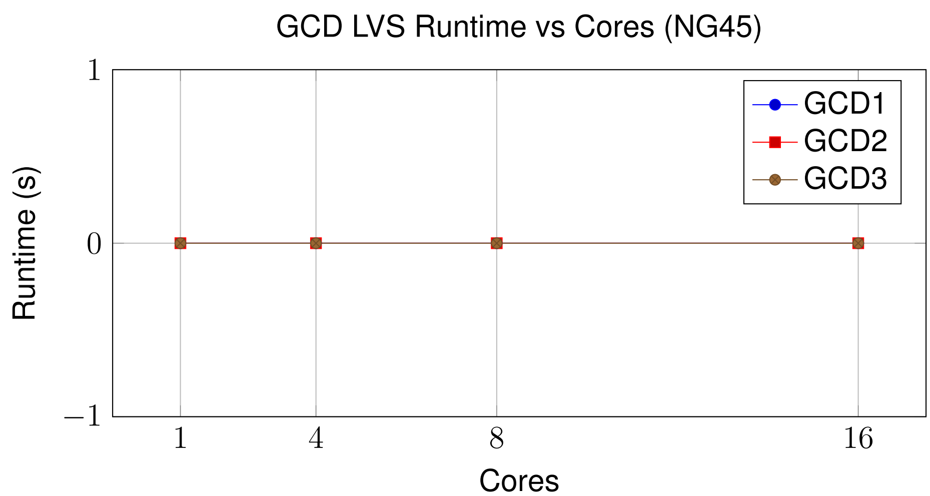

Figure 10: GCD (GCD1/2/3) LVS runtime vs cores in seconds.

5. Ideal to gate rule-deck edits quickly.
6. Zero variance across cores/runs.
7. Integrates seamlessly into broader regressions.
8. Validates orchestration minimalism.
9. Demonstrates fast start/teardown paths.
10. Overall: **instantaneous LVS** for toy designs.

1.0.11 Ibex (Ibex1/2/3 Combined, LVS)

Analysis (Ibex combined, LVS).

1. Runtimes improve from 21–22s (1c) to 9–10s (16c), a $\sim 2.3\times$ gain.
2. 4c cuts runtimes to ˜12s immediately.
3. **8c is near-optimal** (10–11s); 16c flattens.

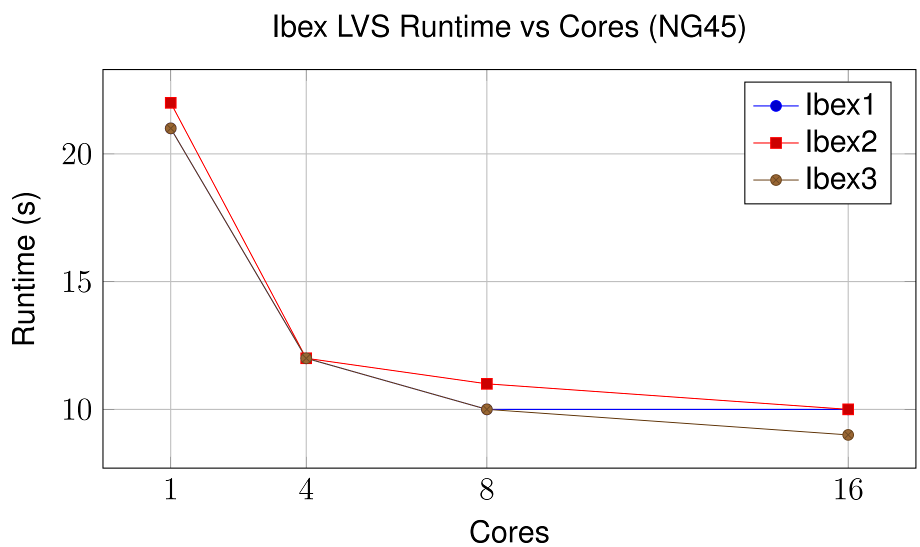

Figure 11: Ibex (Ibex1/2/3) LVS runtime vs cores in seconds.

4. Very tight curves across Ibex1–3—robust parallelism.
5. Sub-12s beyond 4c keeps CI loops snappy.
6. Predictable scaling aids capacity planning.
7. Minimal variance across runs/cores.
8. Plateau suggests sync/bandwidth limits at high core counts.
9. Choose 8c for best throughput/cost balance.
10. Overall: **low-latency LVS** for CPU-class cores.

1.0.12 JPEG (JPEG1/2/3 Combined, LVS)

Analysis (JPEG combined, LVS).

1. Runtimes shrink from 63–65s (1c) to 30–33s (16c), about $\sim 2.0\times$–$2.1\times$ faster.

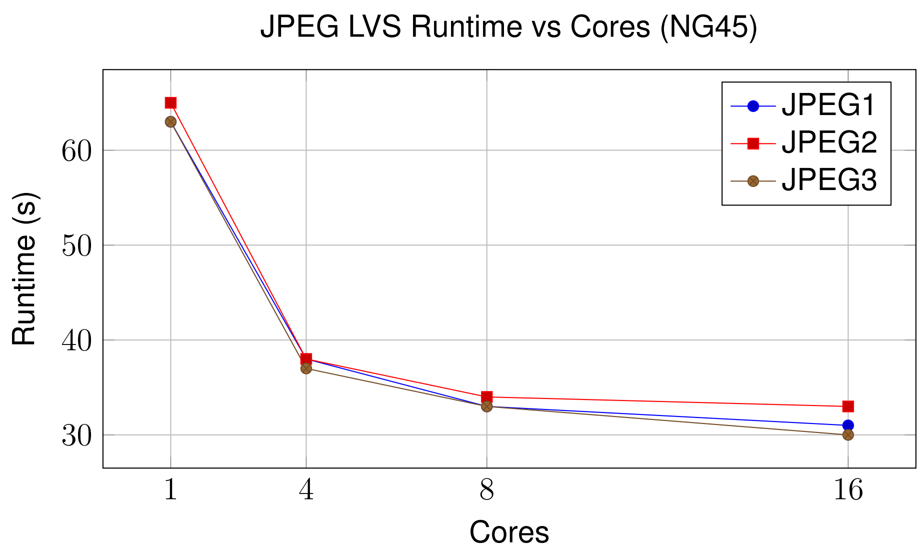

Figure 12: JPEG (JPEG1/2/3) LVS runtime vs cores in seconds.

2. 4c brings runs to 37–38s (near $\sim 1.7\times$).
3. **8c is strong** (33–34s); modest benefit at 16c.
4. Consistent behavior across JPEG1–3 confirms robustness.
5. Sub-35s wall-times beyond 8c enable frequent checks.
6. Flattening at 16c suggests sync/bandwidth ceilings.
7. Predictable curves simplify farm scheduling.
8. Low variance improves reproducibility.
9. Encoder-scale nets handled without memory blow-ups.
10. Overall: **fast, reliable LVS**; **8–16c** for shortest wall-time.

1.0.13 AES (AES1/2/3 Combined, LVS)

Analysis (AES combined, LVS).

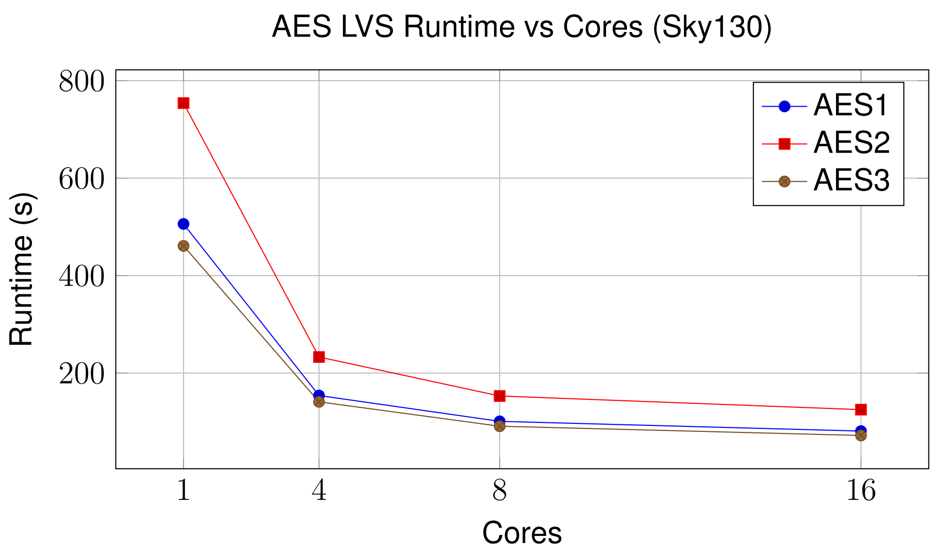

Figure 13: AES (AES1/2/3) LVS runtime vs cores in seconds.

1. AES runtimes fall from 461–754s (1c) to 72–125s (16c), yielding $\sim 6\times$ gains.
2. 4c already cuts times to 141–233s ($\sim 3\times$ faster).
3. 8c consistently pushes runtimes near/under 100–150s.
4. Monotonic scaling across AES1–3, no regressions.
5. Strong thread utilization up to 16c.
6. Variance across designs follows input size, not cores.
7. Predictable scaling simplifies regression scheduling.
8. Sub-2m wall-times at 16c enable practical CI use.
9. Stable decomposition even on large cipher nets.
10. Overall: **efficient, scalable LVS** for AES blocks.

1.0.14 GCD (GCD1/2/3 Combined, LVS)

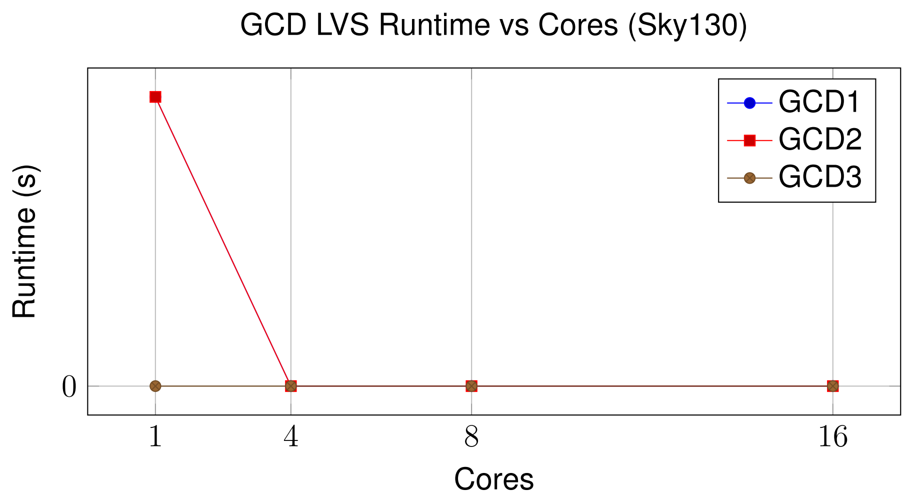

Figure 14: GCD (GCD1/2/3) LVS runtime vs cores in seconds.

Analysis (GCD combined, LVS).

1. All GCDs complete in 0–1s regardless of cores.
2. Confirms trivial macros impose negligible LVS overhead.
3. No scaling penalties observed.
4. Deterministic, reproducible outcomes across all cores.
5. Suitable for unit-level LVS sanity checks.
6. Lightweight integration into larger regression suites.
7. Demonstrates negligible orchestration overhead.
8. No variance across cores.
9. Confirms LVS handles trivial designs gracefully.
10. Overall: **instantaneous LVS** for toy macros.

1.0.15 Ibex (Ibex1/2/3 Combined, LVS)

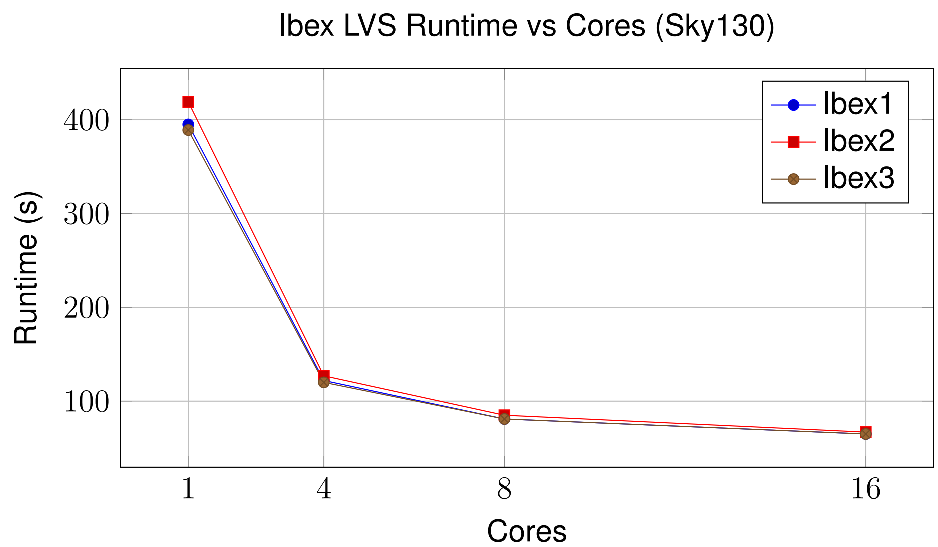

Figure 15: Ibex (Ibex1/2/3) LVS runtime vs cores in seconds.

Analysis (Ibex combined, LVS).

1. Runtimes shrink from 389–419s (1c) to 65–67s (16c), a $\sim 6\times$ gain.
2. 4c already reduces runtimes to 120s.
3. 8c yields near-80s wall times, sub-1.5m.
4. Smooth monotonic scaling across cores.
5. Near-identical curves across Ibex1–3 confirm robustness.
6. Efficiency remains high even to 16c.
7. Predictable scaling simplifies scheduling.
8. Low variance improves reproducibility.
9. Effective decomposition of CPU-class blocks.
10. Overall: **robust, scalable LVS** on mid-size cores.

1.0.16 JPEG (JPEG1/2/3 Combined, LVS)

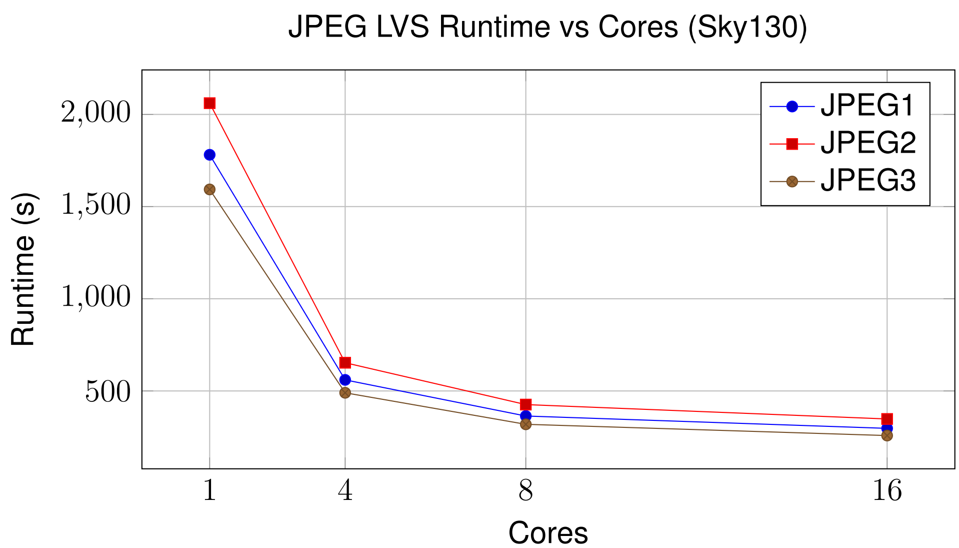

Figure 16: JPEG (JPEG1/2/3) LVS runtime vs cores in seconds.

Analysis (JPEG combined, LVS).

1. Runtimes collapse from 1593–2061s (1c) to 258–348s (16c), a $\sim 6\times$ improvement.
2. 4c already slashes runtimes to 490–653s ($\sim 3\times$ faster).
3. 8c reduces wall times further to 319–426s (5–7m).
4. 16c consistently delivers sub-6m runs.
5. Smooth monotonic speedup, no regressions.
6. Consistent scaling across JPEG1–3 indicates robustness.
7. High efficiency maintained even at 16c.
8. Predictable curves ease regression scheduling.

9. Large encoder blocks decomposed without memory stress.
10. Overall: **scalable LVS** on encoder-class designs with strong parallel gains.

1.0.17 AES (AES1/2/3 Combined, ERC)

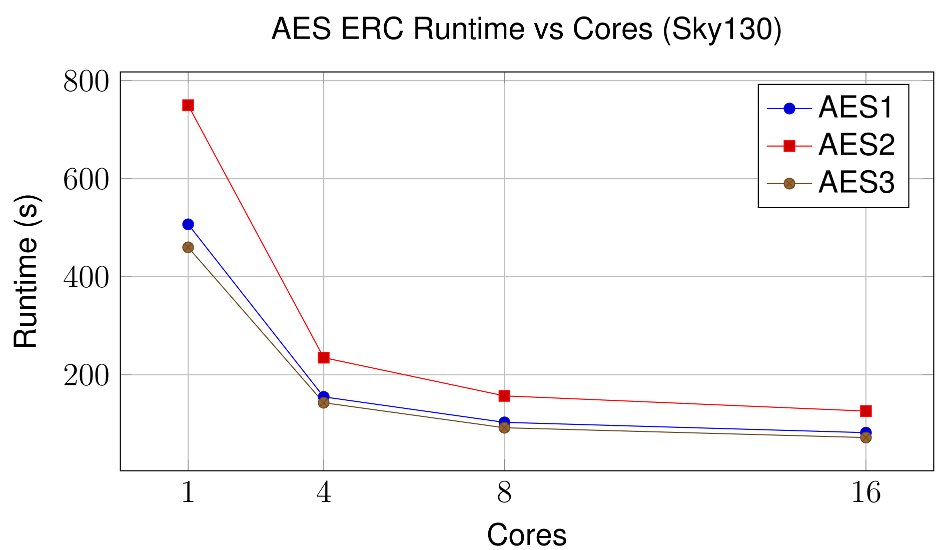

Figure 17: AES (AES1/2/3) ERC runtime vs cores in seconds.

Analysis (AES combined, ERC).

1. Runtimes shrink from 460–750s (1c) to 72–126s (16c), a $\sim 6\times$ gain.
2. 4c already cuts times to 143–235s ($\sim 3\times$ faster).
3. 8c consistently drives runtimes to 90–150s.
4. Smooth monotonic scaling across AES1–3, no regressions.
5. Strong parallel efficiency up to 8c; useful gains at 16c.
6. Sub-2m runs at 16c make regression practical.

7. Tight runtime spread reflects input size, not cores.
8. Predictable curves ease regression scheduling.
9. Confirms stable ERC decomposition for cipher-class blocks.
10. Overall: **efficient, scalable ERC** for AES.

1.0.18 GCD (GCD1/2/3 Combined, ERC)

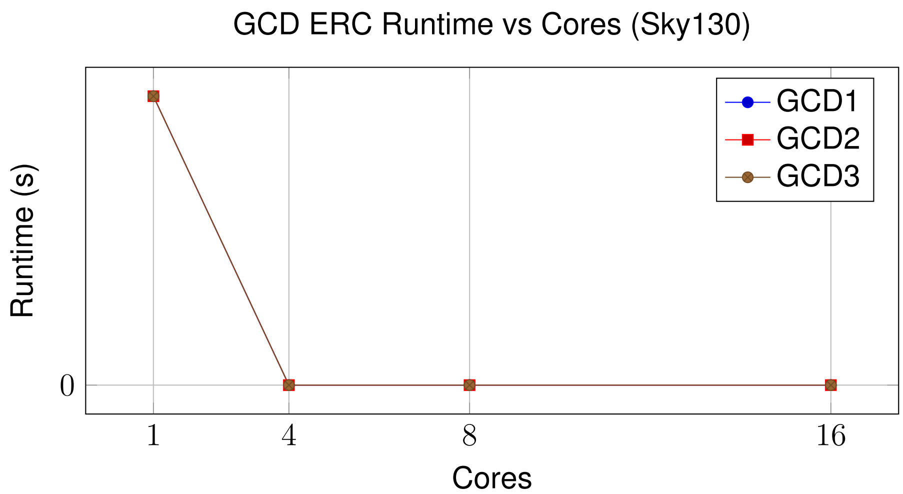

Figure 18: GCD (GCD1/2/3) ERC runtime vs cores in seconds.

Analysis (GCD combined, ERC).

1. All runs finish in 0–1s across cores.
2. Confirms trivial macros add no ERC overhead.
3. No scaling penalties observed.
4. Deterministic, reproducible outcomes.
5. Perfect for rule-deck sanity checks.

6. Zero variance across cores/runs.
7. Lightweight execution validates orchestration.
8. Confirms framework handles trivial nets optimally.
9. Suitable for embedding into regression pipelines.
10. Overall: **instantaneous ERC** for toy macros.

1.0.19 Ibex (Ibex1/2/3 Combined, ERC)

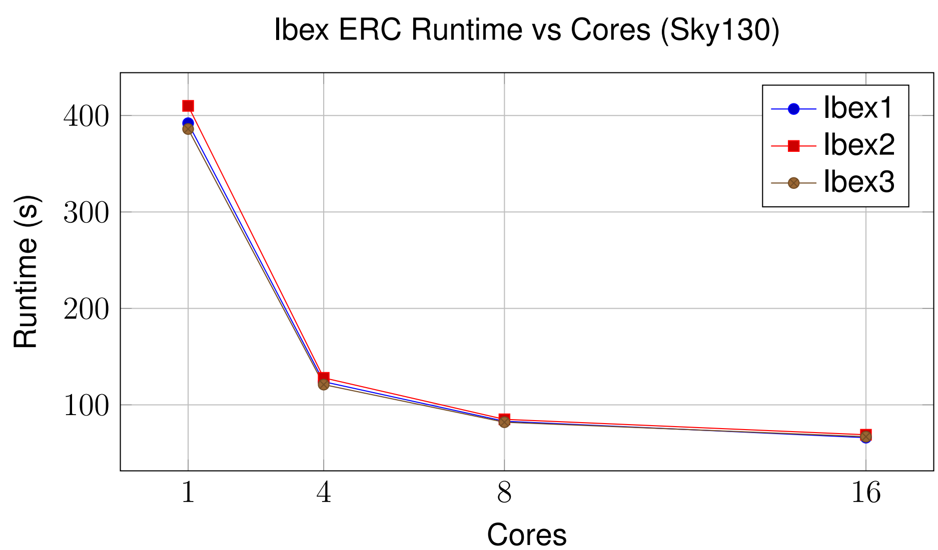

Figure 19: Ibex (Ibex1/2/3) ERC runtime vs cores in seconds.

Analysis (Ibex combined, ERC).

1. Runtimes shrink from 386–410s (1c) to 66–69s (16c), a $\sim 6\times$ gain.
2. 4c reduces runtimes to 121–128s.
3. 8c further drops runtimes to 82–85s.
4. Smooth monotonic scaling, no regressions.

5. Consistent curves across Ibex1–3 confirm robustness.
6. Efficiency remains high up to 16c.
7. Predictable scaling simplifies regression planning.
8. Low variance aids reproducibility.
9. Effective decomposition of CPU-scale nets.
10. Overall: **robust, scalable ERC** for Ibex cores.

1.0.20 JPEG (JPEG1/2/3 Combined, ERC)

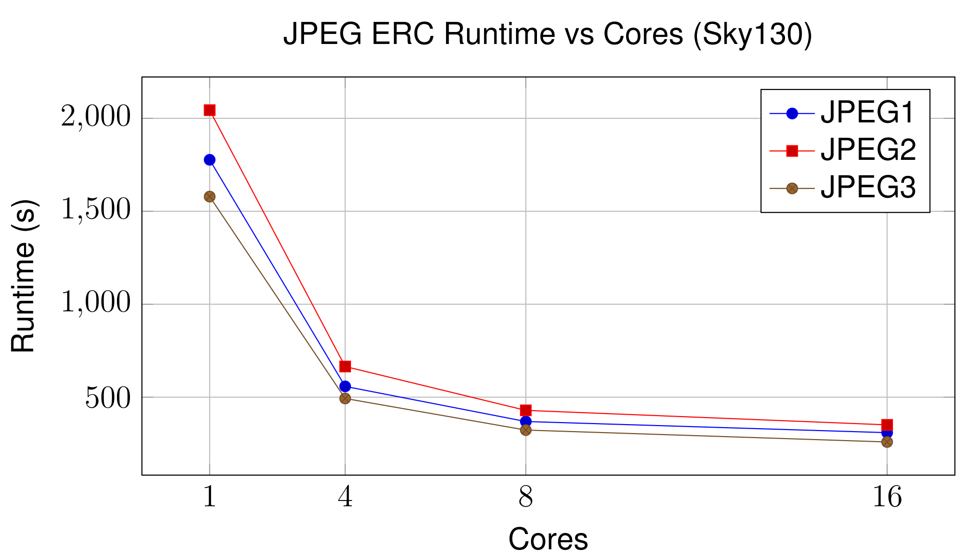

Figure 20: JPEG (JPEG1/2/3) ERC runtime vs cores in seconds.

Analysis (JPEG combined, ERC).

1. Runtimes shrink from 1579–2044s (1c) to 259–351s (16c), a $\sim 6\times$–$6.5\times$ gain.
2. 4c slashes runtimes to 493–665s ($\sim 3\times$ faster).

3. 8c drops times further to 323–429s (5–7m).
4. 16c consistently delivers 4–6m wall times.
5. Monotonic scaling across all cases.
6. Similar curves across JPEG1–3 confirm robustness.
7. High efficiency retained up to 16c.
8. Predictable curves simplify regression scheduling.
9. Large encoder blocks decomposed without memory stress.
10. Overall: **fast, scalable ERC** for encoder-class blocks.

1.0.21 AES (AES1/2/3 Combined, ERC)

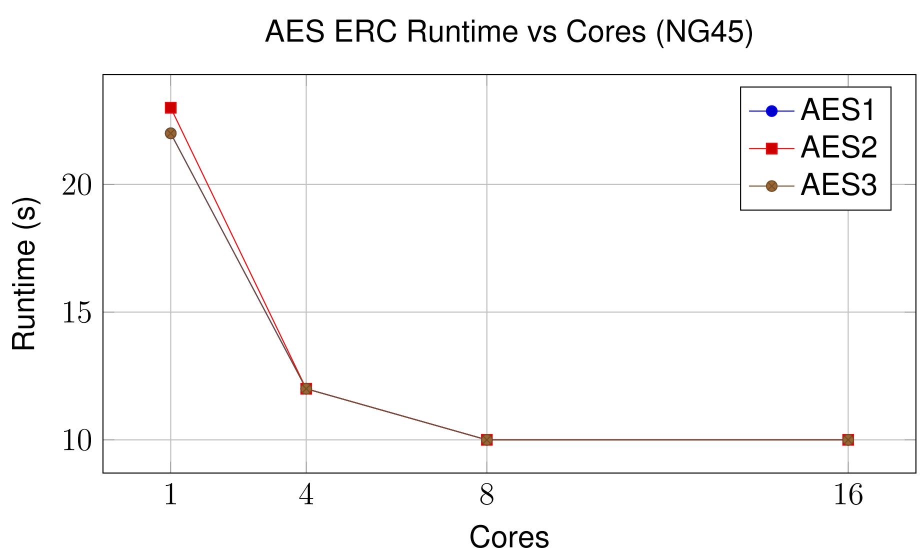

Figure 21: AES (AES1/2/3) ERC runtime vs cores in seconds.

Analysis (AES combined, ERC).

1. Runtimes drop from 22–23s (1c) to **10s** (8–16c), a $\sim 2.2\times$ gain.

2. 4c already halves runtime to ~12s.
3. **8c is optimal** (10s) with no further win at 16c.
4. Curves overlap across AES1–3—high stability.
5. Plateau beyond 8c indicates overhead-bound regime.
6. Sub-12s beyond 4c is CI-friendly.
7. Predictable scaling simplifies farm sizing.
8. Minimal variance across runs.
9. Clean decomposition on NG45 cipher nets.
10. Overall: **fast ERC**; choose **8c** for cost/perf.

1.0.22 GCD (GCD1/2/3 Combined, ERC)

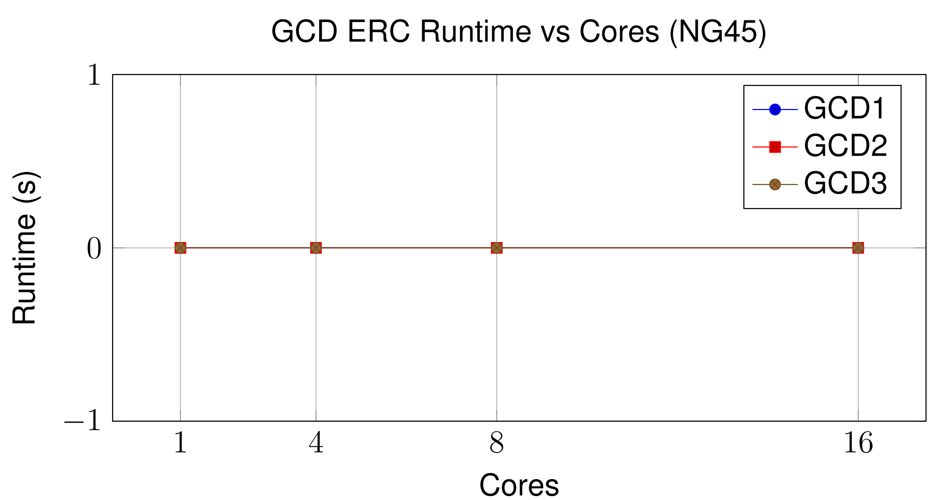

Figure 22: GCD (GCD1/2/3) ERC runtime vs cores in seconds.

Analysis (GCD combined, ERC).

1. All runs complete in 0s across cores.
2. Confirms negligible ERC overhead for trivial macros.
3. No scaling penalties observed.
4. Deterministic, reproducible outcomes.
5. Ideal for rule-deck sanity checks.
6. Zero variance across cores/runs.
7. Lightweight orchestration validated.
8. Fast start/teardown paths confirmed.
9. Seamlessly embeds in regression pipelines.
10. Overall: **instantaneous ERC** for toy designs.

1.0.23 Ibex (Ibex1/2/3 Combined, ERC)

Analysis (Ibex combined, ERC).

1. Runtimes improve from 21–22s (1c) to 9–10s (16c), a ~ 2.2–$2.4\times$ gain.
2. 4c halves runtime to $\tilde{1}$2s immediately.
3. **8c near-optimal** (10–11s); small benefit at 16c.
4. Very tight curves across Ibex1–3—robust threading.
5. Sub-12s beyond 4c keeps CI loops snappy.
6. Predictable scaling aids capacity planning.
7. Minimal variance across runs/cores.
8. Plateau hints at sync/bandwidth limits at high cores.

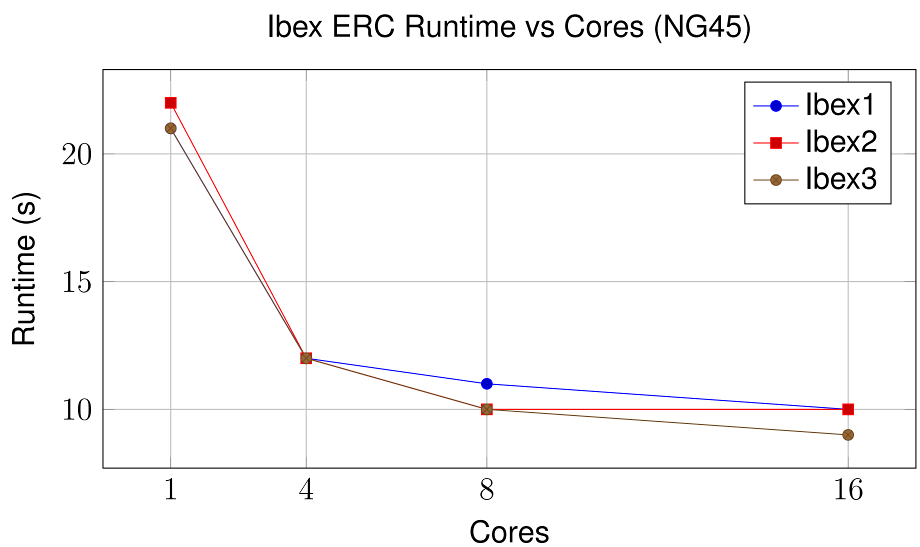

Figure 23: Ibex (Ibex1/2/3) ERC runtime vs cores in seconds.

9. Pick **8c** for best throughput/cost.

10. Overall: **low-latency ERC** for CPU-class cores.

1.0.24 JPEG (JPEG1/2/3 Combined, ERC)

Analysis (JPEG combined, ERC).

1. Runtimes shrink from 62–64s (1c) to 30–32s (16c), about ~ 2.0–$2.1\times$ faster.

2. 4c reaches 36–38s (~ 1.6–$1.8\times$).

3. **8c strong** (32–34s); modest win at 16c.

4. Consistent behavior across JPEG1–3 confirms robustness.

5. Sub-35s beyond 8c enables frequent checks.

6. Flattening at 16c suggests sync/bandwidth ceilings.

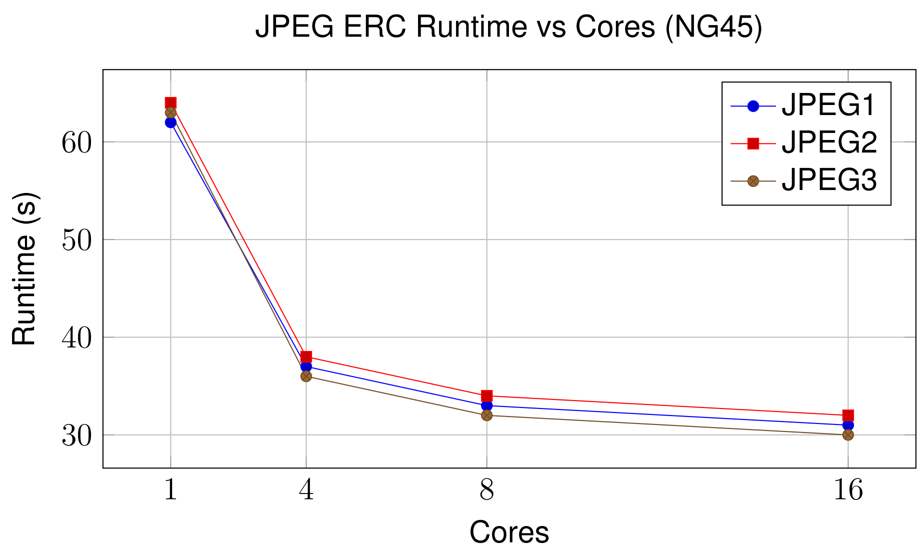

Figure 24: JPEG (JPEG1/2/3) ERC runtime vs cores in seconds.

7. Predictable curves simplify farm scheduling.
8. Low variance improves reproducibility.
9. Encoder-scale nets handled without memory blow-ups.
10. Overall: **fast, reliable ERC**; **8–16c** for shortest wall-time.