CMOS+X: Stacking Memories based on Oxide Transistors upon GPGPU Platforms

Faaiq Waqar
Georgia Institute of Technology
Atlanta, GA, USA
faaiq.waqar@gatech.edu

Ming-Yen Lee
Georgia Institute of Technology
Atlanta, GA, USA
mlee838@gatech.edu

Seongwon Yoon
Georgia Institute of Technology
Atlanta, GA, USA
gabrielyoon@gatech.edu

Seongkwang Lim
Georgia Institute of Technology
Atlanta, GA, USA
s_kwang.lim@gatech.edu

Shimeng Yu
Georgia Institute of Technology
Atlanta, GA, USA
shimeng.yu@ece.gatech.edu

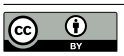
Abstract

In contemporary general-purpose graphics processing units (GPGPUs), the continued increase in raw arithmetic throughput is constrained by the capabilities of the register file (single-cycle) and last-level cache (high bandwidth), which require the delivery of operands at a cadence demanded by wide single-instruction multiple-data (SIMD) lanes. Enhancing the capacity, density, or bandwidth of these memories can unlock substantial performance gains; however, the recent stagnation of SRAM bit-cell scaling leads to inequivalent losses in compute density.

To address the challenges posed by SRAM's scaling and leakage power consumption, this paper explores the potential CMOS+X integration of amorphous oxide semiconductor (AOS) transistors in capacitive, persistent memory topologies (e.g., 1T1C eDRAM, 2T0C/3T0C Gain Cell) as alternative cells in multi-ported and high-bandwidth banked GPGPU memories. A detailed study of the density and energy tradeoffs of back-end-of-line (BEOL) integrated memories utilizing monolithic 3D (M3D)-integrated multiplexed arrays is conducted, while accounting for the macro-level limitations of integrating AOS candidate structures proposed by the device community —an aspect often overlooked in prior work. By exploiting the short lifetime of register operands, we propose a multi-ported AOS gain-cell capable of delivering 3× the read ports in 76% of the footprint of SRAM with >70% lower standby power, enabling enhancements to compute capacity, such as larger warp sizes or processor counts. Benchmarks run on a validated NVIDIA Ampere-class GPU model, using a modified version of Accel-Sim, demonstrate improvements of up to 5.2× the performance per watt and an average 8% higher geometric mean instruction per cycle (IPC) on various compute- and memory-bound tasks.

CCS Concepts

• **Hardware → Memory and dense storage**; **Analysis and design of emerging devices and systems**; • **Computer systems organization → Parallel architectures**.



MemSys '25, Washington, DC, USA

ACM ISBN 979-8-4007-2002-4/25/10
https://doi.org/10.1145/3767110.3767120

Keywords

GPGPU, Oxide Semiconductors, Back-End-of-Line Integration, Multi-Ported Memory, Cache Memory, Embedded DRAM

ACM Reference Format:
Faaiq Waqar, Ming-Yen Lee, Seongwon Yoon, Seongkwang Lim, and Shimeng Yu. 2025. CMOS+X: Stacking Memories based on Oxide Transistors upon GPGPU Platforms. In *International Symposium on Memory Systems (MemSys '25), October 07–08, 2025, Washington, DC, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3767110.3767120

1 Introduction

Graphics processing units (GPUs) have become pivotal to server scaling, as noted by NVIDIA's 427% increase in data-center revenue from FY24 to FY25 [50]. Though seminal engineering efforts of the GPU targeted the acceleration of raster graphics and video workloads [75] (as their namesake implies), their combination of high-bandwidth Single-Instruction-Multiple-Data (SIMD) execution units (stream multiprocessors, SMs) and on-chip memories have proved ideal for highly parallelized processing of numerically intensive floating-point arithmetic [17], leading to their ubiquity in the processing of modern AI/ML training and inference and large-scale scientific computing.

The performance of a GPU is often bottlenecked by the available bandwidth, capacity, and die area of on-chip memory subsystems [68]. In workloads with low arithmetic intensity, SMs frequently idle while awaiting memory accesses during execution, resulting in underutilization of execution units [13]. Scaling the SM count intensifies contention for the shared L2 cache, while the number of resident warps per SM is capped by the size of its register file (RF) and the per-thread register allocation. Raising the warps-per-SM budget within a fixed die area, therefore, trades off against the total number of SMs that can be integrated [1]. Since modern GPUs support many registers per thread, the compiler directs operand spillover into the L1 data cache (L1D) when register file capacity is exhausted, imposing increased traffic and occupancy pressure on the slower L1D [19]. Tasks such as *backpropagation* and *blackscholes*, which demonstrate low intra-warp divergence, benefit from higher threads per warp; however, the number of operands demanded per cycle (and hence the required ports or banks) scales linearly with the warp size [45]. The imposition of these memory requirements has inevitably led to the sharp rise in cumulative memory capacity (especially that of the register files and the L2) in modern GPUs

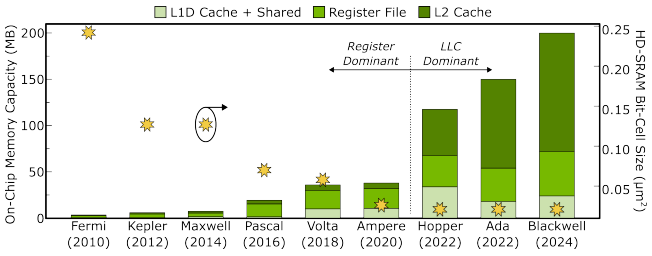


Figure 1: Scaling of GPU shared memory, registers, L1, and last level cache (LLC) vs. corresponding SRAM cell in NVIDIA architectures

by over two orders of magnitude from NVIDIA's Fermi (2010) to Blackwell (2024) architectures [51], while the density of the high density SRAM (HD-SRAM) bit cell has only climbed by one order of magnitude (Fig. 1). This disparity is the key indication that **GPU memory scaling is driven by architectural demand, not fundamental SRAM densification**, and as such, is limited by SRAM technology scaling constraints.

Recent advances in semiconductor fabrication have enabled the integration of active devices, such as non-volatile memories, in the back-end-of-line (BEOL) stack. By relying on low-temperature (<400 °C) deposition steps, multiple tiers of memory can be fabricated above the front-end-of-line (FEOL) logic without degrading the underlying CMOS transistors [61]. Among the most promising options are amorphous-oxide-semiconductor (AOS) transistors, which combine ultra-low off-state leakage (< fA/μm; three to four orders of magnitude lower than Si MOSFETs) with adequate electron mobility (~ 20 $cm^2/V{\cdot}s$) [76]. Leading research institutes, such as IMEC, have demonstrated functional prototypes based on AOS 2T0C arrays [64], and major foundries, including TSMC, have also reported AOS 1T1C macros with high yield [8]. Prior modelling studies show that AOS 2T0C memories can outperform SRAM in TPU buffers [40], boost energy efficiency in digital compute-in-memory (DCIM) accelerators [38], [32] and deliver ~4.5× higher density (at 256 MB) with nominal performance improvements when deployed as a shared CPU L3 cache [73]. We hypothesize that, in gain-cell configurations (§4.2, §5) with small storage node capacitance and decoupled read/write paths, the speed of AOS gain cell memories may be sufficient to serve as single-cycle register memories in GPUs with *lower base clock frequency* than their CPU counterparts [11]. Moreover, to meet the needs of memory-bound tasks, several AOS-based candidates, such as BEOL-compatible 1T1C eDRAM and 2T0C/3T0C gain cell topologies, offer a practical avenue to enhance the overall bandwidth and density of the large shared L2. To understand the viability of these respective memory candidates, a critical evaluation of their achievable density, bandwidth, and energy efficiency is required under constrained design exploration (e.g., imposed by sneak-path currents, increased parasitic capacitance, IR drop, lower mobility), which must be well-characterized. In performing this, this paper makes the following contributions:

- Using Accel-Sim [28], we evaluate the lifetime of operands in GPU register files to determine the retention requirements of register memories. Furthermore, we propose a stacked multi-read port AOS gain-cell (NT0C) that enables 3× the read ports in ~76% of the footprint of a comparative 8T-SRAM bank.
- Using NS-cache [73], we analyze the limitations in AOS 2T0C array scaling under sneak path and IR drop constraints studied in SPICE, 1T1C access time and sense margin tradeoffs, and 3T0C gate loading and leakage under read path threshold voltage. We demonstrate that (1) the f_{max} limitations on peak bandwidth of AOS 1T1C banks are easily overcome through increased partitioning, (2) the AOS 3T0C speed vs. leakage tradeoffs make it unfavorable for high-speed cache.
- We evaluate highly banked AOS L2 caches at iso-footprint in a modified version of Accel-Sim integrated on a verified NVIDIA Ampere RTX 3070 model. Our benchmarking reveals that AOS 1T1C integration can enable up to ~5.1× the performance per watt and ~6.1× memory density over a baseline HD-SRAM L2 cache.

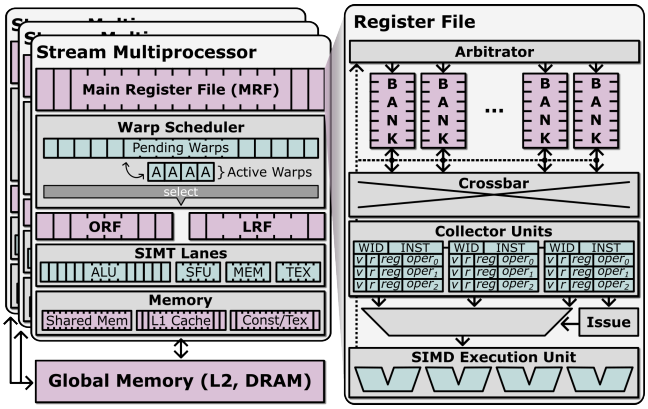


Figure 2: Organization of stream multiprocessor (SM), register files, and global memories

2 Background

2.1 GPGPU Organization

Modern standalone GPUs comprise tens to hundreds of streaming multiprocessors (SMs), each hosting a programmer-controlled shared memory, private level 1 data (L1D) cache, register files (operand, last result, and main), and wide Single-Instruction-Multiple-Thread (SIMT) execution lanes containing arithmetic (ALU) and special-function units (SFU) (Fig. 2). When a kernel is launched, groups of threads in cooperative-thread arrays (CTA) or thread blocks are mapped to SMs, which are then partitioned into warps. Warps are time-multiplexed by the scheduler and executed in lock-step. To feed these wide SIMT lanes, each SM's register file contains tens of thousands of registers (~65k/SM in Blackwell), which are highly banked and, in smaller RFs, multi-ported (e.g., NVIDIA's Fermi architecture used 3-read/1-write (3R1W) in its operand register file (ORF) [19]).

Swaths of on-chip data are interleaved across LLC banks (slices), which are distributed across numerous memory partitions (Fig. 3). The memory partitions are connected to SMs through an on-chip interconnection network; each partition contains L2 slices, request schedulers, and a memory controller for off-chip DRAM (GDDR6X or HBM) [3]. In Ampere-class GPUs, there are eight partitions, each

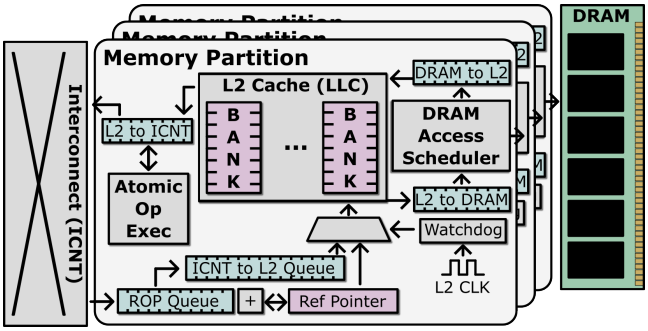


Figure 3: Organization of a memory partition, containing L2 slices, and refresh timing model incorporated in Accel-Sim

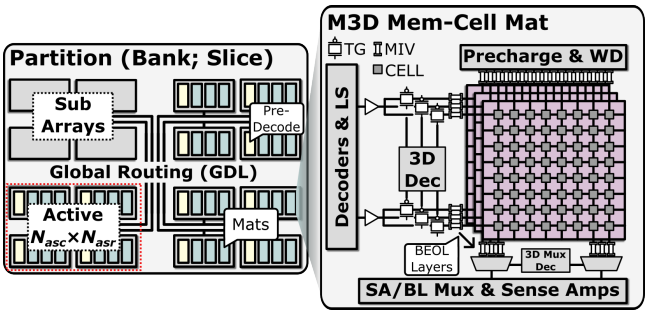


Figure 4: Organization of a monolithic-3D (M3D) bank/slice. Active subarray rows (N_{asr}) and columns (N_{asc}) often called a sub-bank

housing two L2 banks [49]. The L2 employs a write-back policy with respect to global memory [62].

2.2 Organization of a Memory Bank

Memory banks (slices) are topologically organized into a matrix of subarrays interconnected by global routes that comprise address, broadcast, and distributed data lanes [59]. These global routes, sometimes referred to as the global data line (GDL), often employ an H-Tree routing topology in RC-based memory simulators. When a transaction is received, it is routed over the GDL to a set of 'active' subarrays within each column (N_{asc}) and row (N_{asr}), which each deliver a sub-block of the aggregate block size of the bank. In some contexts, such as the nomenclature adopted by CACTI [74], this group of concurrently activated subarrays ($N_{asc} \times N_{asr}$) is referred to as a sub-bank. The number of transactions that can be issued to a bank per cycle is limited by the number of ports (N_p); however, with pipelining, it is possible to operate sub-banks in parallel if their activations are tracked, though they cannot be issued concurrently if $N_p = 1$. Each subarray is composed of a pre-decoder and a set of mats, which operate concurrently. Based on the operation concurrency, the maximum bandwidth (BW_B) for the bank with split read/write paths in a uniform cache access (UCA) model can be approximated as:

$$BW_B \approx N_p \times \max\left(\frac{W_{\text{Block}}}{t_{\text{precharge}} + t_{\text{mat,read}}}, \frac{W_{\text{Block}}}{t_{\text{mat,write}}}\right) \quad (1)$$

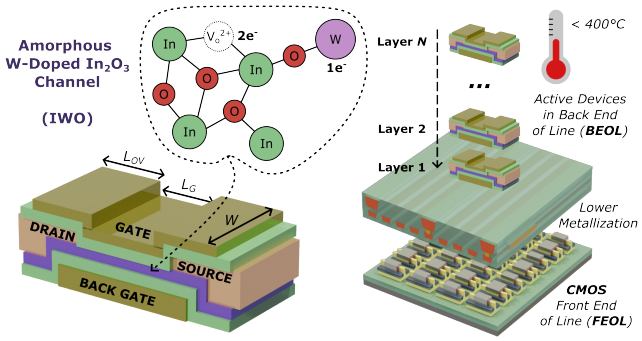


Figure 5: Double-gated IWO transistor geometry, active monolithic 3D integration of devices above the FEOL

Where W_{Block} is the block width per bank, and $t_{\text{precharge}}$ is the precharge latency. This dependence on mat latency makes cell access time a critical memory parameter, as discussed in §5.2. A mat contains an array of memory cells, and local peripherals used to drive data in and out of the array, typically decoders, pre-chargers, write drivers, sense-amplifiers (SAs), multiplexers (bitline, sense-amp), and level-shifters (when high-voltage swings are employed). We adopt a monolithic 3D (M3D) mat design in which each level is individually accessed using a 3D decoder that drives a set of transmission gates, allowing access to a specified level in the BEOL memory through the decoder/level-shifter drivers (Fig. 4). This 3D-decoded access scheme integrates a level-multiplexer and mux-decoder that allows BL sharing of FEOL SAs. In an M3D design, peripheral circuits are tucked under the memory array and connected by BEOL MIVs, which offer higher I/O density than TSVs [38].

2.3 Oxide Semiconductor Transistors in the BEOL

Amorphous oxide semiconductors (AOS) such as indium oxide (In_2O_3) are a class of semiconducting oxides that exhibit moderate electron mobility (~20 $cm^2/V{\cdot}s$), in which conduction is primarily governed by donor-like defects (N_D) such as oxygen vacancies. Dopants such as germanium (Ge), tin (Sn), or tungsten (W) are used when In_2O_3 is employed as a channel material to curb the formation of defects, thereby improving its stability and increasing its threshold voltage (V_t) [57]. Leading foundries (TSMC, Samsung) and research institutes (IMEC) are actively working on AOS channel materials for the integration of BEOL memories [70],[4],[56]. Demonstrations of stacked AOS transistors have been characterized in up to ten monolithic tiers and scaled to 10 nm gate length (L_g) [78],[16].

3 Simulation Methodology

To build a cohesive evaluation of the design, technology, and system-level integration of BEOL-compatible AOS memories (Fig. 5), we employ a precise quantitative study that utilizes finite-element physical models, SPICE simulation, and cycle-accurate GPU simulation. Modeling of lab-measured double-gated (DG) long-channel W-doped In_2O_3 (IWO) transistors is performed in Sentaurus Technology CAD (TCAD), from which a scaled 7 nm technology model is

developed (L_g = 15 nm, L_{ov} = 30 nm), and measured (I_d-V_{ds}, I_d-V_{gs}, and $C_{gg}/C_{gd}/C_{gs}$ parameters) for varying donor-defect densities (N_D). Extracted parameters are utilized to develop ML-assisted compact models [9] used in subsequent SPICE simulation, while Si-CMOS reference circuits are built with the ASAP7 PDK [12]. NS-Cache [73] is utilized to conduct an exhaustive search of the power, performance, and area (PPA) of various bank and subarray configurations presented in this work, using 7 nm FinFET predictive libraries. A baseline HD-SRAM cell of 0.0262 μm^2 is used based on the advanced foundry 7 nm platform technology [10]. Accel-Sim/GPGPU-Sim's verified NVIDIA Ampere RTX3070 model is used for system performance evaluation. We extend Accel-Sim's memory-partition timing to incorporate AOS refresh overheads, enabling cycle-accurate assessment of their impact on kernel execution (Fig. 3).

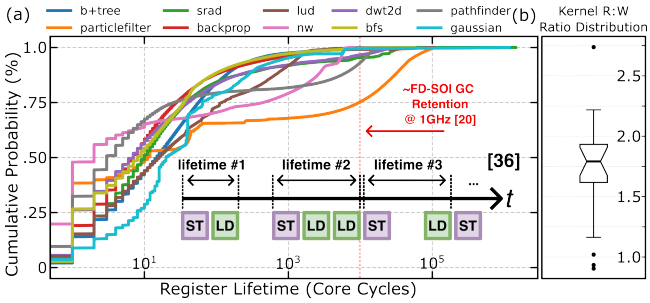


Figure 6: Measured reg lifetimes on select Rodinia [7] benchmarks. Nearly all regs are limited to a 10^5-cycle boundary (b) register read:write ratio distribution tracked per kernel

4 Scaling Register Files, CTAs, and Warps

4.1 On the Lifetime of Operands

The GPU register file must be able to deliver up to two source operands and one destination per thread for each warp-wide instruction within a single cycle (~1 ns in NVIDIA Volta to Hopper architectures) [43]. Consequently, the underlying register memories must (1) operate at high speed (~sub-nanosecond), and (2) be heavily banked or multi-ported to service multiple read/write requests concurrently. In some cases, (2) is taken to an extreme, as illustrated by the Intel Itanium microprocessor's 12R10W register file [18]. Capacitive memories that require a refresh operation seem ill-suited as a register memory, as refresh operations temporarily remove a bank from service, thereby hindering the high-speed requirement. Nevertheless, this has not impeded the proposal of eDRAM register memories in the literature, as seen in Si 3T1D [27] and FD-SOI 4T0C [20] cells. GainSight [36] argues that the key to understanding a persistent memory's suitability for a level in the memory hierarchy lies in the lifetime of data blocks (i.e., how long data is retained and utilized). Using Accel-Sim, we track the lifetime of registers in each SM, as defined by GainSight (Fig. 6a), on a set of randomly selected benchmarks from Rodinia [7]. We find that, though operands have a low median lifetime (2-20 cycles), over 99% of register operands are overwritten or evicted within 10^5 cycles (~100 μs on a 1 GHz SM clock) when accounting for the wide tail of the distribution. Although this may be beyond the retention of a gain cell on an Si or FD-SOI platform (~6 μs) [20], it is far below the achievable retention times in AOS gain cells, which enable retention times of (milli)seconds through low leakage.

4.2 Alternative Paths to Multi-Porting

Multi-porting, which enables multiple accesses per cycle to a memory bank, is often employed in register files in both multi-core CPUs and GPUs. Multi-porting may be realized through several means: banks can be replicated [33], time-multiplexed on a higher frequency clock domain [25], or virtually multi-ported through banking (at the cost of bank stalls during contention). The following study places particular emphasis on cell-level multi-porting, as (1) the speed of AOS memories is lower than Si (lower mobility), (2) the read and write paths in a gain cell are intrinsically bifurcated, and (3) leveraging M3D-stacking opens the opportunity for compact means of multi-porting layout. In SRAM, cell-level multi-porting (MP) can quickly become unfavorable due to interconnect congestion and increased transistor counts (~4T per write port and 2T per read port) that exponentially inflate cell size (~O(N_P^2)) [48]; nevertheless, cell-level MP is still employed in CPUs with modest register counts to meet high speed single-cycle latency targets, (e.g., IBM Power & Intel Itanium [18],[34]). Because the GPU ORF employs additional read ports, and kernel register accesses are read-heavy (Fig. 6b), we focus on read multi-porting. Although additional write ports in a gain cell only require one transistor, the overhead of level shifting on write paths [73] imposes significant area penalties for duplicated periphery.

In Fig. 7a, we illustrate a conventional SRAM with split read (2T) and write paths (bidirectional), extrapolated from a 10T-derived SRAM design in [38], and our proposed AOS (DG-IWO) gain cell with NR1W ports. Fig. 7c and 7d illustrate the layout of each cell. For 7 nm BEOL design rules, we adopt the N7 metal-x (M_x, 40 nm) and metal-y (M_y, 76 nm) metallization pitches discussed in [47], an MIV pitch (60 nm) discussed in [41], as well as the CPP (54 nm) and fin pitch (27 nm) matching the foundry's 7 nm platform technology [10]. We assume the utilization of up to five M_x and five M_y layers. Because the number of stacked tiers in the AOS gain cell scales with the port count, Fig. 7c depicts abstract "functional" metal layers, and metallization limits are analyzed in further sections. Under these rules, we estimate that an 8T (1R1W) SRAM cell consumes 33.2% more area than a 6T SRAM cell, closely resembling the ~30% footprint increase observed in IBM's 65 nm PD-SOI process [6]. In both cells, the transconductance of a read transistor (R_i) is used to sink current from the pre-charged read-bitline (RBL) based on the stored value (*SN* in the gain cell, *Q* in the SRAM). For the gain cell, this operation requires applying a differential voltage to the V_{DS} of the selected cell by driving the RWL to V_{SS}. In contrast, in SRAM, a pulse of Vdd is applied to the gate of the read-gating (R_G) transistor, allowing the current to sink (Fig. 7b). A challenge posed by each read port in SRAM is that each read path adds additional subthreshold leakage from the precharged RBL, as a potential difference exists across each 2T read path. On the contrary, both the RWL and RBL (attached at the read port source and drain) are peripherally controlled in the 1T read path of the MP gain-cell; thus, leakage is suppressed, and static power can be derived from the cell retention:

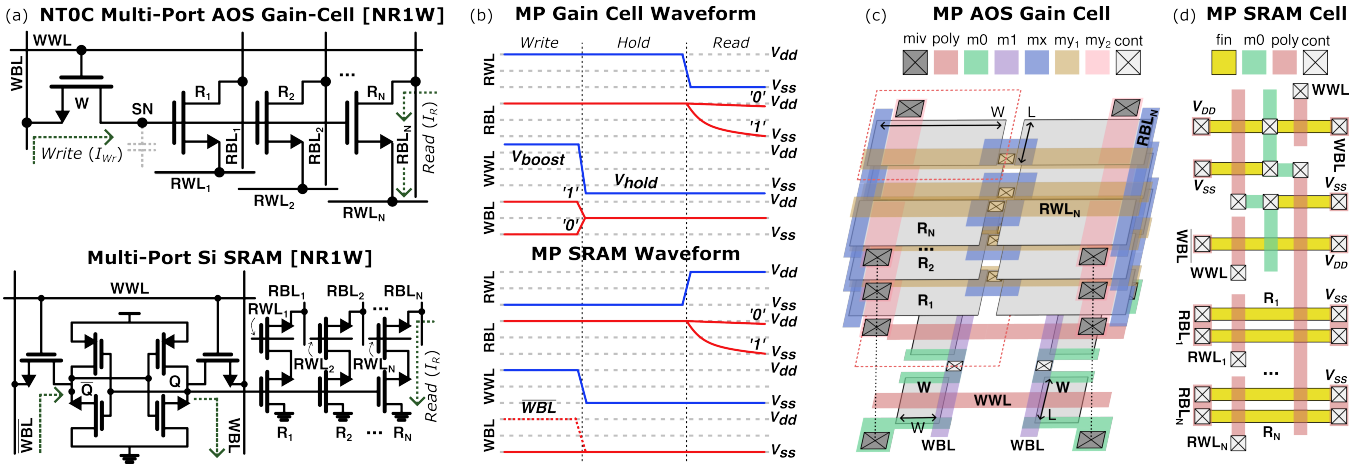


Figure 7: (a) Multi-ported AOS gain-cell (GC) and SRAM schematics, (b) cell operation diagrams, (c)-(d) physical layout and connectivity

$$P_{\text{static}} \approx \frac{C_{\text{SN}}\,\Delta V_{\text{SN}}^2}{t_{\text{ret}}};\text{NT0C, 1T Read Port} \tag{2}$$

Where C_{SN} is the storage node capacitance, ΔV_{SN} is the change in stored voltage before a refresh is issued, and t_{ret} is the retention period. In [73], the trade-off between retention and access time is discussed in an IWO 2T0C using the V_t, and the hold/boost voltage (V_{hold}, V_{boost}), following which we study parameters of a cell with an optimized R_i width (W_{RA}) of 150 nm, a write transistor (W) with a nominal width (W_{WA}) of 30 nm, aiming for a write-access time (t_{wr}) of 400 ps, t_{ret} of 10 ms (~100× requirement in §4.1), V_{hold} of -0.4 V and V_{boost} of 1.2 V. The C_{gg} of $\sum R_i$ dominates C_{SN}, and t_{wr} is inversely proportional to write current (I_{wr}), which scales with W_{WA}. Therefore, to preserve high write speed as read ports (N_{PR}) are added, W_{WA} is widened in proportion to N_{PR}, increasing both cell size and static power (Fig. 8a and 8b). Additionally, since the pitch of the upper metallization is relaxed (~18× the pitch of M_y), cell stacking reaches a ceiling with $N_{\text{PR}} = 3$, resulting in a sharp increase in footprint as two read transistors are integrated into each level to minimize cell area when the number of fine-pitch metallization tiers becomes a limitation. Nevertheless, we find that an NR1W AOS gain cell consumes over four orders of magnitude less standby power than SRAM, while occupying a smaller footprint.

An additional benefit of multi-porting AOS gain-cells can be elucidated from the theory presented for split-gated AOS transistors [58]. It is posited that the capacitive coupling phenomenon in AOS 2T0C gain cells, which refers to the modulation of the storage node voltage (V_{SN}) caused by pulsing the BL/WL voltage, is proportional to the ratio of all parasitically coupled capacitances to the storage node, imposed by the charge neutrality condition. This modulation reduces the sense margin and heightens read disturbance. However, as N_{PR} is increased, so too is the number of capacitances coupled to the SN, leading to a theoretical 4× reduction in read capacitive coupling in 4R1W over 1R1W:

$$\begin{aligned}\Delta V_{\text{SN}} &\propto \frac{W_{\text{WA}}}{W_{\text{WA}} + 2W_{\text{RA}}N_{PR}}\Delta V_{\text{WWL}} \\ &= \frac{W_{\text{RA}}}{W_{\text{WA}} + 2W_{\text{RA}}N_{PR}}\Delta V_{\text{RWL}}\end{aligned} \tag{3}$$

In Fig. 8c, we plot the theoretical reduction in RBL/RWL coupling and the simulated coupling measured in SPICE as a function of the number of read ports. In simulation, the reduction measured in 4R1W is closer to 3×. Coupled with a split-gated AOS transistor geometry, the multi-porting of AOS gain cells provides an avenue for suppressing capacitive coupling without requiring high secondary gate bias voltages.

4.3 Macro-Level Performance, Power, and Area

Before evaluating the performance of NR1W cells at the bank level, we first examine the intrinsic cell-level limitations imposed by the single-transistor (1T) read port of the AOS gain cell, which places constraints on the allowable memory array dimensions. In the following, a subscript s denotes selected lines/cells, and a subscript u denotes unselected lines/cells in the same port. (1) IR Drop: Because the RWL_s is used to sink current from RBL_s when the SN stores a '1', the current must travel through the entire RWL_s to the driver (sink), creating a voltage divider effect that reduces the V_{gs} of Rs,

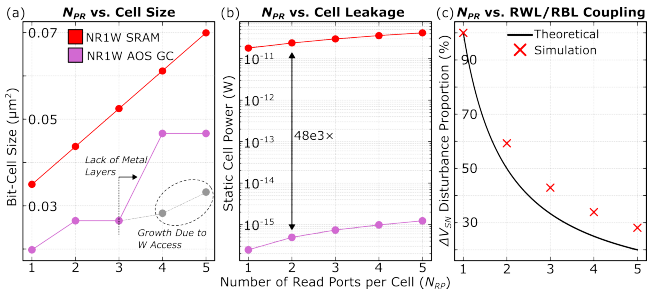


Figure 8: (a) Cell size of MP memories, (b) MP cell standby power, and (c) MP AOS RWL/RBL coupling, as a function of read ports

thus limiting worst-case read speed as the number of columns (N_{col}) increases. (2) Sneak Path: During the read-out of a '1', the RBL_s discharges over R_s onto RWL_s; however, each RWL_u is held at Vdd, thus leading to a reverse polarity ΔV_{ds} over each R_u, which in the worst case (i.e., when each SN stores '1') causes RBL_s to prematurely settle before a sufficient read margin (RM) can develop (Fig. 9a), thereby limiting the number of rows (N_{row}). To quantify these effects in the upper limit, we simulate the read port of a 5R1W cell (for sizing) with an IR drop imposed by N_{col} = 512 over varying N_{row} and V_t, in the worst-case column data pattern of all '1's (Fig. 9b). Read margin is measured as the V_{RBL} drop between '0' and '1' stored at the SN; we track (i) the peak RM, (ii) the time at which RM saturates and (iii) if/when the RM crosses the 200 mV level required for a 100 mV sense threshold. In the literature, the conventional guidance for architecting an AOS GC read port is to use a lower V_t than the write port counterpart, since this will increase sink current for an increased C_{RBL} (§5.2.1) [73],[52],[55]. However, we refine this view: in cases with very few rows (i.e., N_{row} = 64), a lower V_t does accelerate read speed. However, as N_{row} is increased, this relationship quickly becomes parabolic, and a higher V_t is required to curb read failure caused by sneak path leveling. From this, we set a maximum N_{row} of 64 (extendible to 128 if a folded BL layout is used [65]) as the upper boundary.

In NVIDIA's Fermi architecture, each MRF bank has a capacity of 8kB, each 16B wide with 32b per register [19]. We utilize the findings from the prior section in NS-Cache to model 8kB multi-ported SRAM and AOS gain-cell banks with a 128-bit W_{Block}. We constrain the random cycle time (RCT) of each bank to 750 ps and exhaustively sweep design points with up to 8×8 subarrays per bank and mats per subarray, as well as up to 2 M3D integrated tiers of memory (N_L). The outcomes (area, static power, and dynamic energy consumption) as a function of N_{PR} are plotted in Fig. 10a, for the minimum area 8 kB bank configuration under timing and sizing constraints. With a single BEOL tier, an AOS 2R1W bank can be placed in relatively the same footprint of an equivalent 1R1W 8T SRAM bank, and under the case that two tiers of memory can be integrated, a 3R1W AOS GC bank using cell-level multi-porting

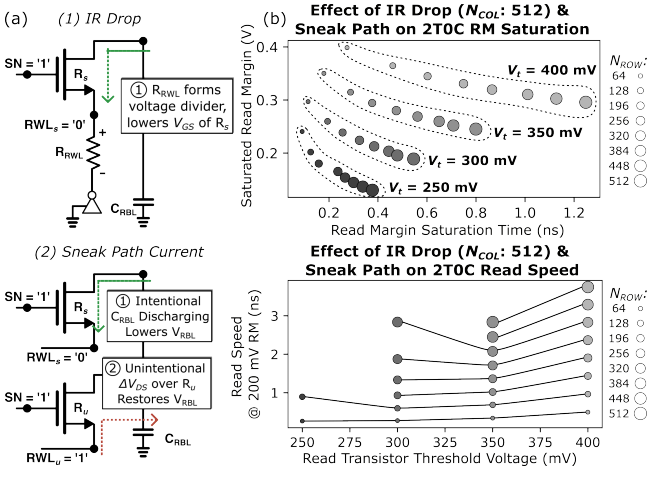


Figure 9: (a) Illustration of IR drop and sneak path challenges using single AOS transistor port, (b) Effects on read margin and delay

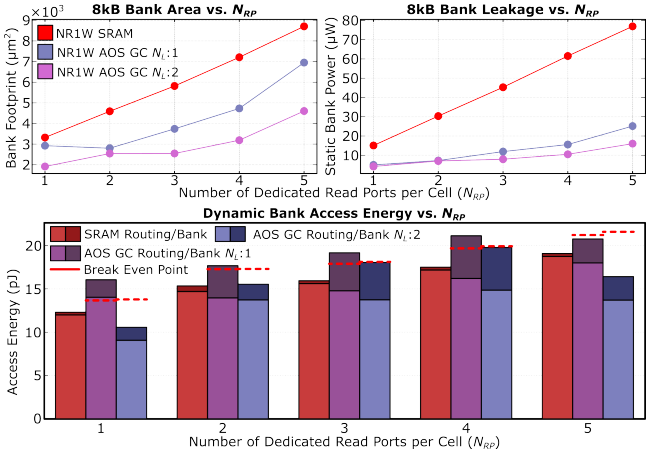


Figure 10: (a) bank area comparison, (b) static power comparison, and (c) dynamic write access energy breakdown of MP memories

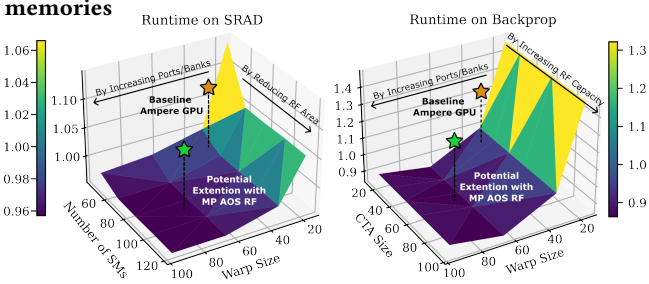


Figure 11: Warp, CTA, and SM scaling on compute-intensive workloads and translation using MP AOS GC integration normalized to NVIDIA Ampere RTX3070 baseline (Table 2)

can integrated into the space if a 1R1W SRAM bank at ~0.76× in footprint without sacrificing RCT. Reductions in static power are not as striking as at the cell level, owing to the observation in [73] that increased capacitance of AOS devices necessitates larger drivers to optimize latency, thus increasing the leakage of peripheral circuits, however the static power can be dramatically reduced by 72.3%-79.1% over SRAM, significant considering the static power of register files is considered to consumer ~17% of SM static power in NVIDIA's Volta GPUs [21]. The key drawback, however, is that the dynamic energy consumption, especially for write operations in the AOS gain-cell, is often higher than that of SRAM, even when including the inter-bank-level routing power. This increase is attributed to the higher WWL switching energy ($\propto C_{\text{WWL}} \times \Delta V_{\text{WWL}}^2$) and constraints on the mat size, resulting in higher subarray activations and, consequently, higher mat-level access energy per operation (~5× that of SRAM with 1R1W). We observe that this access energy is reduced using stacking, as inactivated subarrays can be stacked on top of one another, leading to shorter routing. Prior work has reported that the high-access frequency of register files yields a ratio of ~6:1 in dynamic to static power consumption [42]. From this simplification, one can estimate that if total leakage is diminished, the increase in dynamic operation energy required to maintain the same overall power is ~16.6%, as the bank static power reduction approaches 100%. We plot this 'break-even point'

in Fig. 10c alongside the dynamic energy consumption as a target, showing that in 60% of maximized density cases, the M3D AOS gain-cell bank maintains or improves overall register file power while improving footprint and portability.

To understand the implications on system performance, we utilize Accel-Sim's PTX mode to trace benchmarks that benefit from increased CTA size (i.e., warps per SM), increased SM count, and larger warp size (i.e., threads per warp). Each of these, respectively, is bound in some manner to the register file (i.e., by capacity, area, and bandwidth). For example, if a 3R1W device were adopted in place of an 8T SRAM, warp sizes of up to 96 threads (3× the NVIDIA Ampere baseline) may be enabled. If a 2R1W were used to halve the number of banks, the register file area per SM would be reduced by over 2×, allowing the integration of more SMs. Even if a simple 1R1W cell is adopted, the size advantages could be leveraged to increase capacity, thereby doubling the CTA size. In Fig. 11, we demonstrate that arithmetic-intensive applications, such as *backpropagation* and *srad* from the Rodinia suite runtime, can be reduced by ~10% by leveraging the density and portability of AOS gain-cell register files.

5 Scaling the LLC (L2)

The GPU last level cache (LLC. i.e., the unified L2 in NVIDIA GPUs) reduces pressure on DRAM channels and hides long off-chip memory latencies that would otherwise stall SIMT execution pipelines [1]. Since the shift to post-Ampere architectures, the L2 cache has increased rapidly from 6MB to 120MB (Blackwell), primarily due to the data-hungry demands of data center applications, such as large language model (LLM) training and inference [46]. Nevertheless, one may (naively) assume that this points to the need for greater capacity at all costs; however, the relationship between workloads and the LLC is not so simple. Generally, memory-bound problems dependent on the L2 can be classified into one of three categories: (1) *Capacity-limited* problems, such as large matrix operations, where the kernel's working set exceeds the effective LLC capacity [79]. (2) *Bandwidth-limited* problems, such as database systems in which query-execution time is bound by the peak L2 bandwidth [5]. (3) *Latency-limited* workloads with irregular or fine-grained control flows, such as graph analytics, in which time to a hit is critical due to their data-dependent dynamic behavior [53].

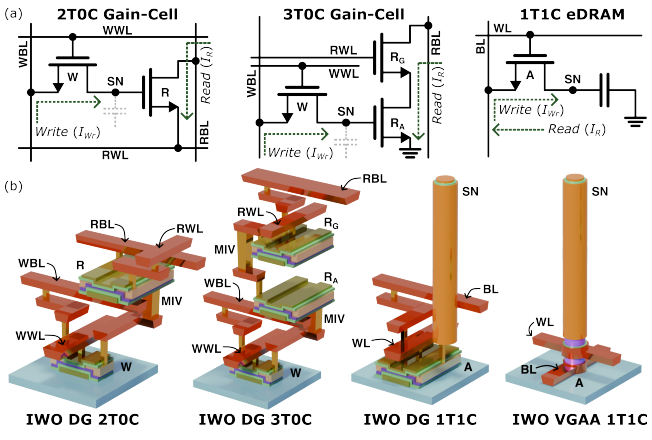


Figure 12: (a) AOS capacitive memory schematics and operating currents (b) 3D illustration of M3D integrated bit-cells, including double gated (DG) and vertical gate-all-around (VGAA) designs

From a designer's perspective of AOS LLCs, addressing each of these challenges involves leveraging the intrinsic spatial density of M3D design. Because the AOS LLC area, and thus capacity, is tightly bound to the area of peripherals, the largest array size that supports a target sub-bank RCT should be considered to minimize the number of duplicate mats per subarray and the ratio between FEOL and BEOL active footprints. It is worthwhile to target lower operating voltages and reductions in parasitic capacitance, as these qualities increase peripheral size due to bulky I/O level-shifting and large driver sizes [73]. As discussed in §2.2, the cache bandwidth ceiling is bound by the mat latency (and thus cell latency), the number of ports, and the number of banks. As will be discussed in the following subsections, the limitations of AOS devices (i.e., lower current density and parasitic capacitance) may increase minimum tmat, thus lowering the bandwidth ceiling per bank. Hence, adjusting to increasing the number of banks is critical to improve bandwidth, while understanding that: (1) cache fragmentation leads to increases in tag addressing widths, and thus tag capacity [60], (2) increasing banks inflates routing complexity, which, unless offset by smaller banks, dominates total L2 latency [30].

5.1 Comparison of AOS Cache Candidates

As discussed in depth as part of the constraints on array sizing in §4.3, the overall density and static power efficiency of AOS 2T0C integration are limited by a set of intrinsic (i.e., sneak path current, IR drop, small C_{SN}) and extrinsic (i.e., large buffers, level-shifter overhead, split-peripherals) factors. Thus, it is worthwhile to concurrently study alternative cell topologies using AOS devices that remedy some of the shortcomings of the 2T0C gain cell, while investigating the extent to which the relaxed timing requirements of the shared GPU LLC may mitigate the shortcomings of AOS 2T0C. We consider two alternative topologies for the following ablation studies in this work: (1) the 3T0C gain-cell, which adds an additional read-gating transistor (RG) and shifts the RWL to the read control gate (the same read-operation principle as the 8T SRAM). (2) a BEOL-compatible 1T1C eDRAM, of which the access transistor (A) is an AOS device and incorporates a dedicated stacked capacitor as the C_{SN} [8]. The M3D layouts and operational principles of the three proposed topologies are illustrated in Fig. 12. The contact sharing assumptions, cell sizes, and technological advantages and limitations are summarized in Table I. The remainder of this subsection discusses the tradeoffs of AOS 3T0C and 1T1C designs in terms of array sizing before proceeding to PPA comparisons at the macro level.

5.1.1 Leakage and Speed in AOS 3T0C. In §4.2, we briefly note that a shortcoming of the 2-transistor (2T) readout port in 8T SRAM is the increased static power of the cell, a function of the potential difference created by the pre-charged RBL and V_{ss} connection over the read port. To achieve high speed in 8T SRAM (where differential BL sensing isn't used), the V_t and fin count (thus, W_{eff}) of the read gating transistor (RG) and the read access transistor (RA) are modulated [6]. Though lowering the V_t improves speed (higher V_{ov}), it increases the leakage ~exponentially in tandem. The schematic equivalence of the 2T read port in an AOS 3T0C lends itself to this

Table 1: AOS Cell Topology Assumptions in 7 nm Platform

Cell Topology	Contact		Shared	Cell Size	Advantages	Limitations
2T0C GC	Read	BL	N	0.0195μm^2	(+) Access Speed (+) Lower R Loading	(-) Small C_{SN} (-) Sneak Path (-) IR Drop (-) Split R/W Periph.
		WL	Y			
	Write	BL	Y			
		WL	N			
3T0C GC	Read	BL	Y	0.0251μm^2	(+) No Sneak Path (+) No IR Drop	(-) High Leakage (-) Larger Cell (-) Higher R Loading (-) Split R/W Periph.
		WL	N			
	Write	BL	Y			
		WL	N			
1T1C eDRAM	BL		Y	DG: 0.027μm^2 VGAA: 0.0182μm^2	(+) Dedicated C_{SN} (+) Fewer Periph.	(-) Destructive Read (-) Slower Access
	WL		N			

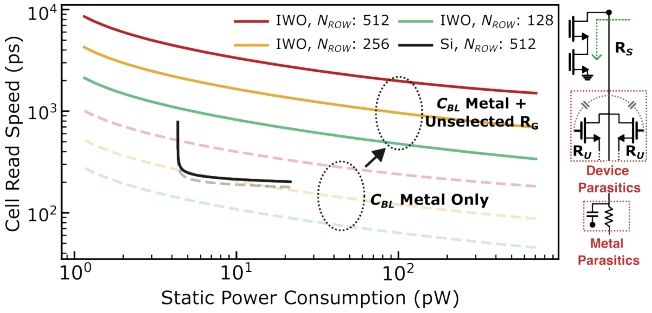


Figure 13: Read speed vs. leakage in 2T read port of 7 nm Si and AOS

same tradeoff, which we study using SPICE in the case of a single-fin CMOS 2T read port and an AOS 2T read port with a 150 nm W_{RA} and W_{RG} (Fig. 13). We observe that the AOS 3T0C cell pays a higher price for high speed than its Si counterpart primarily due to (1) the reduced current density of the AOS channel transistor and (2) the higher parasitic capacitance created by large overlap regions, which increases the total bitline charge and thus the sink current density requirement of the read port for the same read speed. Therefore, to achieve sub-nanosecond RM development, an AOS 3T0C array with an N_{row} exceeding 128 imposes cell standby power orders of magnitude higher than HD-SRAM (~14 pW). This, alongside the increased peripheral leakage (drivers) and cell-size disadvantages when compared to 2T0C and 1T1C, makes the DG IWO 3T0C an unsuitable choice for high-bandwidth, energy-efficient last-level cache memory (§5.2.3). Alternative AOS device geometries, such as the self-aligned gated structure with plasma-treated, conductive source/drain [35], may prove better suited for 3T0C integration; reductions in the BL capacitance contribution from device parasitics shift the read/leakage tradeoff downwards, showing a larger leakage suppression under the same read speed over Si (Fig. 13).

5.1.2 Access Speed and Readout in AOS 1T1C. Given the BEOL process compatibility of stacked capacitor fabrication, it is worthwhile to consider the advantages of a fully BEOL-compatible 1T1C eDRAM utilizing an AOS access transistor (Fig. 12a), reminiscent of that recently demonstrated by TSMC [8]. For one, split read/write paths in gain-cell topologies require additional peripherals to decode and drive both R/W paths, the footprint of which is truncated in an AOS 1T1C array using a single BL/WL per cell, thus delivering greater density and static power benefits. Additionally, using a dedicated capacitor, as opposed to solely the parasitic RA gate capacitance, allows us to raise C_{SN} and thus enable more flexibility in retention, as higher cumulative charge is stored. However, these transformations also have limitations; for one, the C_{SN} cannot be arbitrarily set, as the RM (ΔV_{BL}) is a function of the BL capacitance (C_{BL}), SN capacitance (C_{SN}), and minimum SN voltage after retention losses (V_{min}):

$$\Delta V_{BL} = \frac{1}{1 + C_{BL}/C_{SN}} \left(\frac{1}{2} V_{DD} - \frac{I_{leak} \cdot t_{ret}}{C_{SN}} \right) \quad (4)$$

The higher bitline capacitance contribution of AOS device integration increases the minimum C_{SN} for 100 mV RM (Fig. 14a). We find that at N_{row} = 128, the minimum C_{SN} must be ~10 fF when the access device width is 200 nm (V_{min} = 600 mV), and that the minimum C_{SN} is a strong function of the access device width. The secondary bottleneck to consider is the decreased current density of the AOS access device, which increases access time. Since the

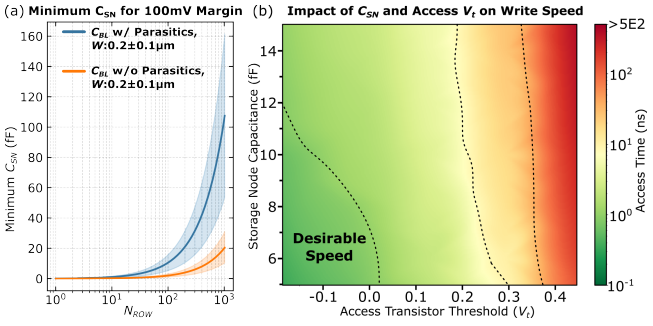


Figure 14: (a) Impact of cell parasitics on minimum C_{SN} requirement, (b) relationship of C_{SN} and access V_t on write speed in AOS 1T1C

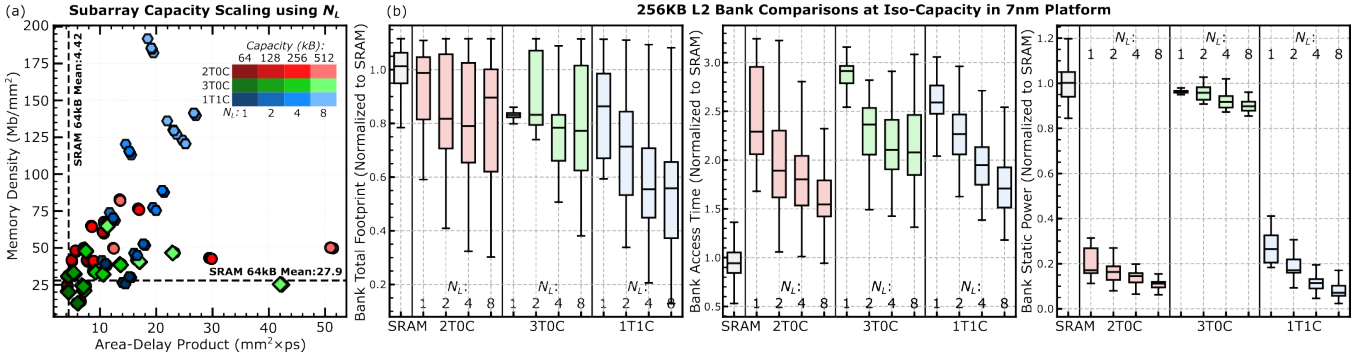


Figure 15: (a) 3D stacking speed-density tradeoff demonstrated in a 64kB (baseline) subarray with 1 ns RCT (excluding write-back). (b) 256kB memory bank footprint, access time, and static power distribution comparison of AOS cache candidates and SRAM.

readout of the 1T1C cell is destructive, a write-back must be issued during every read transaction; thus, access speed is a critical bottleneck in 1T1C arrays. If the RM for a desired N_{row} sets the lower bound for C_{SN}, then the requisite RCT/access speed sets the upper bound. We study the relationship between access speed, C_{SN}, and V_t in a 100 nm width IWO transistor using a V_{cc} of 750 mV (Fig. 14b). To achieve sub-nanosecond cell access speed, it is desirable to use a smaller C_{SN} (< 8-12 fF) with a low or negative V_t (≤ 0 V), although the former comes at the expense of limiting N_{row}. Based on the previous results, the following sections set C_{SN} to 10 fF, an access device width of 300 nm (limiting N_{row} to 64), and a V_{hold} of -300 mV (requiring negative level shifting). Shown in Table I and Fig. 12b, a wide access device in the quasi-planar double-gated geometry imposes a restrictive cell footprint. Therefore, a vertical gate-all-around (VGAA) structure with a cylindrical channel is adopted to improve cell density [8].

5.1.3 Subarray and Bank-Level Comparison. To draw comparisons between macro-level implementations of AOS-based cache memories, we conduct two distribution-based studies: (1) given a single subarray with a variable number of mats and 1 ns RCT constraint, what is the effect on density and speed if multiple stacked memory layers are used to *increase capacity*? (2) Given a 256 kB bank with a variable number of sub-banks and nanosecond sub-bank RCT, what is the effect on area, speed, and static power if stacked memory layers are used to *increase density* (Fig. 13)? Given the lower access speed of 1T1C, the RCT restriction is placed on the read operation without the write back processes.

We discern from Fig. 15a, which illustrates the subarray study (1), that AOS cache memories exhibit a higher area-delay product than their SRAM counterparts, as expected due to their slower operation speeds resulting from the lower electron mobility in AOS devices and the differential sensing achieved using feedback in SRAM. The density benefits of gain-cell topologies taper off quickly as multiple tiers are integrated, trading off quickly with delay as the N_L is increased, resulting in a reduced slope. One reason for this is the overhead of 3D decoding in gain cell topologies compared to 1T1C: aside from the additional peripherals (drivers) for split R/W paths, the WWL drivers in the gain-cell topology are larger to handle the high voltage swing requirement (V_{hold} to V_{boost}), and thus the transmission gate used to decode the selected layer is proportionally sized to last stage of the driver, leading to a higher footprint cost per layer than the 1T1C counterpart with reduced WL swing. As a result, at eight layers we observe that a 512kB 2T0C subarray under a 1 ns RCT restriction can achieve memory densities of 82.87 Mb/mm^2 (~2.72× SRAM peak), 3T0C can achieve 65.68 Mb/mm^2 (~2.16× SRAM peak), and 1T1C can achieve 191.8 Mb/mm^2 (~6.1× SRAM peak).

To impose a limitation on the total footprint in our bank-level study, we estimate the area of a 256 kB read-optimized bank, which is reflective of the L2 partition capacity used in an NVIDIA Ampere GPU [3],[49], yielding a footprint of ~80,000 μm^2. We plot the distributions of bank footprint, access time and static power for AOS cache candidates in ascending N_L in Fig. 15b. We first observe that switching the loading of the read decode path from the source of RA to the gate of RG increases partitioning of arrays, and thus larger footprints and bank latencies in 3T0C compared to 2T0C and 1T1C, that cannot be explained by cell size increases alone. Although not at the level of SRAM, the AOS 2T0C bank speed median and minimum values strongly outclass those of AOS 3T0C and 1T1C. Gain-cell topologies exhibit larger footprints than eDRAM and stronger leveling off of footprint reduction as a function of N_L. Similar to the prior subarray result, 1T1C achieves the highest footprint reduction as a function of the number of tiers among the three candidates. The reduced cell-level leakage significantly improves the overall consumption of the array in 2T0C and 1T1C, although the higher driver leakage (as a function of sizing) maintains averages in the ~10-40% range of SRAM, with substantial reductions realized using stacked arrays. Conversely, to maintain reasonable read speeds in 3T0C, the cell-level leakage is orders of magnitude higher than its 2T0C and 1T1C counterparts. At V_t = 250 mV, the per-cell 3T0C static power is ~2.67 pW, resulting in bank-level static power consumptions comparable to that of SRAM. For the stated density, static power, and speed limitations, we exclude 3T0C from the following system-level benchmarking study, which investigates the implications of M3D AOS memory integration for enhancing LLC capacity and improving GPU performance.

6 Benchmarking Methodology

We evaluate the proposed integration of IWO 2T0C and 1T1C L2 caches using the cycle-accurate GPGPU simulator Accel-Sim [28]. We model the baseline system after a verified NVIDIA Ampere RTX

Table 2: Baseline GPU Benchmarking Configuration

Parameter	Configuration
Number of SMs	46
Schedulers per Core	4, Loose Round-Robin[54]
GPU Memory Interface	256-bit GDDR6
GPU Memory Capacity	8 GB
L1/Shared Memory Capacity	128 KB per SM
Register File Capacity	256 KB per SM

Table 3: Evaluated Benchmarks and Corresponding Domain

Application	Abbrev.	Domain
Covariance Computation [7]	cov	Pattern Recognition
Particle Filter [7]	pfil	Medical Imaging
Discrete 2D Wavelet Transform [7]	dwt2d	Image/Video Compression
Convolutional Neural Network Training [2]	cnn	Deep Learning
Matrix Transpose and Vector Multiplication [23]	atax	Linear Algebra
Back Propagation [7]	backprop	Pattern Recognition
Matrix Vector Product and Transpose [23]	mvt	Linear Algebra
Pathfinder [7]	pfin	Dynamic Programming
3D Convolution [7]	3dconv	Image Processing
GEMM Kernel Inference [2]	gemm	Deep Learning
Needleman-Wunsch [7]	nw	Bioinformatics
B+ Tree [7]	b+tree	Search
RNN + GRU Training [2]	rnn	Deep Learning
Correlation Computation [7]	corr	Signal Processing
3 Matrix Multiplication [23]	3mm	Linear Algebra

3070 GPU model [3], with system simulation parameters listed in Table II. Though the capacity of the L2 is small in Ampere GPUs (4 MB) compared to the state of the art (Blackwell, 126 MB), studying the implications on a validated, compact model can provide broader insights into the performance implications on larger GPGPU LLCs. To provide a comprehensive evaluation of performance, we randomly select 15 compute- and memory-bound applications from the Rodinia [7], Polybench [23], and DeepBench [2] benchmarking suites, targeting relevant workloads in scientific simulation, image/signal processing, graph analytics, linear algebra, and deep learning primitives (Table III). To achieve cycle-accurate replay in Accel-Sim, we utilize Accel-Sim's trace-driven mode, which employs dynamic SASS instructions logged using NVIDIA's NVBit instrumentation framework [69]. We modify the integrated memory partition model to account for the impact of (distributed) refresh operations on performance, which are incorporated into reservation failure statistics and discussed further at the end of §7.

To understand the impact of both high bandwidth and high capacity in scaled L2 AOS-based LLCs, we model four M3D-based L2 systems, two for each AOS memory type: (1) *Iso-Banking (IB)*: assuming the same footprint and number of data banks, what is the performance of an L2 that leverages stacking to maximize per-bank capacity? (2) *Iso-Bank Capacity (IBC)*: assuming the same bank capacity and aggregate footprint, what is the performance of an L2 that leverages stacking to decrease bank size and maximize the number of banks? The parameters of each evaluated system are shown in Table IV. We set a relaxed L2 footprint constraint per memory partition of 200,000 μm^2. In *IB* studies, we halt increasing N_L once no configurations exist within the footprint constraint. The L2 clock domain is used to model the operating frequency, and the raster operations pipeline (ROP) latency is used to model the

Table 4: Benchmarking L2 Cache Parameters

Config	Memory Type	Num Banks	L2 Capacity	N_L	L2 Clock Domain	L2 Area p. Partition	ROP Latency	Ref Period
Baseline	SRAM	8×2	4 MB	1	1132 MHz	160,328 μm^2	187 cyc.	N/A
2T0C IB	2T0C IWO	8×2	8 MB	2	1067 MHz	130,453 μm^2	184 cyc.	859 μs
2T0C IBC	2T0C IWO	8×8	16 MB	8	1132 MHz	195,044 μm^2	188 cyc.	215 μs
1T1C IB	1T1C IWO	8×2	32 MB	8	724 MHz	175,771 μm^2	190 cyc.	244 μs
1T1C IBC	1T1C IWO	8×16	32 MB	8	724 MHz	166,785 μm^2	190 cyc.	244 μs

L2 latency [53]. Because memory partitions (and thus L2 banks) are highly distributed using complex networks on chip (NoCs), we only consider alterations to the ROP latency based on local changes in latency within each partition relative to the SRAM baseline. Additionally, the overhead of tag memories is omitted from this study; however, we note that tag/directory overhead will increase proportionally to capacity [73] and with additional partitioning [60].

7 Evaluation

In Fig. 16, we track total performance (instructions per cycle, IPC), application runtime, and performance per watt for each benchmark and its geometric mean. Based on pairings where performance is optimized according to *IBC* configurations or high-capacity 1T1C configurations, benchmarks can be broadly categorized as either bandwidth-limited (e.g., gemm) or capacity-limited (e.g., correlation), respectively. Both 1T1C-*IB* and 1T1C-*IBC* deliver elevated mean performance over the SRAM baseline, indicating that the elevated capacity made possible using AOS 1T1C integration surpasses the limitation on maximum per-bank bandwidth. However, there is a clear distinction between 1T1C-*IB* mean IPC and runtime, and the IPC and runtime in the lower-capacity 2T0C-*IBC*. This reinforces the necessity for bandwidth in both bandwidth-limited applications and the maintenance of performance in compute-bound applications. 1T1C-*IBC* demonstrates the highest performance, yielding a ∼8% increase in IPC over SRAM and ∼2.7× mean increase in Perf/W. In select cases, IPC is up to ∼38% higher than the baseline, and Perf/W can be more than 5× higher. Although a few cases suffer slightly from the lower frequency in 1T1C-*IBC* cases (e.g., *particle-filter*), these performance reductions are relatively negligible (< 1%). Although 1T1C-*IBC* demonstrates the highest mean Perf/W, the highest peak Perf/W is observed in 2T0C-*IB* cases such as *gemm* and *rnn*. This should not be taken at face value: because power is the energy consumed over time, and the runtime of applications is longer with lower leakage (smaller capacity) in 2T0C-*IB* cases than in 1T1C-*IBC*, the higher performance per watt does not always indicate greater efficiency. Based on the performance gains observed in both *IBC* cases, we determine that the best use case (in terms of performance) utilizing AOS memories is leveraging stacking to reduce bank size and increase bank count.

We present miss rates and energy consumption breakdowns per benchmark at the bank level in Fig. 17. This analysis yields two key findings. First, strongly reduced miss rates do not necessarily equate to higher performance, even when access frequency is high. Take, for example, *cnn*, which has the highest read access frequency among all benchmarks and a ∼40% reduction in miss rate in 1T1C cases but only demonstrates a ∼2% increase in IPC

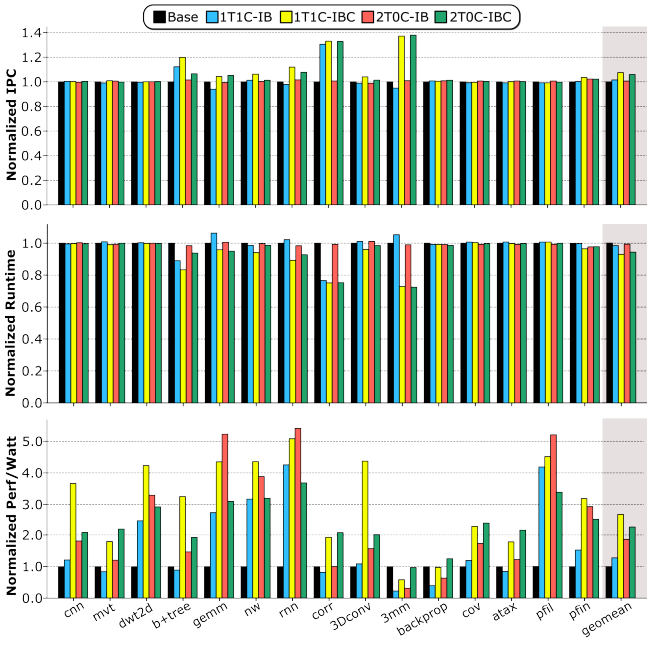


Figure 16: Instructions per cycle (IPC), runtime and performance per watt comparisons of evaluated systems, normalized to baseline

due to its ultimately compute-bound nature [72]. However, a goal of the GPGPU L2 is to alleviate the bandwidth requirements on the off-chip DRAM and reduce costly off-chip accesses, to which the corresponding reductions in miss rate aid in overall energy reduction. Second, at the bank level, SRAM leakage is often the dominant consumer of energy in the L2 cache; however, in cases with many read and write accesses (e.g., *3mm*), dynamic energy consumption can quickly outpace static power. As was seen in §2, array-level dynamic power consumption is increased in AOS caches due to both increased capacitances and voltage swings, and restrictions on array size (which result in higher activation per transaction). This means that bank-level energy is dramatically increased in cases

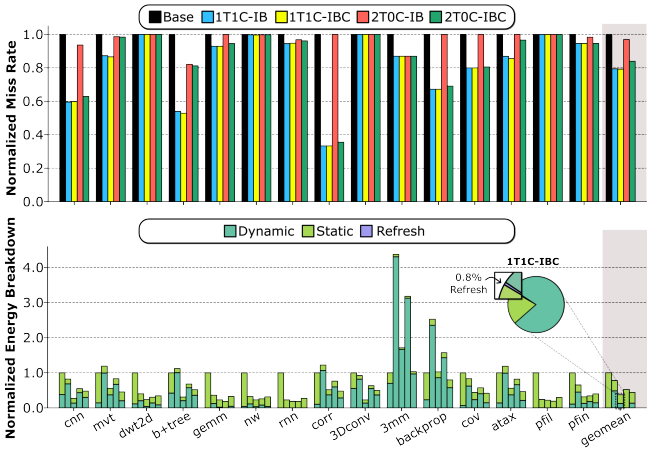


Figure 17: Miss rate and corresponding system energy breakdown normalized to baseline. Refresh contributes negligible overhead

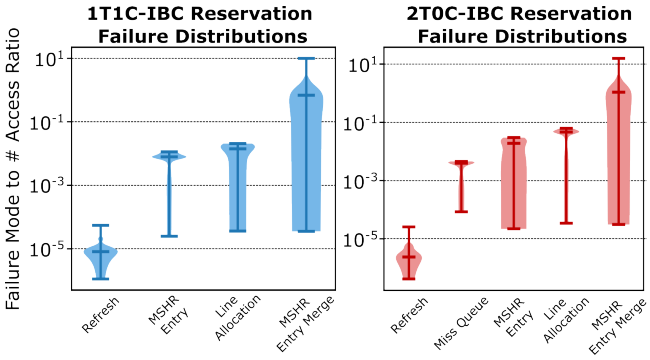


Figure 18: Distribution of reservation failure modes in 1T1C- and 2T0C-*IBC* configurations gathered across benchmarked kernels

such as *backprop* and *3mm*, which reduces the performance per watt. Reinforcing the prior statements on stacking methodology made in the preceding paragraph, we observe that *IBC* configurations consume less dynamic energy than their *IB* counterparts, as the size of each tiered array can be reduced, thereby imposing lower access energy. We also observe that infrequent refresh operations incur minimal energy costs across all AOS LLC configurations, with a breakdown shown for the case with the highest geometric mean refresh energy consumption: 1T1C-*IBC*, where line refreshes consume ~0.8% of all energy (Fig. 17).

To understand the impact of extended retention (or infrequent refresh), we plot the distribution of modes of L2 reservation failures as a ratio of total L2 accesses per kernel across all benchmarks for *IBC* cases (Fig. 18). It is worth defining the most common non-refresh-based modes of reservation failures for context. A *line allocation failure* refers to when a new tag entry for the access is needed, but all ways in the target set are reserved by outstanding misses or pending fills. An *MSHR entry failure* occurs when a fresh miss status holding register (MSHR) entry needs to be allocated, but the table is full. An *MSHR merge failure* occurs when a miss for an address is already being tracked, but the per-MSHR merge list is full, and another requester cannot be recorded. Finally, a *miss queue failure* occurs when the miss queue can no longer allocate more incoming requests to off-chip DRAM. In our benchmarking, distributed refreshes occur for one cycle (1T1C) to two cycles (2T0C) in the L2 clock domain at a period of $t_{\text{ret}}/N_{\text{row}}/N_L$, as each layer in the 3D decoding scheme operates as an independent but peripherally coalesced matrix. Notably, the peak frequency of refresh-induced read and write reservation failures at this periodicity is comparable to that of the lower bound of MSHR entry failures and is typically overshadowed by MSHR entry and line allocation failure modes by several orders of magnitude, indicating that retention levels in the tens to hundreds of ms range are sufficient for large GPGPU cache performance without incurring notable performance degradation due to L2 stalling. We also observe that the move from 4× to 8× capacity eliminates all miss queue failures in the 1T1C case, but heightens the distribution of line allocation failures, as contention over oversubscribed indices in each set (due to the higher hit rate) grows. Consequently, this notion poses the need to increase associativity alongside the set size [29].

8 Related Work

Several emerging memories have been the subject of study for GPU register file (RF) integration due to their density and static power advantages. Magnetic tunnel junction (MTJ) based multi-port racetrack memory [42], STT-MRAM [39],[71],[22], and SOT-MRAM [44] register files have been the focus of prior GPU system-level studies. Much of this body of work is centered on minimizing the shift-operation and write-speed bottlenecks of emerging non-volatile memory register files, which carry latencies on the order of ~tens of nanoseconds. Volatile Si eDRAM variants [27] and SRAM-DRAM hybrid memory [77] have also been studied at the system level as a compact RF solution. Low-leakage multi-ported SRAM designs [26] and FDSOI eDRAM [20] have been studied with GPU-register-centric integration in mind at the circuit and device levels.

STT-MRAM [63],[24], Spintronic tape memories [66], and eDRAM [67] have also been the focus of GPU LLC expansion in prior work, typically benchmarked on Fermi architecture lines. Like their register file study counterparts, these works investigate microarchitectural methods for hiding long write latencies, minimizing the overhead of shift operations, and reducing the refresh overheads inherent to each memory type. Morpheus [13] discusses utilizing idle SM resources (L1D, Shared Memory, RF) as configurable extensions to the LLC.

Monolithic 3D integration has been studied in GPU memory and network subsystems, such as multi-tier register file banks and mesh networks [31] and the private L1 data cache [14]. Additionally, an M3D integrated NoC for GPU cache bypassing has been explored [15]. We note that these works are not technology aware and presume that multi-tiered CMOS integration is achieved with comparable performance to FEOL devices; therefore, they are opportunistic.

Finally, dense AOS-based memory systems have been the subject of a few benchmarking studies in CPU, TPU, and non-Von Neumann computing systems. [38],[32] benchmark DCIM integration of hybrid 2T0C-RRAM and 2R1W AOS gain cells. [37] performs a benchmark of a Sn-doped In_2O_3 (ITO) 2T0C L1D extension in a TPU benchmark; however, the device-centric nature of the work leaves little room for discussion on macro- or system-level design assumptions and fundamental bottlenecks. [40] Focuses on the integration of IWO 2T0C as a TPU buffer memory, comparing it to other mainstream and emerging memories. The CPU cache-based IWO 2T0C study, as discussed in [73], examines system implications using GEM5. Although the study examines limitations on bank-level write, leakage, and retention implications in write design, there is limited discussion on read implications, and it lacks a study of multiple system configurations.

9 Conclusion

This paper presents a study on the integration opportunities of monolithically 3D stacked amorphous oxide semiconductor (AOS) memories in capacitive memory topologies to tackle GPU memory-system bottlenecks. By observation of the relatively short lifetime of register operands, we develop integration methods for a high-speed multi-ported AOS gain cell capable of delivering three times the read ports, roughly three-quarters of the bank size of comparable 1R1W SRAM. Furthermore, we investigate the integration of stacked BEOL-compatible 3T0C, 2T0C, and 1T1C memories, demonstrating that 2T0C can achieve densities 2.72× higher than SRAM without sacrificing the maximum operating frequency, and 1T1C can deliver up to 6.1× the density of SRAM in 8-tier configurations. Benchmarking on a baseline NVIDIA Ampere GPU in a modified version of Accel-Sim indicates that AOS-based 1T1C last-level caches can boost IPC by a geometric mean of 8%, and as high as 38%, and performance per watt up to 5.1× over the baseline HD-SRAM system. Such CMOS+X stacking, therefore, offers a manufacturable path to reclaim area, bandwidth, and energy headroom that conventional SRAM scaling can no longer provide, enabling larger warp sizes, higher SM counts, and/or denser LLCs without increasing die size.

Acknowledgments

The authors thank Yu-Ming Lin and Huai-Ying Huang of TSMC, Hsinchu, Taiwan, for guiding technical discussions.

References

[1] Tor M Aamodt, Wilson Wai Lun Fung, Timothy G Rogers, and Margaret Martonosi. 2018. *General-purpose graphics processor architectures.* Springer.
[2] Baidu Research. 2018. DeepBench: Benchmarking Deep Learning Operations on Different Hardware. [Online]. Available: https://github.com/baidu-research/DeepBench.
[3] Ali Bakhoda, George L Yuan, Wilson WL Fung, Henry Wong, and Tor M Aamodt. 2009. Analyzing CUDA workloads using a detailed GPU simulator. In *2009 IEEE international symposium on performance analysis of systems and software.* IEEE, 163–174.
[4] Attilio Belmonte and Gouri Sankar Kar. 2025. Disrupting the DRAM roadmap with capacitor-less IGZO-DRAM technology. *Nature Reviews Electrical Engineering* (2025), 1–2.
[5] Jiashen Cao, Rathijit Sen, Matteo Interlandi, Joy Arulraj, and Hyesoon Kim. 2023. Gpu database systems characterization and optimization. *Proceedings of the VLDB Endowment* 17, 3 (2023), 441–454.
[6] Leland Chang, Robert K Montoye, Yutaka Nakamura, Kevin A Batson, Richard J Eickemeyer, Robert H Dennard, Wilfried Haensch, and Damir Jamsek. 2008. An 8T-SRAM for variability tolerance and low-voltage operation in high-performance caches. *IEEE Journal of Solid-State Circuits* 43, 4 (2008), 956–963.
[7] Shuai Che, Michael Boyer, Jiayuan Meng, David Tarjan, Jeremy W Sheaffer, Sang-Ha Lee, and Kevin Skadron. 2009. Rodinia: A benchmark suite for heterogeneous computing. In *2009 IEEE international symposium on workload characterization (IISWC).* Ieee, 44–54.
[8] Katherine H Chiang, Yen-Chung Ho, Ming-Yen Chuang, Chih-Yu Chang, Yun-Feng Kao, Hsin-Yi Yang, Chien-Hua Huang, Yu-Jen Chien, Yin-Hao Wu, Yi-Ching Liu, et al. 2025. Integration of 0.75 VV DD Oxide-Semiconductor 1T1C Memory with Advanced Logic for an Ultra-Low-Power Low-Latency Cache Solution. In *2025 Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits).* IEEE, 1–3.
[9] Gwanghyeon Choe, Jisu Kwak, and Shimeng Yu. 2023. Machine Learning-Assisted Compact Modeling of W-Doped Indium Oxide Channel Transistor for Back-End-of-Line Applications. *IEEE Trans. Electron Devices* 71, 1 (2023), 231–238.
[10] Gihun Choe, Jungyoun Kwak, and Shimeng Yu. 2023. Machine learning-assisted compact modeling of W-doped indium oxide channel transistor for back-end-of-line applications. *IEEE Transactions on Electron Devices* 71, 1 (2023), 231–238.
[11] Hong Jun Choi, Dong Oh Son, Cheol Hong Kim, and Jong Myron Kim Kim. 2014. Impact of clock frequency and number of cores on gpu performance. In *2014 International Conference on IT Convergence and Security (ICITCS).* IEEE, 1–4.
[12] Lawrence T Clark, Vinay Vashishtha, Lucian Shifren, Aditya Gujja, Saurabh Sinha, Brian Cline, Chandarasekaran Ramamurthy, and Greg Yeric. 2016. ASAP7: A 7-nm finFET predictive process design kit. *Microelectronics Journal* 53 (2016), 105–115.
[13] Sina Darabi, Mohammad Sadrosadati, Negar Akbarzadeh, Joël Lindegger, Mohammad Hosseini, Jisung Park, Juan Gómez-Luna, Onur Mutlu, and Hamid Sarbazi-Azad. 2022. Morpheus: Extending the last level cache capacity in GPU systems using idle GPU core resources. In *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO).* IEEE, 228–244.
[14] Cong Thuan Do, Young-Ho Gong, Cheol Hong Kim, Seon Wook Kim, and Sung Woo Chung. 2019. Exploring the relation between monolithic 3D L1 GPU cache capacity and warp scheduling efficiency. In *2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED).* IEEE, 1–6.

[15] Cong Thuan Do, Cheol Hong Kim, and Sung Woo Chung. 2023. Aggressive GPU cache bypassing with monolithic 3D-based NoC. *The Journal of Supercomputing* 79, 5 (2023), 5421–5442.

[16] Xinlv Duan, Kailiang Huang, Junxiao Feng, Jiebin Niu, Haibo Qin, Shihui Yin, Guangfan Jiao, Daniele Leonelli, Xiaoxuan Zhao, Zhaogui Wang, et al. 2022. Novel vertical channel-all-around (CAA) In-Ga-Zn-O FET for 2T0C-DRAM with high density beyond 4F 2 by monolithic stacking. *IEEE Transactions on Electron Devices* 69, 4 (2022), 2196–2202.

[17] Kayvon Fatahalian, Jeremy Sugerman, and Pat Hanrahan. 2004. Understanding the efficiency of GPU algorithms for matrix-matrix multiplication. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*. 133–137.

[18] Eric S Fetzer, David Dahle, Casey Little, and Kevin Safford. 2006. The parity protected, multithreaded register files on the 90-nm itanium microprocessor. *IEEE Journal of Solid-State Circuits* 41, 1 (2006), 246–255.

[19] Mark Gebhart, Stephen W Keckler, Brucek Khailany, Ronny Krashinsky, and William J Dally. 2012. Unifying primary cache, scratch, and register file memories in a throughput processor. In *2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE, 96–106.

[20] Roman Golman, Robert Giterman, and Adam Teman. 2024. Multi-ported GC-eDRAM bitcell with dynamic port configuration and refresh mechanism. *Journal of Low Power Electronics and Applications* 14, 1 (2024), 2.

[21] Nilanjan Goswami, Bingyi Cao, and Tao Li. 2013. Power-performance co-optimization of throughput core architecture using resistive memory. In *2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 342–353.

[22] Nilanjan Goswami, Bingyi Cao, and Tao Li. 2013. Power-performance co-optimization of throughput core architecture using resistive memory. In *2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 342–353.

[23] Scott Grauer-Gray, Lifan Xu, Robert Searles, Sudhee Ayalasomayajula, and John Cavazos. 2012. Auto-tuning a high-level language targeted to GPU codes. In *2012 innovative parallel computing (InPar)*. Ieee, 1–10.

[24] Shaopu Han and Yanfeng Jiang. 2024. Advanced hybrid MRAM based novel GPU cache system for graphic processing with high efficiency. *AIP Advances* 14, 1 (2024).

[25] Mark Horowitz, Paul Chow, Don Stark, Richard T Simoni, Arturo Salz, Steven Przybylski, John Hennessy, Glenn Gulak, Anant Agarwal, and John M Acken. 1987. MIPS-X: A 20-MIPS peak, 32-bit microprocessor with on-chip cache. *IEEE Journal of Solid-State Circuits* 22, 5 (1987), 790–799.

[26] Shen-Fu Hsiao and Pu-Cheng Wu. 2014. Design of low-leakage multi-port SRAM for register file in graphics processing unit. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2181–2184.

[27] Naifeng Jing, Yao Shen, Yao Lu, Shrikanth Ganapathy, Zhigang Mao, Minyi Guo, Ramon Canal, and Xiaoyao Liang. 2013. An energy-efficient and scalable eDRAM-based register file architecture for GPGPU. *ACM SIGARCH Computer Architecture News* 41, 3 (2013), 344–355.

[28] Mahmoud Khairy, Zhesheng Shen, Tor M Aamodt, and Timothy G Rogers. 2020. Accel-sim: An extensible simulation framework for validated gpu modeling. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 473–486.

[29] Mahmoud Khairy, Mohamed Zahran, and Amr G Wassal. 2015. Efficient utilization of gpgpu cache hierarchy. In *Proceedings of the 8th Workshop on General Purpose Processing using GPUS*. 36–47.

[30] Changkyu Kim, Doug Burger, and Stephen W Keckler. 2002. An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches. In *Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*. 211–222.

[31] Chang Hyun Kim, Hun Jae Lee, Sung Woo Chung, and Seon Wook Kim. 2019. Design of a GPU Register File Using a Monolithic 3D Technique. *IEIE Transactions on Smart Processing & Computing* (2019), 499–505.

[32] Jungyoun Kwak, Gihun Choe, Junmo Lee, and Shimeng Yu. 2024. Monolithic 3D transposable 3T embedded DRAM with back-end-of-line oxide channel transistor. In *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 1–5.

[33] Charles Eric LaForest and J Gregory Steffan. 2010. Efficient multi-ported memories for FPGAs. In *Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays*. 41–50.

[34] Hung Q Le, William J Starke, J Stephen Fields, Francis P O'Connell, Dung Q Nguyen, Bruce J Ronchetti, Wolfram M Sauer, Eric M Schwarz, and Michael T Vaden. 2007. Ibm power6 microarchitecture. *IBM Journal of Research and Development* 51, 6 (2007), 639–662.

[35] Jin Kyu Lee, Soobin An, and Soo-Yeon Lee. 2023. Self-aligned top-gate IGZO TFT with stepped structure for suppressing short channel effect. *IEEE Electron Device Letters* 44, 11 (2023), 1845–1848.

[36] Peijing Li, Matthew Hung, Yiming Tan, Konstantin Hoßfeld, Jake Cheng Jiajun, Shuhan Liu, Lixian Yan, Xinxin Wang, H-S Philip Wong, and Thierry Tambe. 2025. GainSight: Application-Guided Profiling for Composing Heterogeneous On-Chip Memories in AI Hardware Accelerators. *arXiv preprint arXiv:2504.14866* (2025).

[37] Shuhan Liu, Koustav Jana, Kasidit Toprasertpong, Jian Chen, Zheng Liang, Qi Jiang, Sumaiya Wahid, Shengjun Qin, Wei-Chen Chen, Eric Pop, et al. 2024. Design guidelines for oxide semiconductor gain cell memory on a logic platform. *IEEE Transactions on Electron Devices* 71, 5 (2024), 3329–3335.

[38] Shuhan Liu, Robert M Radway, Xinxin Wang, Filippo Moro, Jean-Francois Nodin, Koustav Jana, Lixian Yan, Shuting Du, Luke R Upton, Wei-Chen Chen, et al. 2025. Monolithic 3-D Integration of Diverse Memories: Resistive Switching (RRAM) and Gain Cell (GC) Memory Integrated on Si CMOS. *IEEE Transactions on Electron Devices* (2025).

[39] Xiaoxiao Liu, Mengjie Mao, Xiuyuan Bi, Hai Li, and Yiran Chen. 2015. An efficient STT-RAM-based register file in GPU architectures. In *The 20th Asia and South Pacific Design Automation Conference*. IEEE, 490–495.

[40] Anni Lu, Junmo Lee, Tae-Hyeon Kim, Muhammed Ahosan Ul Karim, Rebecca Sejung Park, Harsono Simka, and Shimeng Yu. 2024. High-speed emerging memories for AI hardware accelerators. *Nature Reviews Electrical Engineering* 1, 1 (2024), 24–34.

[41] Yandong Luo, Sourav Dutta, Ankit Kaul, Sung Kyu Lim, Muhannad Bakir, Suman Datta, and Shimeng Yu. 2022. A compute-in-memory hardware accelerator design with back-end-of-line (BEOL) transistor based reconfigurable interconnect. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 12, 2 (2022), 445–457.

[42] Mengjie Mao, Wujie Wen, Yaojun Zhang, Yiran Chen, and Hai Li. 2014. Exploration of GPGPU register file architecture using domain-wall-shift-write based racetrack memory. In *Proceedings of the 51st Annual Design Automation Conference*. 1–6.

[43] Sparsh Mittal. 2016. A survey of techniques for architecting and managing GPU register file. *IEEE Transactions on Parallel and Distributed Systems* 28, 1 (2016), 16–28.

[44] Sparsh Mittal, Rajendra Bishnoi, Fabian Oboril, Haonan Wang, Mehdi Tahoori, Adwait Jog, and Jeffrey S Vetter. 2017. Architecting SOT-RAM based GPU register file. In *2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 38–44.

[45] Veynu Narasiman, Michael Shebanow, Chang Joo Lee, Rustam Miftakhutdinov, Onur Mutlu, and Yale N Patt. 2011. Improving GPU performance via large warps and two-level warp scheduling. In *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*. 308–317.

[46] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2025. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology* 16, 5 (2025), 1–72.

[47] Stefan Nikolić, Francky Catthoor, Zsolt Tőkei, and Paolo Ienne. 2021. Global is the new local: FPGA architecture at 5nm and beyond. In *The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. 34–44.

[48] Hiroki Noguchi, Shunsuke Okumura, Yusuke Iguchi, Hidehiro Fujiwara, Yasuhiro Morita, Koji Nii, Hiroshi Kawaguchi, and Masahiko Yoshimoto. 2008. Which is the best dual-port SRAM in 45-nm process technology?—8T, 10T single end, and 10T differential—. In *2008 IEEE International Conference on Integrated Circuit Design and Technology and Tutorial*. IEEE, 55–58.

[49] NVIDIA Corporation. 2020. NVIDIA Ampere GA102 GPU Architecture, White Paper, ver. 2.0. [Online]. Available: https://www.nvidia.com/content/PDF/nvidia-ampere-ga-102-gpu-architecture-whitepaper-v2.pdf.

[50] NVIDIA Corporation. 2024. NVIDIA Announces Financial Results for First Quarter Fiscal 2025. Press release. [Online]. Available: https://nvidianews.nvidia.com/news/nvidia-announces-financial-results-for-first-quarter-fiscal-2025.

[51] NVIDIA Corporation. 2025. NVIDIA Technologies and GPU Architectures. [Online]. Available: https://www.nvidia.com/en-us/technologies/.

[52] Hyungrock Oh, Attilio Belmonte, Manu Perumkunnil, Jerome Mitard, Nouredine Rassoul, Gabriele Luca Donadio, Romain Delhougne, Arnaud Furnemont, Gouri Sankar Kar, and Wim Dehaene. 2021. Enhanced data integrity of In-Ga-Zn-oxide based capacitor-less 2T memory for DRAM applications. In *ESSDERC 2021-IEEE 51st European Solid-State Device Research Conference (ESSDERC)*. IEEE, 275–278.

[53] Molly A O'Neil and Martin Burtscher. 2014. Microarchitectural performance characterization of irregular GPU kernels. In *2014 IEEE International Symposium on Workload Characterization (IISWC)*. IEEE, 130–139.

[54] S. Girish Pandey and Shobha Gopalakrishnan. 2019. Improving GPGPU Performance Using Efficient Scheduling. In *Proc. 2019 Int. Conf. Intelligent Sustainable Systems (ICISS)*. Palladam, Tamil Nadu, India, 570–577.

[55] Tanvir Haider Pantha, Sharadindugopal Kirtania, Khandker Akif Aabrar, Sunbin Deng, Suman Datta, and Sourav Dutta. 2024. Design space exploration of oxide semiconductorbased monolithic 3D gain cell memory. In *2024 IEEE European Solid-State Electronics Research Conference (ESSERC)*. IEEE, 125–128.

[56] Jae Chul Park, Sang Wook Kim, Sun Il Kim, Huaxiang Yin, Ji Hyun Hur, Sang Hun Jeon, Sung Ho Park, I Hun Song, Young Soo Park, U In Chung, et al. 2009. High performance amorphous oxide thin film transistors with self-aligned top-gate structure. In *2009 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 1–4.

[57] Jin-Seong Park, Jae Kyeong Jeong, Yeon-Gon Mo, Hye Dong Kim, and Chang-Jung Kim. 2008. Control of threshold voltage in ZnO-based oxide thin film transistors. *Applied Physics Letters* 93, 3 (2008).
[58] Omkar Phadke, Sharadindu Gopal Kirtania, Dyutimoy Chakraborty, Suman Datta, and Shimeng Yu. 2024. Suppressed Capacitive Coupling in 2 Transistor Gain Cell With Oxide Channel and Split Gate. *IEEE Transactions on Electron Devices* (2024).
[59] Matt Poremba, Sparsh Mittal, Dong Li, Jeffrey S Vetter, and Yuan Xie. 2015. Destiny: A tool for modeling emerging 3d nvm and edram caches. In *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 1543–1546.
[60] Xiaowei Ren, Daniel Lustig, Evgeny Bolotin, Aamer Jaleel, Oreste Villa, and David Nellans. 2020. Hmg: Extending cache coherence protocols across modern hierarchical multi-gpu systems. In *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 582–595.
[61] Sandeep Kumar Samal, Deepak Nayak, Motoi Ichihashi, Srinivasa Banna, and Sung Kyu Lim. 2016. Monolithic 3D IC vs. TSV-based 3D IC in 14nm FinFET technology. In *2016 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*. IEEE, 1–2.
[62] Mohammad Hossein Samavatian, Hamed Abbasitabar, Mohammad Arjomand, and Hamid Sarbazi-Azad. 2014. An efficient STT-RAM last level cache architecture for GPUs. In *Proceedings of the 51st Annual Design Automation Conference*. 1–6.
[63] Mohammad Hossein Samavatian, Mohammad Arjomand, Ramin Bashizade, and Hamid Sarbazi-Azad. 2015. Architecting the last-level cache for GPUs using STT-RAM technology. *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 20, 4 (2015), 1–24.
[64] Subhali Subhechha, Stefan Cosemans, Attilio Belmonte, Nouredine Rassoul, Shamin Houshmand Sharifi, Peter Debacker, Diederik Verkest, Romain Delhougne, and Gouri Sankar Kar. 2023. Demonstration of multilevel multiply accumulate operations for AiMC using engineered a-IGZO transistors-based 2T1C gain cell arrays. In *2023 IEEE International Memory Workshop (IMW)*. IEEE, 1–4.
[65] Daisaburo Takashima, Shigeyoshi Watanabe, Hiroaki Nakano, Yukihito Oowaki, and Kazunori Ohuchi. 1994. Open/folded bit-line arrangement for ultra-high-density DRAM's. *IEICE transactions on electronics* 77, 5 (1994), 869–872.
[66] Rangharajan Venkatesan, S. G. Ramasubramanian, Swagath Venkataramani, Kaushik Roy, and Anand Raghunathan. 2014. STAG: Spintronic-Tape Architecture for GPGPU Cache Hierarchies. In *Proc. 41st Annu. Int. Symp. Comput. Archit. (ISCA)*. Minneapolis, MN, USA, 253–264.
[67] Rangharajan Venkatesan, Shankar Ganesh Ramasubramanian, Swagath Venkataramani, Kaushik Roy, and Anand Raghunathan. 2014. Stag: Spintronic-tape architecture for gpgpu cache hierarchies. *ACM SIGARCH Computer Architecture News* 42, 3 (2014), 253–264.
[68] Nandita Vijaykumar, Gennady Pekhimenko, Adwait Jog, Abhishek Bhowmick, Rachata Ausavarungnirun, Chita Das, Mahmut Kandemir, Todd C Mowry, and Onur Mutlu. 2015. A case for core-assisted bottleneck acceleration in GPUs: enabling flexible data compression with assist warps. *ACM SIGARCH Computer Architecture News* 43, 3S (2015), 41–53.
[69] Oreste Villa, Mark Stephenson, David Nellans, and Stephen W Keckler. 2019. Nvbit: A dynamic binary instrumentation framework for nvidia gpus. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*. 372–383.
[70] CC Wang, CC Kuo, CH Wu, A Lu, HY Lee, CF Hsu, PJ Tzeng, TY Lee, FR Hou, MH Chang, et al. 2024. P-type SnO semiconductor transistor and application. In *2024 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 1–2.
[71] Jue Wang and Yuan Xie. 2015. A write-aware STTRAM-based register file architecture for GPGPU. *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 12, 1 (2015), 1–12.
[72] Yu Emma Wang, Gu-Yeon Wei, and David Brooks. 2019. Benchmarking TPU, GPU, and CPU platforms for deep learning. *arXiv preprint arXiv:1907.10701* (2019).
[73] Faaiq Waqar, Jungyoun Kwak, Junmo Lee, Minji Shon, Mohammadhosein Gholamrezaei, Kevin Skadron, and Shimeng Yu. 2025. Optimization and Benchmarking of Monolithically Stackable Gain Cell Memory for Last-Level Cache.
[74] Steven JE Wilton and Norman P Jouppi. 2002. CACTI: An enhanced cache access and cycle time model. *IEEE Journal of solid-state circuits* 31, 5 (2002), 677–688.
[75] Edward Wyrwas. 2018. *Body of knowledge for graphics processing units (GPUs)*. Technical Report.
[76] H Ye, J Gomez, W Chakraborty, S Spetalnick, S Dutta, K Ni, A Raychowdhury, and S Datta. 2020. Double-gate W-doped amorphous indium oxide transistors for monolithic 3D capacitorless gain cell eDRAM. In *2020 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 28–3.
[77] Wing-kei S Yu, Ruirui Huang, Sarah Q Xu, Sung-En Wang, Edwin Kan, and G Edward Suh. 2011. SRAM-DRAM hybrid memory with applications to efficient register files in fine-grained multi-threading. In *Proceedings of the 38th annual international symposium on Computer architecture*. 247–258.
[78] Saravanan Yuvaraja, Hendrik Faber, Mritunjay Kumar, Na Xiao, Glen Isaac Maciel García, Xiao Tang, Thomas D Anthopoulos, and Xiaohang Li. 2024. Three-dimensional integrated metal-oxide transistors. *Nature Electronics* 7, 9 (2024), 768–776.
[79] Xia Zhao, Almutaz Adileh, Zhibin Yu, Zhiying Wang, Aamer Jaleel, and Lieven Eeckhout. 2019. Adaptive memory-side last-level GPU caching. In *Proceedings of the 46th international symposium on computer architecture*. 411–423.