Superscalar OoO Renamer + EVES-Inspired Value-Prediction Confidence Layer

C++ on top of the 721sim cycle-accurate skeleton · NCSU ECE 721 (Advanced Microprocessor Architecture, Prof. Eric Rotenberg) · Feb-Apr 2026 · Projects 2-4, three-phase build

Topics: *Register renamer class (RMT + AMT + FL + AL + GBM + branch checkpoints) · MIPS R10000-style branch masks · phase-bit circular buffers · per-stage SVP/VPQ integration · EVES-inspired confidence layer (FPC + per-type denominators + cooldown + SafeStride) · three-round HPC sweep tuning · VR-1 recovery · 22 SPEC validation runs*

The one-liner: Three-phase microarchitecture project on the 721sim cycle-accurate superscalar OoO simulator. P2: wrote the C++ renamer class from spec - eight data structures, MIPS R10000-style branch checkpoints, phase-bit circular buffers. P3: wired that renamer into the full 721sim pipeline, filling the 18 intentionally-omitted FIX_ME code segments across the seven stage files to produce a working baseline OoO pipeline. P4 (built with a project partner): added the value-prediction path - a Stride Value Predictor + Value Prediction Queue - plus an EVES-inspired confidence layer (FPC, per-instruction-type denominators, cooldown, SafeStride), tuned over three HPC sweep rounds.

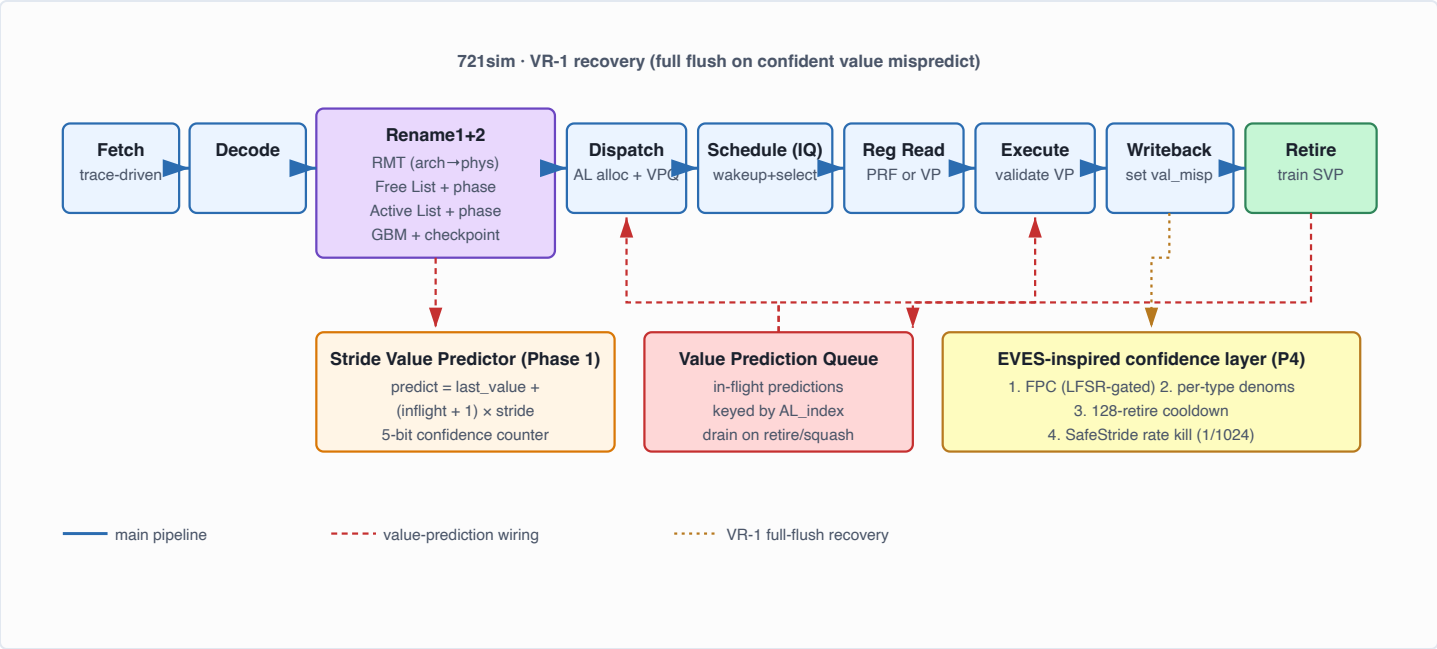
What landed: All Project 2 + 3 validation runs match course-provided golden IPC outputs. The Phase 1 SVP wins on most SPEC benchmarks but regresses on hmmer, povray, sjeng, and bzip2. The tuned EVES config (4/2/2 INTALU/FPALU/LOAD denominators) reaches **2.0294 H-mean IPC, +0.49% over the baseline, with zero per-benchmark regression**.

Scope - three phases

Phase	What I wrote	Authorship
P2 - Renamer class	Full C++ renamer (`renamer.h` / `renamer.cc`) - eight structures, full public API for rename / dispatch / branch-resolve / retire / squash + fault-signaling bits.	Solo
P3 - Renamer integration	Integrated the Project 2 renamer into the simulator and filled the 18 intentionally-omitted FIX_ME code segments across `rename.cc` , `dispatch.cc` , `register_read.cc` , `execute.cc` , `writeback.cc` , `retire.cc` , `squash.cc` - producing a working baseline OoO pipeline. No value prediction yet.	Solo
P4 - Value prediction + EVES competition	Integrated the value-prediction path (Stride Value Predictor + Value Prediction Queue) across the relevant stages, then designed and shipped the four-mechanism EVES-inspired confidence layer (FPC + per-type denoms + cooldown + SafeStride). CLI plumbing for `--vp-eves --vp-eves-denoms` . Three rounds of HPC sweeps.	**Built with a project partner**

The 721sim skeleton (Rotenberg's course infrastructure) provided the trace front-end, fetch unit, payload buffer, execution lanes, LSU shell, IQ, and pipeline-register skeleton. The Phase 1 SVP - including the E-Stride formula `predict = last_value + (inflight+1)×stride` and the 5-bit confidence counter - was also course-provided. We integrated it into the pipeline and added the confidence overlay in Project 4.

Pipeline at a glance



721sim pipeline (top spine) with the value-prediction band (bottom). The EVES confidence layer governs when the SVP is allowed to issue a confident prediction; a confident mispredict triggers the VR-1 full-flush recovery, so the confidence design is the critical lever.

Renamer internals

Eight cooperating structures:

- **RMT** - current arch→phys mapping. Read by source renames, overwritten on destination rename.
- **AMT** - committed arch→phys mapping. Only updated at `commit()`. Used to restore the RMT on a full-pipeline squash (exception / load violation / value mispredict - no specific branch to restore from, so restore from committed state).
- **Free List + Active List** - circular FIFOs with head/tail pointers and *phase bits*. Phase bit disambiguates empty (head==tail, same phase) from full (head==tail, opposite phase). Cheaper than reserving a slot or carrying a separate count.
- **PRF + PRF_ready** - values and ready bits stored separately. Ready bits flip every cycle for every in-flight producer/consumer; isolating them gives a tighter abstraction than co-locating with values.
- **GBM** - `uint64_t` Global Branch Mask. Each in-flight branch owns one bit (= its branch ID = checkpoint index). Each in-flight instruction carries a copy of the GBM-at-rename as its `branch_mask`. MIPS R10000-style.
- **Branch checkpoints** - per-branch recovery snapshot: shadow RMT + FL_head + FL_head_phase + GBM-at-checkpoint. No AL snapshot - AL tail is reconstructed from the branch's AL_index, much cheaper.

Two squash paths

> **Branch mispredict - `resolve(AL_index, branch_ID, false)` :** restore RMT + FL_head + GBM from the branch's checkpoint, clear that branch's bit, walk AL tail back to the slot after AL_index, recompute counts. The br_misp bit on the AL entry is deliberately *not* set, because recovery happens inside this function - setting it would cause a second full squash at retire.

> **Full pipeline squash - `squash()` :** restore RMT from AMT (no specific branch to restore from), clear GBM and checkpoint count, reset AL to empty, rebuild FL by scanning AMT for unused phys regs, mark every PRF ready. This is what fires on a confident value-mispredict under VR-1 - and that's why the EVES confidence layer matters so much: every confident miss costs the entire pipeline.

Per-stage integration (Project 3)

Each pipeline-stage file has FIX_ME blocks that connect the instruction payload (`PAY.buf[index]`) to the renamer API. Notable details: 721sim has three source slots (A, B, D - D is for stores/ternary ops) and one destination (C); branch checkpoints are taken only on instructions with `PAY.buf[i].checkpoint` set; the `branch_mask` rides the pipeline registers (not the payload). Retire is a two-step `precommit()` → `commit()` - `precommit` inspects the AL head for any of (exception, load_viol, val_misp) and routes to full squash if found; otherwise `commit` updates the AMT and recycles the old phys reg back to the FL tail.

EVES-inspired confidence layer (Project 4)

EVES (Enhanced-VTAGE + Enhanced-Stride) is Seznec's CVP-1 entry - 30.8% IPC on the original CVP-1 traces. We didn't ship the full thing: (1) E-VTAGE needs new tagged geometric-history tables and branch-history plumbing that doesn't drop into 721sim cleanly, (2) the Phase 1 SVP already implemented E-Stride. We focused on the confidence layer, which is where the regressions actually come from.

Why the regressions: under VR-1, break-even accuracy on confident predictions is ~99.5% (Calder 1999). The base SVP's 5-bit counter reaches "confident" after 31 consecutive correct predictions - nowhere near enough evidence for a 99.5% bar. Phase boundaries in hmmer / povray / sjeng / bzip2 trigger confident mispredicts → full flushes → net loss.

Four shipped mechanisms, all gated by `--vp-eves --vp-eves-denoms` :

#	Mechanism	What it does
1	**Forward Probabilistic Counter**	16-bit LFSR generates a sample per `train()`; confidence counter increments only when `(sample % denom) == 0`. At denom=4, it now takes ~124 correct predictions to reach confident (vs. 31 baseline) - effective deepening of the threshold.
2	**Per-instruction-type denominators**	Three buckets: INTALU, FPALU, LOAD (matching 721sim's existing `parameters.h` flags). Three, not Seznec's four - no cache-hint signal in 721sim, so LONGLAT folds into INTALU. Winner denoms: 4 / 2 / 2.
3	**128-retire cooldown**	Single global counter. Predictions gated for 128 retires after any confident mispredict. Catches mispredict bursts at phase boundaries. ~10 lines of code.
4	**SafeStride rate monitor**	16-bit miss-rate counter; disables stride prediction globally if rate > 1/1024. Counters halve every 1M retires so a phase doesn't kill prediction permanently. Global panic button for hostile workloads. Our addition - not in the original EVES paper.

Explored but deliberately not shipped: E-VTAGE (multi-table tagged predictor - needs new infrastructure), VR-5 selective squash/replay (needs invasive renamer-checkpoint changes), compact entries (storage optimization, low H-mean leverage).

Three-round HPC sweep

Round	What was tried	What we learned
1	Five symmetric configs.	None beat baseline. xalancbmk regression dominated.
2	LOAD denom = 1 (aggressive on loads to recover xalancbmk).	xalancbmk recovered, but **mcf collapsed from 1.68 → 0.82 IPC**. Pushing one workload up can push another off a cliff.
3	LOAD held at ≥ 2 to protect mcf; pushed INTALU denom down.	Two configs beat baseline. Winner: **4/2/2** (INTALU/FPALU/LOAD), 2.0294 H-mean IPC, +0.49%, zero per-benchmark regression.

What the +0.49% represents: not "make the CPU 0.49% faster" - "make value prediction net-positive on every workload instead of net-positive on average." Baseline regressed on four benchmarks; tuned EVES regresses on none. EVES *protects* the SVP's existing wins rather than discovering new ones.

Design decisions and trade-offs

What it demonstrates:

- Wrote the renamer class from scratch - eight structures, MIPS R10000-style branch masks, phase-bit circular buffers, AMT-based full-squash, per-checkpoint shadow RMT for branch recovery.
- Integrated the renamer across every relevant 721sim pipeline stage - the 18 FIX_ME code segments. All Project 2 + 3 validation runs match golden IPC bit-for-bit.
- Built the value-prediction path (Stride Value Predictor + Value Prediction Queue) and the EVES-inspired confidence layer (with a project partner) - four mechanisms, sweep-tuned over three HPC rounds to the 4/2/2 winner, +0.49% H-mean IPC, zero per-benchmark regression.

Scope and limitations:

- **The 721sim infrastructure was given.** My contribution is the renamer class, the per-stage FIX_ME integrations, and the EVES layer - not the simulator skeleton.
- **The base SVP was course-provided.** The E-Stride formula and the 5-bit confidence counter came with the course infrastructure. We integrated the value-prediction path and added the confidence overlay in Project 4.
- **Project 4 was built with a project partner.** Projects 2 and 3 were my own work.
- **EVES is Seznec's algorithm, not ours.** FPC and per-type denoms are from the EVES paper. Cooldown and SafeStride are our additions.
- **Behavioral simulator, not RTL.** RTL design work is covered in the ECE 564 CNN Pipeline guide.

Common questions about this project

Question	Short answer
How does the renamer work?	Eight C++ structures: RMT (speculative arch→phys), AMT (committed arch→phys, drives full-squash recovery), FL + AL as circular FIFOs with phase bits, PRF + separate ready-bit array, a uint64_t GBM, per-branch checkpoint pool with shadow RMT + FL_head + GBM. MIPS R10000-style branch masks let in-flight instructions know which branches they depend on.
What is the difference between RMT and AMT?	RMT is speculative front-end mapping, written at rename. AMT is committed back-end mapping, updated only at retire. On a full pipeline squash (exception / load-viol / value mispredict - no specific branch to restore from), the renamer restores RMT *from* AMT. The AMT is ground truth for committed architectural state.
Why phase bits on FL and AL?	Circular-buffer empty/full disambiguation. When head==tail you can't tell empty from full without extra state. Phase bits toggle on every wrap - same phase + head==tail = empty; opposite phase + head==tail = full. Cleaner than reserving a slot or maintaining a separate count.
Why does it take EVES to fix the regressions?	VR-1 makes a confident mispredict a full pipeline flush, so break-even is ~99.5% (Calder 1999). The base SVP's 5-bit counter saturates after 31 correct predictions - not enough evidence for a 99.5% bar. EVES slows saturation (FPC), tunes per-instruction-type aggressiveness (denominators), pauses predictions after a miss (cooldown), and globally bails out on hostile workloads (SafeStride).
How did the sweep find 4/2/2?	Three HPC rounds. R1: five symmetric configs, none beat baseline (xalancbmk regression). R2: pushed LOAD denom to 1 to recover xalancbmk - mcf collapsed 1.68 → 0.82 IPC. R3: held LOAD ≥ 2 to protect mcf, pushed INTALU denom down. Winner: 4/2/2 at 2.0294 H-mean IPC, +0.49%, zero per-benchmark regression.
Why only three buckets in the per-type table?	Seznec's table has four buckets, including a cache-hint-distinguished LONGLAT. 721sim has no cache-hint plumbing, so LONGLAT folds into INTALU. Three buckets exactly match the eligibility flags already in `parameters.h` - cleaner integration, no fabricated signals.

Question	Short answer
How would this look in RTL?	The renamer is a real critical-path block. RMT and AMT are small register-file-like SRAMs; FL is a small dedicated SRAM with head/tail; AL is a circular buffer with associated bit storage; branch checkpoints become physically-realized shadow-RMT snapshots, one per outstanding branch. VPQ would be a CAM keyed on AL_index. None of that timing is in this simulator - it's behavioral C++ - but the structural map transfers directly.
How much of the base OoO simulator is original?	No - 721sim is course infrastructure. I wrote the renamer class (P2) and per-stage integrations (P3). The ECE 563 OoO simulator (Dec 2025, separate guide) is a different codebase I did write from scratch - that's the from-the-ground-up cycle-accurate sim. 721sim is the heavier-infrastructure follow-on focused on renamer + VP.