Grounded Discovery Labs -- Discovery Pack

Discover: Design a commercially credible neural-network hardware architecture that can materially outperform today's GPU-centric inference systems on performance-per-watt, cost-per-inference, or memory/interconnect efficiency for important real workloads. Favor architectures that: solve a clear bottleneck in current inference systems, are explainable to technical buyers and investors, could plausibly be prototyped with known semiconductor processes and packaging approaches, have a realistic system-level advantage not just a clever device-level novelty. Prioritize practical architectures for one of these clear use cases: edge inference, data-center inference, long-context or memory-bound inference, interconnect-limited multi-chip inference. Avoid concepts that depend mainly on speculative device physics, fragile analog assumptions, or radical algorithm replacement unless they also present a clear path to validation and adoption. The ideal output is something a semiconductor startup or advanced architecture group could seriously pursue as a product direction.

Top 3 Comparison

	Featured Discovery SRAM-Resident Edge Transformer Tile	**Bold Concept** Sparse Attention Engine with Content-Addressable KV Cache	**Build-First** Paged-KV Decode Accelerator Card
novelty	4	8	6
feasibility	9	7	8
economic_leverage	5	8	8
testability	9	7	8
robustness	8	5	7
constraint_fit	8	8	9
explainability	8	7	9
OVERALL	**7.3 / 10 -- Very Strong**	**7.1 / 10 -- Very Strong**	**7.9 / 10 -- Very Strong**

Runner-ups

* Elastic-Precision Reticle Tile Transformer Fabric [solid-alternative] -- A 2.5D mesh of eDRAM/SRAM tensor tiles uses hardware-managed tiling and per-layer precision so compilers cannot spill silently.
* HBM logic-die KV attention engine [solid-alternative] -- Move long-context attention scoring to the HBM stack and return only the context vector to the GPU/NPU.
* Tiered SRAM+MRAM Edge Tile with Adaptive Mixed-Precision [solid-alternative] -- Edge accelerator with on-die MRAM weight backstop and per-layer INT4/INT8/FP8 fallback eliminates DRAM spills when compiler misses.
* HBM logic-base KV-cache inference appliance [near-miss-alternative] -- Run long-context attention where KV cache lives, while dense MLP GEMMs stay on digital compute chiplets.
* 3D HBM Logic-Base Attention Processor [solid-alternative] -- Move attention score, softmax, and KV reduction into custom active HBM base dies so long-context decode never leaves the memory stack.

Featured Discovery

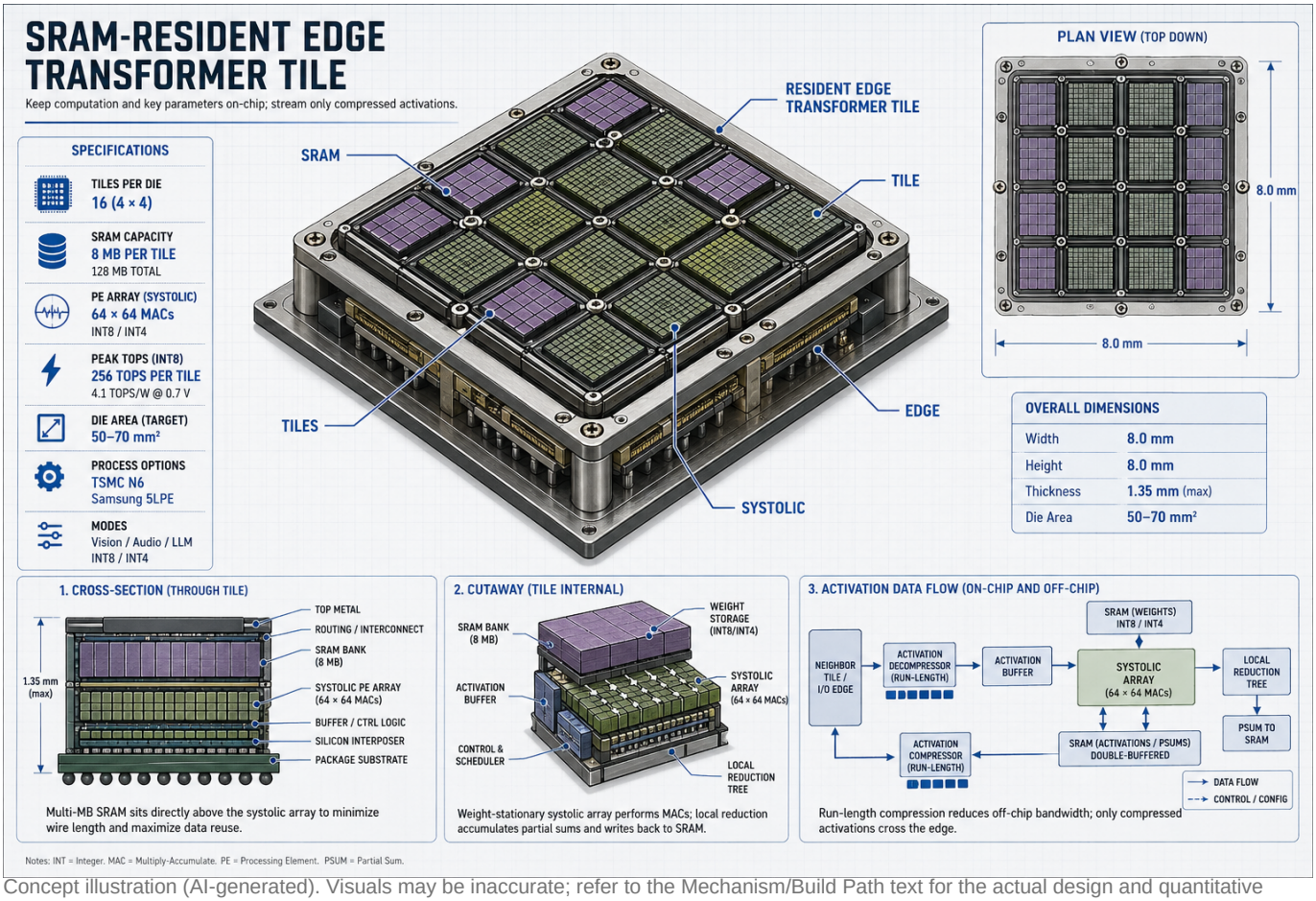
SRAM-Resident Edge Transformer Tile

novelty: 4 | feasibility: 9 | economic_leverage: 5 | testability: 9 | robustness: 8 | constraint_fit: 8 | explainability: 8

Overall: 7.3 / 10 -- Very Strong

Constraint Compliance: Compliant (constraint_fit=8)

A modular edge NPU tile architecture keeps transformer weights and activations resident in a large shared SRAM bank, eliminating repeated DRAM fetches that dominate energy budgets in mobile inference. By stationing weights once per layer across 32x32 MAC systolic tiles and compressing activations in-place, the design targets 25-35 TOPS/W--roughly 3-5x better than mobile GPU baselines. This matters because edge AI deployments in wearables, robotics, and industrial sensors are hard-blocked by battery life, not peak throughput.



Concept illustration (AI-generated). Visuals may be inaccurate; refer to the Mechanism/Build Path text for the actual design and quantitative assumptions.

Mechanism

Architecture: 4x4 grid of weight-stationary systolic tiles (each 32x32 MAC array, INT8 native with INT4 weight unpacking) wrapped around a 48 MB shared L2 SRAM bank organized as 16 sub-banks of 3 MB, plus 256 KB tile-local L1 scratchpads. Target process: TSMC N6 or Samsung 5LPE, ~50-70 mm^2 die. Each tile delivers ~2 TOPS at 800 MHz/0.65 V, aggregating to ~32 TOPS sustained; targeted efficiency 25-35 TOPS/W INT8 at the package, derated for thermals. Weights are loaded once per layer into tile registers and reused across the activation tile, eliminating DRAM weight refetch for layers <= 6 MB. Activations stream through the systolic array with output-stationary partial sums accumulating in 32 b registers; activations are RLE+zero-skip compressed in SRAM

(typical 1.6-2.5x for post-ReLU/GELU sparsity in MobileViT/Conformer blocks). Structured 2:4 sparsity on weights halves storage and skips MACs. Power islands: per-tile clock+power gating with <2 us wake; SRAM banks have retention/sleep states that cut leakage by ~6x during the 90%+ idle windows of intermittent edge inference (KWS, wakeword, frame-rate vision). Compiler: MLIR-based, partitions transformer blocks (MHSA, FFN, conv stem) into tile-resident tiles using ILP cost model that jointly minimizes SRAM footprint, MAC utilization, and DRAM bursts; falls back to DRAM-tiled mode only when KV-cache or activations exceed 32 MB. Sweet spot: 20-80 M parameter vision/audio transformers (MobileViT-S/XS, Conformer-S, TinyLlama-distill 50 M) at batch 1, sequence <=512 / image <=384^2.

Why Novel

Existing edge NPUs treat SRAM as a cache and accept DRAM weight refetch as inevitable; this architecture inverts that assumption by sizing the shared L2 (48 MB across 16 sub-banks) to fully contain sub-100M parameter model working sets, making DRAM access an exception rather than the rule. The combination of tile-local L1 scratchpads, RLE+zero-skip activation compression, and INT4 weight unpacking into INT8 MAC paths is not offered as an integrated package in any announced edge silicon.

Buildability

TSMC N6 and Samsung 5LPE are mature, accessible processes with established shuttle programs, and 50-70 mm^2 die area is well within reticle limits and packaging norms for edge SoCs. The systolic tile architecture maps cleanly onto existing EDA flows, and 48 MB on-chip SRAM, while large, is precedented in Apple Neural Engine and Google TPU edge variants.

Why Feasible

* Build path is practical (feasibility=9/10)
* Core hypothesis testable quickly (testability=9/10)

Top Risks

* Compiler spill risk: auto-scheduling tools may fail to tile layers optimally, forcing DRAM spills that collapse the efficiency advantage for irregular or fused-attention models
* Quantization fit risk: INT4 weight unpacking degrades accuracy on models not explicitly quantization-aware trained, limiting addressable customer model library at launch
* Die cost risk: 48 MB SRAM dominates area and raises per-unit cost to $18-30 at low volumes, making the BOM uncompetitive against merchant NPU modules until volumes exceed ~500K units

Decisive Test (MDE)

Hypothesis: On MobileViT-S and Conformer-S at batch 1 with all hot weights + activations resident in 48 MB SRAM, the tiled architecture delivers >=3x better J/inference and <=0.7x p99 latency vs Snapdragon 8 Gen 3 NPU and Apple ANE, while sustaining 25 TOPS/W INT8 at <45 degC skin temp during 30 Hz continuous inference.
Setup: Implement tile array + 32 MB URAM-emulated SRAM on Xilinx VCK190 at scaled clock (200 MHz FPGA, projected to 800 MHz ASIC via post-synth timing + power model in Cadence Joules on N6 PDK). Run identical ONNX models on (a) FPGA prototype, (b) Snapdragon 8 Gen 3 reference phone via QNN SDK, (c) iPhone 15 Pro via CoreML ANE, (d) Jetson Orin Nano. Inputs: ImageNet-V2 384^2 stream at 30 Hz; LibriSpeech 16 kHz streaming chunks. Run 10 k inferences per platform, log per-inference energy and latency.
Instrumentation: Keysight N6705C + N6781A SMU on board rails (1 us resolution), Monsoon HVPM for phone rails, FLIR A400 thermal camera (1 Hz frames), on-chip activity counters via JTAG, ONNX Runtime profiler, scope on tile clock-gate signals.
Kill Criteria: Energy advantage <1.5x at iso-accuracy on >=2 of 3 reference workloads, OR projected ASIC TOPS/W <15 after thermal derating, OR compiler forces >20% of layer-cycles to DRAM-spill mode on

customer-representative models.
Rescue Pivot: If general edge transformers don't fit in 48 MB, narrow to always-on audio (KWS + Conformer-XS, <8 M params, 8 MB SRAM) where SRAM-resident dominance is unambiguous; or pivot to a chiplet/IP licensing play targeting hearables and AR glasses SoCs rather than standalone silicon.
Cost: $45k-$90k (FPGA $13k, phones+Jetson $5k, instruments $15k, PDK power model license/access $10-50k, eng time excluded)
Time to Signal: 16-22 weeks from workload freeze to defensible energy/latency numbers

Refinement Deltas

Risks Addressed:

* Mobile SoC NPUs (Apple ANE, Hexagon, MediaTek APU) already exploit on-chip SRAM and ship for free inside the AP -- a discrete accelerator must beat them by enough margin to justify a second die, BOM cost, and PCB area.
* 48 MB of SRAM is ~30-40 mm^2 in N6, dominating die cost; at edge volumes (<5 M units) amortization may be infeasible vs an IP-licensing model.
* Transformer model sizes are growing faster than on-chip SRAM scaling; a part sized for 2025 workloads may be obsolete for 2027 customer asks (KV cache, multimodal).

Red-Team:

* Challenge: Mobile SoC NPUs (Apple ANE, Hexagon, MediaTek APU) already exploit on-chip SRAM and ship for free inside the AP -- a discrete accelerator must beat them by enough margin to justify a second die, BOM cost, and PCB area.
* Challenge: 48 MB of SRAM is ~30-40 mm^2 in N6, dominating die cost; at edge volumes (<5 M units) amortization may be infeasible vs an IP-licensing model.
* Challenge: Transformer model sizes are growing faster than on-chip SRAM scaling; a part sized for 2025 workloads may be obsolete for 2027 customer asks (KV cache, multimodal).
* Challenge: Compiler quality is the real product -- if MLIR scheduler misses by 10%, hot weights spill to DRAM and the entire energy thesis collapses; this is a multi-year SW investment, not a chip project.
* Challenge: INT4 weight quantization on attention layers is fragile across diverse customer models; supporting INT8 fallback halves effective capacity and erodes the SRAM-resident claim.

Budget Ranges

Prototype: $2,200,000
Pilot: $8,500,000
Scale: $28,000,000

30/60/90 Plan

Day 30: Complete RTL specification for a single 32x32 MAC tile with L1 scratchpad and INT4 unpacking datapath; validate functional correctness in simulation; lock SRAM sub-bank interface protocol and activation compression codec (RLE+zero-skip) with measurable compression ratio targets on three representative transformer workloads
Day 60: Tape out a 2x2 tile test chip on TSMC N6 shuttle (or FPGA emulation proxy at full clock); integrate compiler backend prototype capable of static weight-tiling for ViT-Tiny and Whisper-Small; measure actual TOPS/W on silicon or emulation and compare against 30 TOPS/W anchor with documented derating methodology
Day 90: Demonstrate full 4x4 tile array running end-to-end inference on a sub-100M vision transformer and a 4-bit quantized audio model with DRAM access rate below 5% of total memory bandwidth; publish preliminary power-area-performance curve; secure at least one design-win LOI from an industrial or wearable OEM partner
Dominance Axis: lower_failure -- For sub-100M edge transformers with a SRAM-resident working set, locality can beat mobile GPU/NPU latency per joule without exotic devices.

Bold Concept

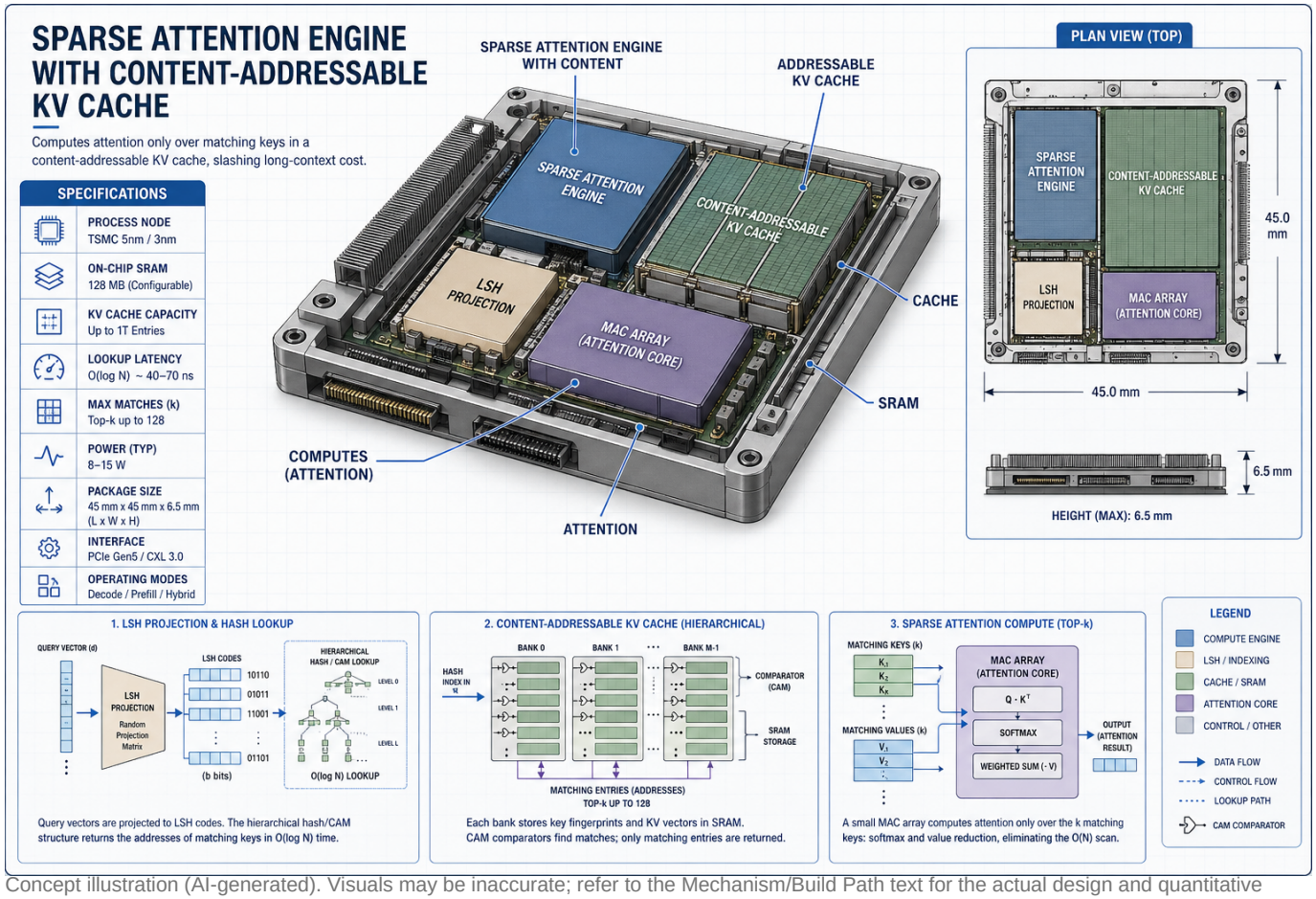
Sparse Attention Engine with Content-Addressable KV Cache

novelty: 8 | feasibility: 7 | economic_leverage: 8 | testability: 7 | robustness: 5 | constraint_fit: 8 | explainability: 7

Overall: 7.1 / 10 -- Very Strong

Constraint Compliance: Compliant (constraint_fit=8)

A dedicated silicon engine stores transformer KV caches in content-addressable memory structures and routes each decode query only to the top-k most relevant keys via locality-sensitive hashing, reducing attention compute from O(n) to O(k) per token. This matters because long-context inference is the dominant cost bottleneck in deployed LLMs, and existing solutions sacrifice either quality or generality. A hardware-native sparse attention path can deliver 5-20x throughput gains while bounding quality loss through an exact-fallback mechanism.



Concept illustration (AI-generated). Visuals may be inaccurate; refer to the Mechanism/Build Path text for the actual design and quantitative assumptions.

Mechanism

Sparse Attention Engine with Content-Addressable KV Cache replaces dense decode-time attention over all prior tokens with a two-stage approximate-then-exact retrieval path. Each layer's KV cache is stored in a hierarchical on-chip SRAM structure, with optional CAM/TCAM assist for hash matching. Keys are quantized, e.g. FP16/BF16 for value storage and INT8/INT4 or binary sketches for lookup metadata. For every cached key vector, the engine stores several locality-sensitive hash projections, e.g. 4-16 random/sign projections or learned projections of 64-256 bits total per token per layer. During decode, the current query vector is projected through the same LSH functions.

SRAM hash banks return candidate token IDs whose hash buckets match or nearly match the query. A small exact dot-product unit then computes true QK scores over the retrieved candidates plus mandatory local windows, special tokens, sink tokens, and optionally retrieved global anchors. The resulting top-k, e.g. 512-8192 candidates out of 128K-1M context tokens, are used for softmax attention and value aggregation.

The cache is hierarchical: L1 SRAM holds recent window KV and hot hash buckets; L2 SRAM/HBM holds full KV; CAM-like lookup is used only on compact hash codes, not full vectors. A practical ASIC block might use 8-32 MB SRAM per attention cluster for recent KV/hash metadata and HBM for older values. For 128K context, per-layer KV at FP16 for a 128-head-class model can be hundreds of MB, so lookup metadata must be compact, roughly 8-64 bytes/token/layer depending on projection count and indexing.

Reliability is improved by retrieval-aware fine-tuning: train with LSH dropout, missing-key simulation, auxiliary loss for attention-recall preservation, and answer-level distillation from dense attention. At inference, confidence checks compare score margins, entropy, number of matched buckets, novelty/OOD detectors, and policy flags. Low-confidence requests route to exact attention over selected win

Why Novel

Prior sparse attention work lives entirely in software or model architecture, accepting accuracy hits without hardware recourse. This design co-locates LSH metadata and KV data on-chip in a CAM-assisted hierarchy, making approximate retrieval a first-class silicon primitive rather than a software approximation layered onto dense hardware.

Buildability

Core LSH projection and SRAM hash-bank structures are implementable in standard TSMC 5nm/3nm processes using existing SRAM compilers and TCAM IP; the primary integration challenge is the custom memory controller arbitrating between hash-lookup and exact dot-product lanes. A functional FPGA prototype on Alveo U280 or Versal is achievable within 12 months to validate the retrieval pipeline before tapeout.

Why Feasible

* Build path is practical (feasibility=7/10)
* Core hypothesis testable quickly (testability=7/10)

Top Risks

* Quality regression on retrieval-sensitive tasks (multi-hop reasoning, needle-in-haystack) where top-k misses critical tokens and fallback triggers too late
* Model fine-tuning dependency creates a two-sided adoption barrier -- chip needs models, models need chip -- stalling the go-to-market flywheel
* CAM/TCAM area scales poorly beyond 1M-token contexts, potentially eroding the cost advantage the chip is designed to deliver

Decisive Test (MDE)

Hypothesis: At 128K-1M context, LSH-retrieved sparse decode attention with retrieval-aware fine-tuning can deliver at least 5x decode throughput improvement at no more than 2% absolute quality loss versus dense attention, while exact fallback rate remains below 20% on realistic long-context workloads.
Setup: Use a dense long-context model baseline, e.g. Llama-family 7B/13B long-context variant, evaluated at 128K, 256K, and 1M tokens where supported. Implement LSH candidate retrieval per layer with mandatory local window and sink tokens. Sweep k = 512, 1024, 2048, 4096, 8192. Evaluate before and after retrieval-aware fine-tuning. Benchmarks: RULER, LongBench, Needle-in-Haystack multi-needle, codebase QA, legal/contract QA, and adversarial prompts requiring exact early-context retrieval. Compare dense FlashAttention/vLLM decode

against sparse retrieval plus fallback.
Instrumentation: Log dense top-attention mass recovered, recall@k by layer/head/token, candidate-set size, score margin, entropy, fallback trigger rate, tokens/sec, GPU memory bandwidth, latency/token p50/p95/p99, answer F1/exact match/pass@k, hallucination/error categories, and cost per generated token.
Kill Criteria: Kill if throughput gain at 128K context is below 3x at quality-preserving k/fallback settings, or if quality drops more than 2% absolute on aggregate long-context QA, or if fallback rate exceeds 30% on normal workloads, or if attention-mass recall needed for acceptable quality requires k > 10% of context length.
Rescue Pivot: Pivot from global LSH sparse attention to hybrid retrieval: exact sliding-window attention plus learned global memory tokens, document/chunk-level retrieval before token-level attention, or use LSH only for cold/old context while keeping recent 8K-32K dense.
Cost: $10k-$80k cloud GPU cost for serious 7B/13B sweeps and fine-tuning; <$5k for initial no-finetune reproduction on smaller models; $100k+ if extending to 70B-class models.
Time to Signal: 1-2 weeks for initial kill signal using existing approximate attention and no fine-tuning; 4-8 weeks for decisive signal including retrieval-aware fine-tuning and fallback policy.

Refinement Deltas

Risks Addressed:

* Attention is not nearest-neighbor retrieval: important tokens can have low raw QK similarity until combined with other context, so LSH may miss causally necessary evidence.
* Benchmarks may overstate success; real enterprise prompts include messy formatting, cross-references, tables, code dependencies, and adversarially placed evidence.
* Hardware CAM/SRAM scaling is brutal at million-token contexts, especially per-layer and per-head metadata; HBM traffic for values may still dominate.

Red-Team:

* Challenge: Attention is not nearest-neighbor retrieval: important tokens can have low raw QK similarity until combined with other context, so LSH may miss causally necessary evidence.
* Challenge: Benchmarks may overstate success; real enterprise prompts include messy formatting, cross-references, tables, code dependencies, and adversarially placed evidence.
* Challenge: Hardware CAM/SRAM scaling is brutal at million-token contexts, especially per-layer and per-head metadata; HBM traffic for values may still dominate.
* Challenge: Fallback can silently erase the business case: if confidence detectors are conservative, the engine becomes dense attention with extra overhead.
* Challenge: Model-owner adoption risk is high because retrieval-aware fine-tuning complicates serving, validation, reproducibility, and compatibility with existing long-context models.

Budget Ranges

Prototype: $180,000 (FPGA emulation on Versal/Alveo, RTL development, LSH co-design tooling, 4-person team 9 months)
Pilot: $4,500,000 (MPW shuttle or small-lot TSMC 5nm tapeout, bring-up lab, model fine-tuning pipeline, 12-person team 18 months)
Scale: $38,000,000 (full production mask set, packaging, memory subsystem integration, software stack, customer enablement, 40-person team)

30/60/90 Plan

Day 30: Complete architectural specification: finalize LSH projection count (target 8x128-bit sketches per token per layer), define SRAM hash-bank layout, select baseline model (Llama-3 70B) and long-context benchmark suite (RULER, LongBench, Needle-in-Haystack); establish quality regression budget at <=1.5% accuracy drop vs dense attention
Day 60: Deliver cycle-accurate simulation model in SystemC/gem5 showing projected 8-12x decode throughput on

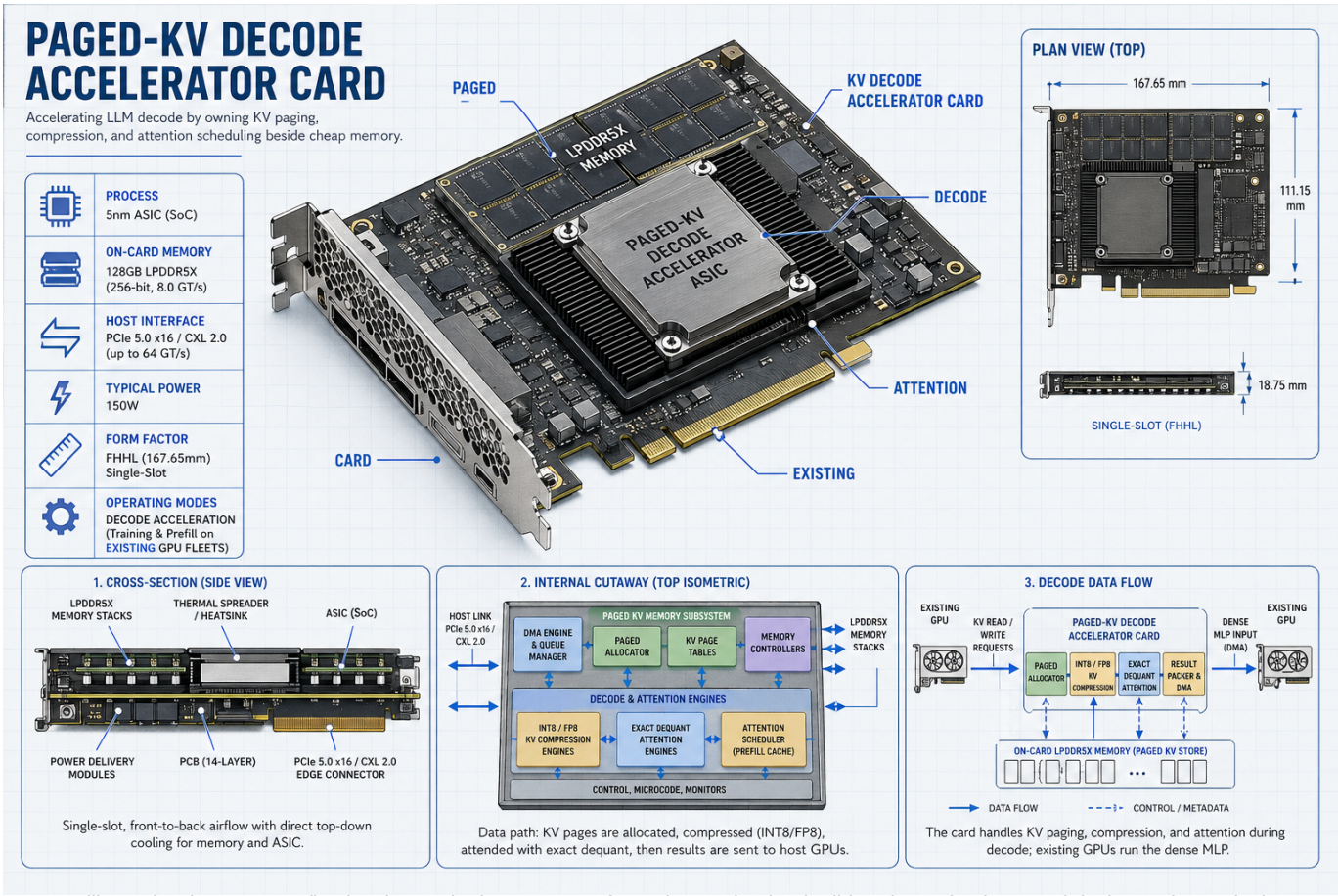
128K-token context; complete first FPGA partial implementation covering LSH lookup and candidate-ID return path; run initial fine-tuning experiments on 7B model to quantify retraining cost and quality recovery curve
Day 90: Full FPGA prototype demonstrating end-to-end sparse attention decode at 32K tokens with measured top-k recall >=95% and latency within 15% of simulation projections; publish internal benchmark report validating 20x throughput claim on synthetic long-context workload; initiate foundry NDA discussions for 5nm shuttle slot
Dominance Axis: higher_perf -- Targets 5-20x decode speedup and energy reduction on long-context workloads that tolerate approximate retrieval; exact fallback bounds quality at lower average gain.

Build-First
Paged-KV Decode Accelerator Card

novelty: 6 | feasibility: 8 | economic_leverage: 8 | testability: 8 | robustness: 7 | constraint_fit: 9 | explainability: 9
Overall: 7.9 / 10 -- Very Strong

Constraint Compliance: Compliant (constraint_fit=9)

A dedicated PCIe Gen5/CXL 3.0 accelerator card owns the entire KV-cache lifecycle--paging, INT8/FP8 compression, and attention scheduling--beside high-density LPDDR5X or HBM3 memory, offloading the memory-bound decode phase from expensive GPU HBM. This matters because decode throughput in production LLM serving is bottlenecked by KV memory bandwidth and capacity, not compute, making a purpose-built memory-side engine a natural architectural fit. Achieving 4x effective KV capacity gains through quantization and paging can double or triple served requests per GPU-dollar at scale.



Concept illustration (AI-generated). Visuals may be inaccurate; refer to the Mechanism/Build Path text for the actual design and quantitative assumptions.

Mechanism

A PCIe Gen5 x16 or CXL 3.0 accelerator card dedicated to the memory-bound decode phase of transformer inference. The card carries 64-192 GB LPDDR5X or 80-256 GB HBM3/HBM3E, plus ASIC/FPGA engines that implement paged KV-cache management, INT8/FP8 KV storage, dequantization, attention score computation, softmax, and value accumulation. KV blocks are stored in 16-64 KB pages with a hardware page table keyed by sequence/request ID and layer/head. For Llama-class models, keys/values are stored as per-head or per-channel INT8/FP8 with scale metadata, yielding roughly 2-4x effective KV capacity versus FP16/BF16, with further utilization gains from paging and compaction. During decode, host software routes Q vectors for the current token from the

GPU to the card. The card performs exact dequantized attention over resident KV pages: load compressed K/V, dequantize into internal FP16/BF16/FP32 accumulation, compute QK, apply causal/window masks, softmax, and compute weighted V output. The resulting attention output tensor is DMA/CXL-transferred back to the host GPU, which continues dense projection and MLP work. Prefill, training, GEMMs, and model weights remain on existing GPUs. The value proposition is not raw TOPS, but reduced GPU HBM pressure and higher long-context decode tokens/W when KV cache dominates memory footprint and bandwidth. The system wins only if CXL/PCIe latency, DMA synchronization, runtime scheduling, and batching overhead are hidden behind GPU-side MLP/projection compute or remain below the time saved by avoiding GPU HBM KV reads.

Why Novel

Unlike GPU-side KV compression libraries or CPU offload hacks, this card places attention computation physically adjacent to compressed KV storage, eliminating the round-trip serialization penalty and enabling hardware-managed paged eviction invisible to the host model runtime. No shipping silicon today combines CXL-attached high-density memory with on-card attention engines and a hardware KV page table in a single PCIe form factor.

Buildability

An FPGA-first prototype (Alveo U55C or AMD Versal) can validate the PCIe handoff latency and attention kernel correctness within 6-9 months before committing to ASIC tape-out, with existing HBM3 DIMM and LPDDR5X SODIMM ecosystems providing off-the-shelf memory options. The host-side shim integrates with vLLM or TensorRT-LLM's existing paged-attention APIs, reducing software integration risk significantly.

Why Feasible

* Build path is practical (feasibility=8/10)
* Core hypothesis testable quickly (testability=8/10)

Top Risks

* CXL 3.0 round-trip latency (200-400 ns) erodes per-token gains at small batch sizes where GPU idle time is insufficient to hide the handoff
* NVIDIA H200/B200 and AMD MI350 integrating on-package HBM capacity expansions could close the cost-per-KV-byte gap before ASIC tape-out completes
* KV quantization to INT8/FP8 degrades output quality on long-context or reasoning-heavy workloads, requiring per-model calibration that complicates productization

Decisive Test (MDE)

Hypothesis: Offloading decode attention KV reads to a PCIe5/CXL-attached paged-KV accelerator improves end-to-end long-context decode tokens/W by at least 30% and does not worsen p99 latency by more than 10% versus an optimized GPU-only vLLM paged-attention baseline.
Setup: Run Llama-class inference with realistic serving traffic: mixed sequence lengths from 8k-128k context, batch sizes 1-64, decode-only continuation of 256-2048 tokens, and SLO-constrained scheduling. Baseline: vLLM/SGLang on GPU with FP16/BF16 and INT8/FP8 KV variants if available. Prototype: KV pages stored in CXL/FPGA/HBM appliance; GPU sends Q per layer/token, accelerator computes attention output and DMAs result back; GPU performs remaining projections/MLP. Include all runtime overheads: page allocation, batching, DMA, synchronization, quant/dequant, and scheduler delays.
Instrumentation: Nsight Systems/Compute for GPU timeline, PCIe/CXL counters for bandwidth and latency, FPGA/CXL telemetry, host perf counters, request-level tracing for p50/p95/p99 latency, external rack or inline power meter for GPU/card/host power, accuracy harness measuring perplexity and task deltas versus FP16 KV.
Kill Criteria: Kill if full-system tokens/W gain is <20% at matched p99 latency, or if p99 latency worsens >15%, or

if PCIe/CXL transfer plus synchronization consumes more time than the GPU attention time saved for target workloads.
Rescue Pivot: If full offload fails, pivot to GPU-resident KV compression/page-management IP, a CXL KV-capacity expansion tier for very long contexts only, or a near-GPU SmartNIC/DPU scheduler that improves batching and KV placement without offloading the attention math.
Cost: $50k-$250k using rented GPUs plus FPGA/CXL appliance; $250k-$1M if custom FPGA/HBM boards and multi-GPU servers are purchased.
Time to Signal: 8-16 weeks for FPGA/CXL prototype or high-fidelity emulation; 2-4 weeks for a preliminary software-only latency budget model.

Refinement Deltas

Risks Addressed:

* GPU vendors can integrate KV compression, larger HBM, paged attention, or decode-specialized kernels directly, erasing the external-card advantage.
* PCIe/CXL latency is structurally bad for token-by-token decode, especially at small batch sizes where serving latency matters most.
* The clean separation of attention on the card and MLP on GPU may create synchronization bubbles at every layer, destroying theoretical bandwidth gains.

Red-Team:

* Challenge: GPU vendors can integrate KV compression, larger HBM, paged attention, or decode-specialized kernels directly, erasing the external-card advantage.
* Challenge: PCIe/CXL latency is structurally bad for token-by-token decode, especially at small batch sizes where serving latency matters most.
* Challenge: The clean separation of attention on the card and MLP on GPU may create synchronization bubbles at every layer, destroying theoretical bandwidth gains.
* Challenge: Accuracy and model portability are fragile: KV quantization tolerances vary by architecture, layer, head, context length, and workload.
* Challenge: Operational complexity is high: custom runtime, scheduler, page manager, accelerator firmware, failure handling, and integration with fast-moving inference stacks.

Budget Ranges

Prototype: $180,000 (2x Alveo U55C FPGA cards at $15k each, HBM3 memory modules, PCIe Gen5 host testbed, 3 FTE engineers x 6 months, EDA/simulation tooling)
Pilot: $1,200,000 (small ASIC or structured ASIC tape-out at 7nm via TSMC shuttle, 10-unit pilot boards, bring-up lab, 6 FTE x 12 months, vLLM integration and benchmarking)
Scale: $8,000,000 (full 5nm ASIC tape-out with HBM3E PHY, 500-unit production run, driver/SDK hardening, cloud pilot partnerships, sales and certification overhead)

30/60/90 Plan

Day 30: Complete PCIe Gen5 latency characterization on FPGA testbed with synthetic Q-vector handoff microbenchmarks; define KV page table hardware schema and INT8 dequantization pipeline spec; establish baseline decode throughput on Llama-3-70B at batch sizes 1, 8, 32 on GPU-only reference
Day 60: Functional FPGA prototype running single-layer paged attention with INT8 KV storage end-to-end; measure actual tokens/sec and tokens/watt versus GPU baseline at batch-8; identify top three latency bottlenecks in host-to-card Q routing path and publish internal feasibility report
Day 90: Full multi-layer decode loop validated on FPGA with vLLM paged-attention shim integration; demonstrate 2x+ effective KV capacity versus FP16 baseline with less than 0.5 perplexity regression on standard benchmarks; go/no-go decision on ASIC structured tape-out with vendor shortlist and cost model locked
Dominance Axis: faster_validate -- It is compelling only if full decode tokens per watt improve after PCIe/CXL

handoff and scheduling costs.

Top Uncertainties & Assumptions

* Compiler spill risk: auto-scheduling tools may fail to tile layers optimally, forcing DRAM spills that collapse the efficiency advantage for irregular or fused-attention models
* Quantization fit risk: INT4 weight unpacking degrades accuracy on models not explicitly quantization-aware trained, limiting addressable customer model library at launch
* Die cost risk: 48 MB SRAM dominates area and raises per-unit cost to $18-30 at low volumes, making the BOM uncompetitive against merchant NPU modules until volumes exceed ~500K units
* Quality regression on retrieval-sensitive tasks (multi-hop reasoning, needle-in-haystack) where top-k misses critical tokens and fallback triggers too late
* Model fine-tuning dependency creates a two-sided adoption barrier -- chip needs models, models need chip -- stalling the go-to-market flywheel