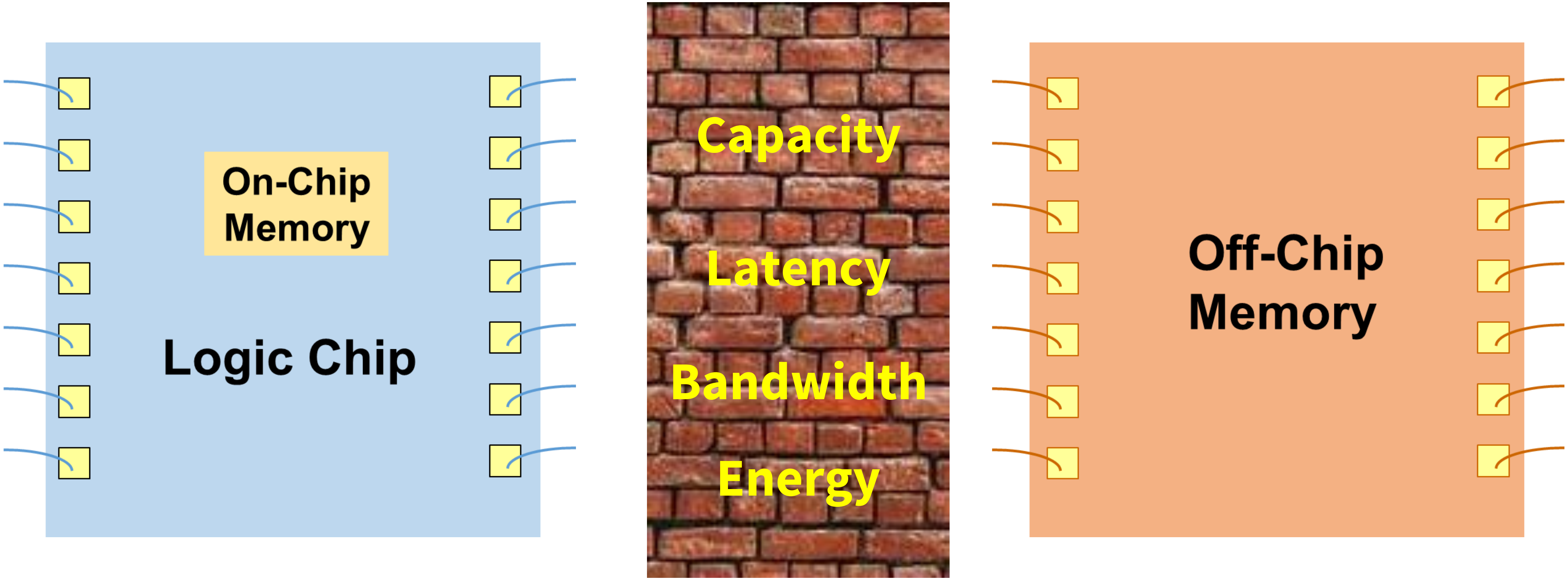
OpenGC: An Open-Source Gain Cell Compiler

Xinxin Wang

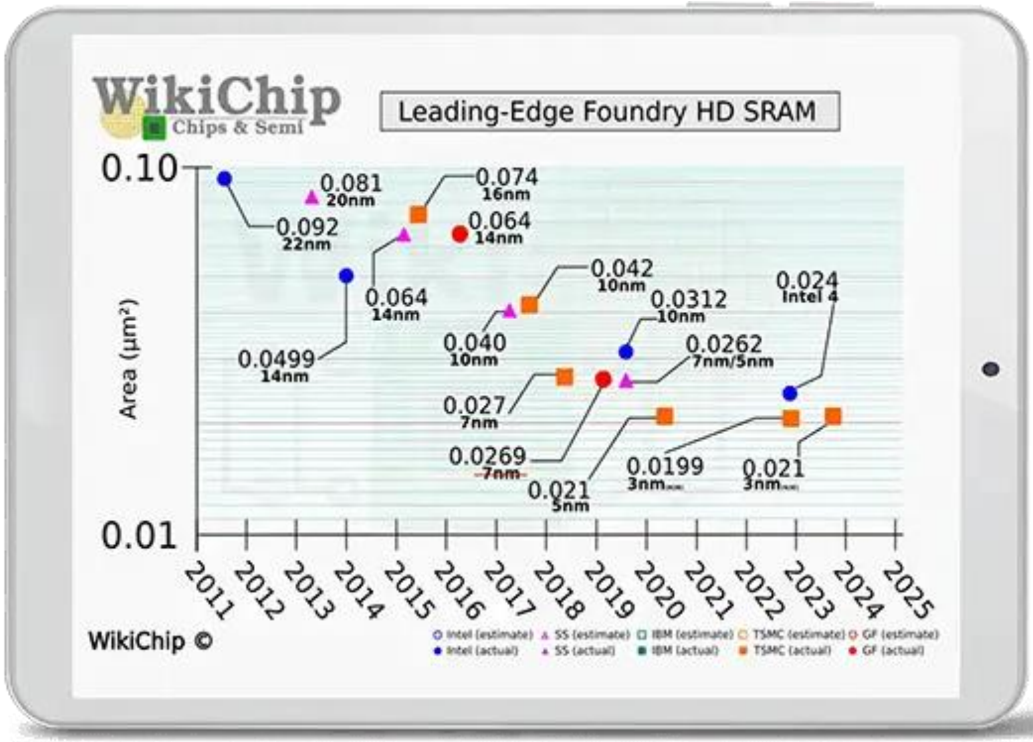
01/10/2025

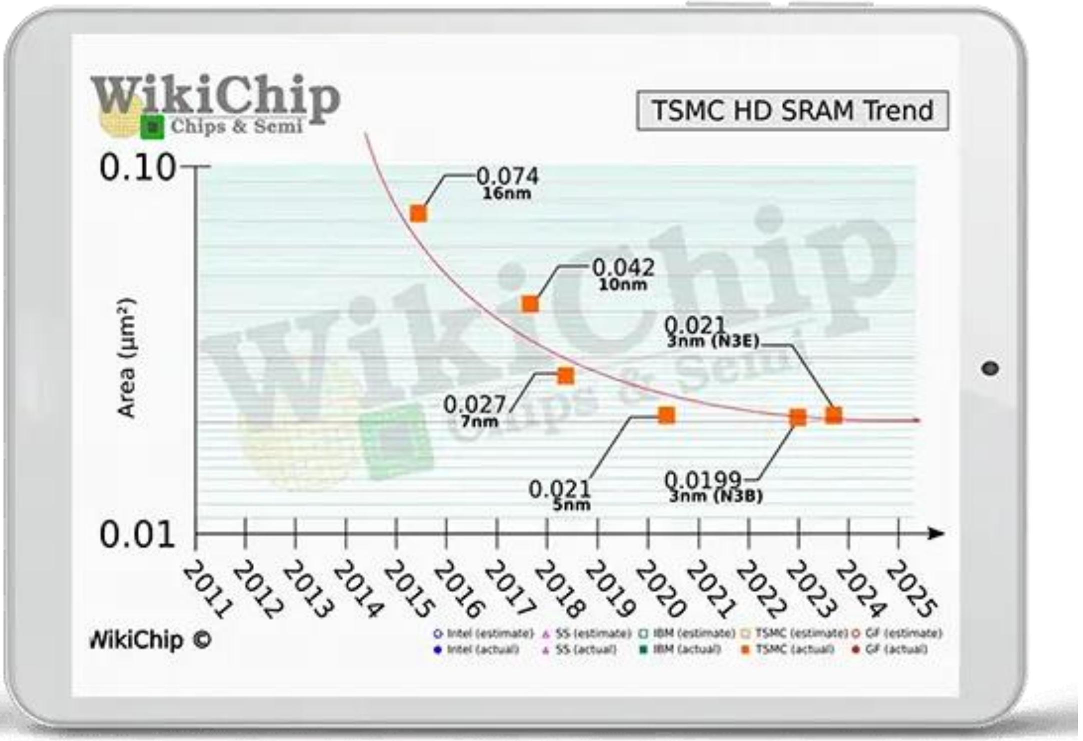
Memory Wall



Slide by Shuhan Liu

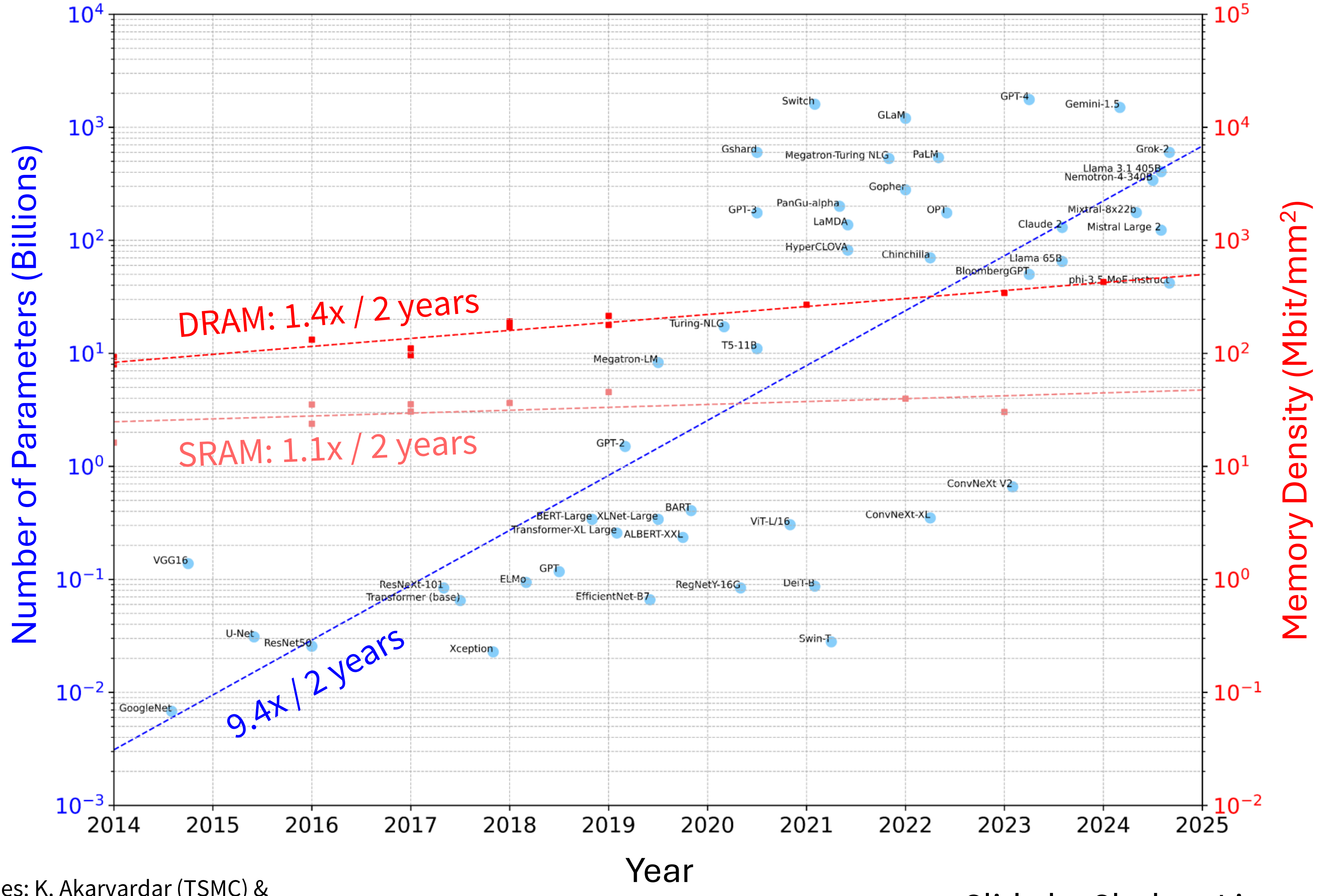
SRAM scaling miniaturization





Source: https://raaam-tech.com/technology/

Memory Needs Outpace Memory Advances



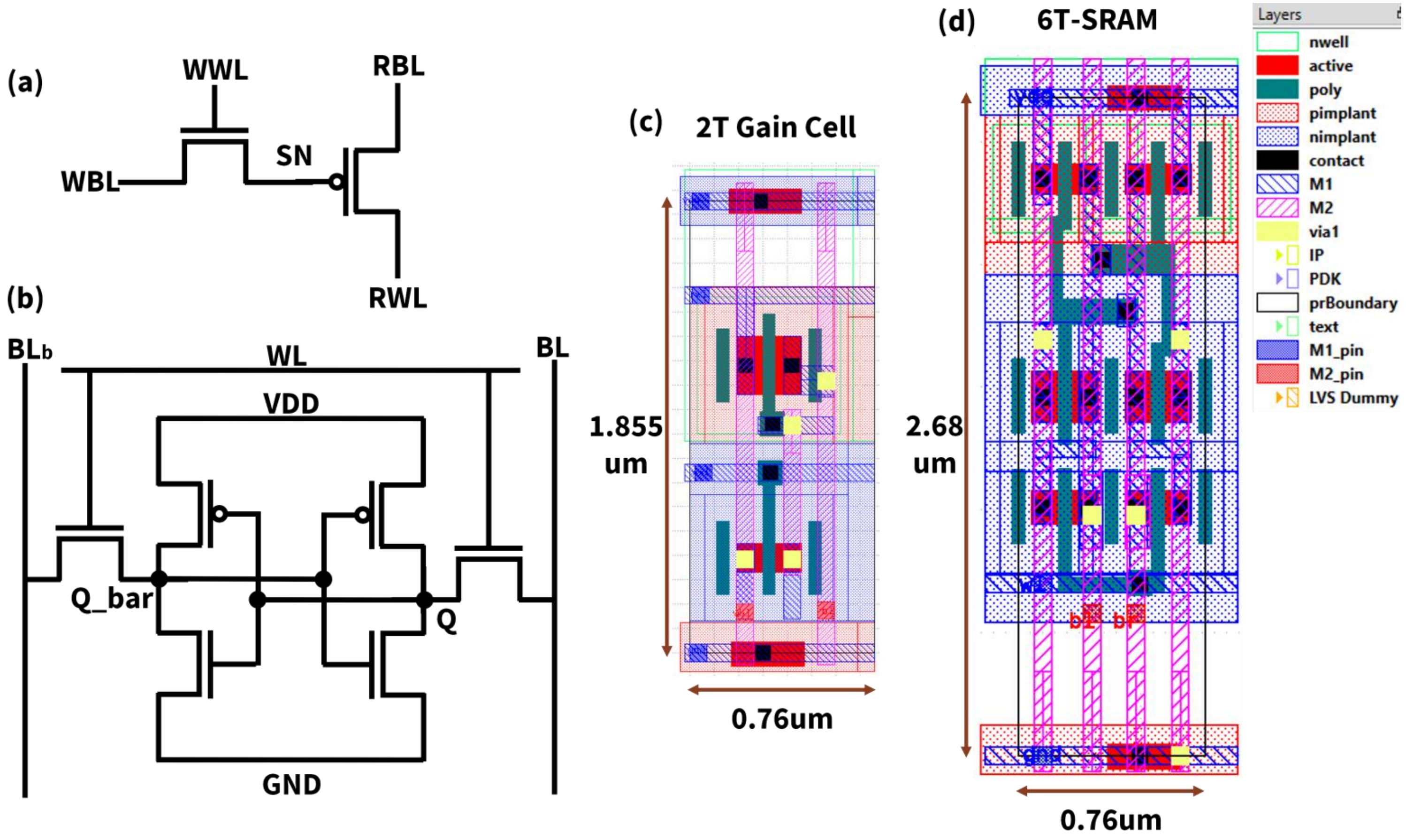
We need innovations for high-density on-chip memory

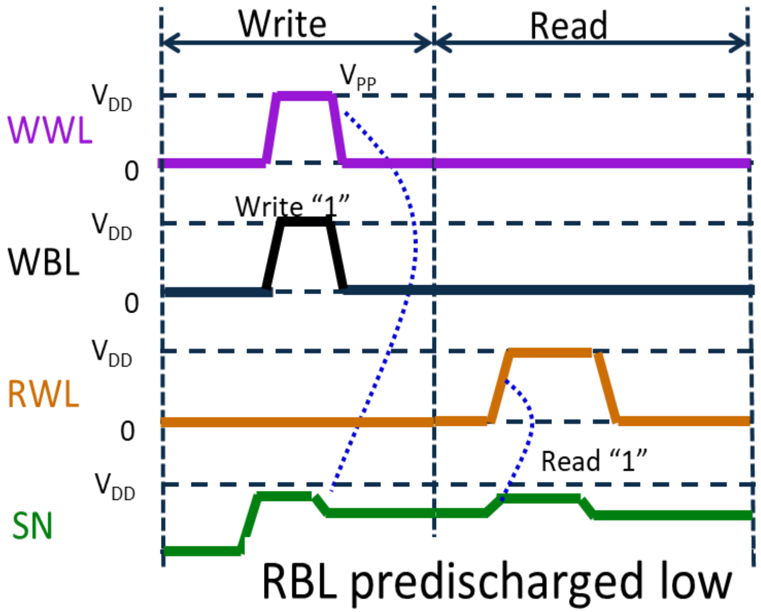
Sources: K. Akarvardar (TSMC) &
https://nano.stanford.edu/downloads/technology-integration-trend

Slide by Shuhan Liu

Stanford University

Gain Cell memory: higher density than SRAM

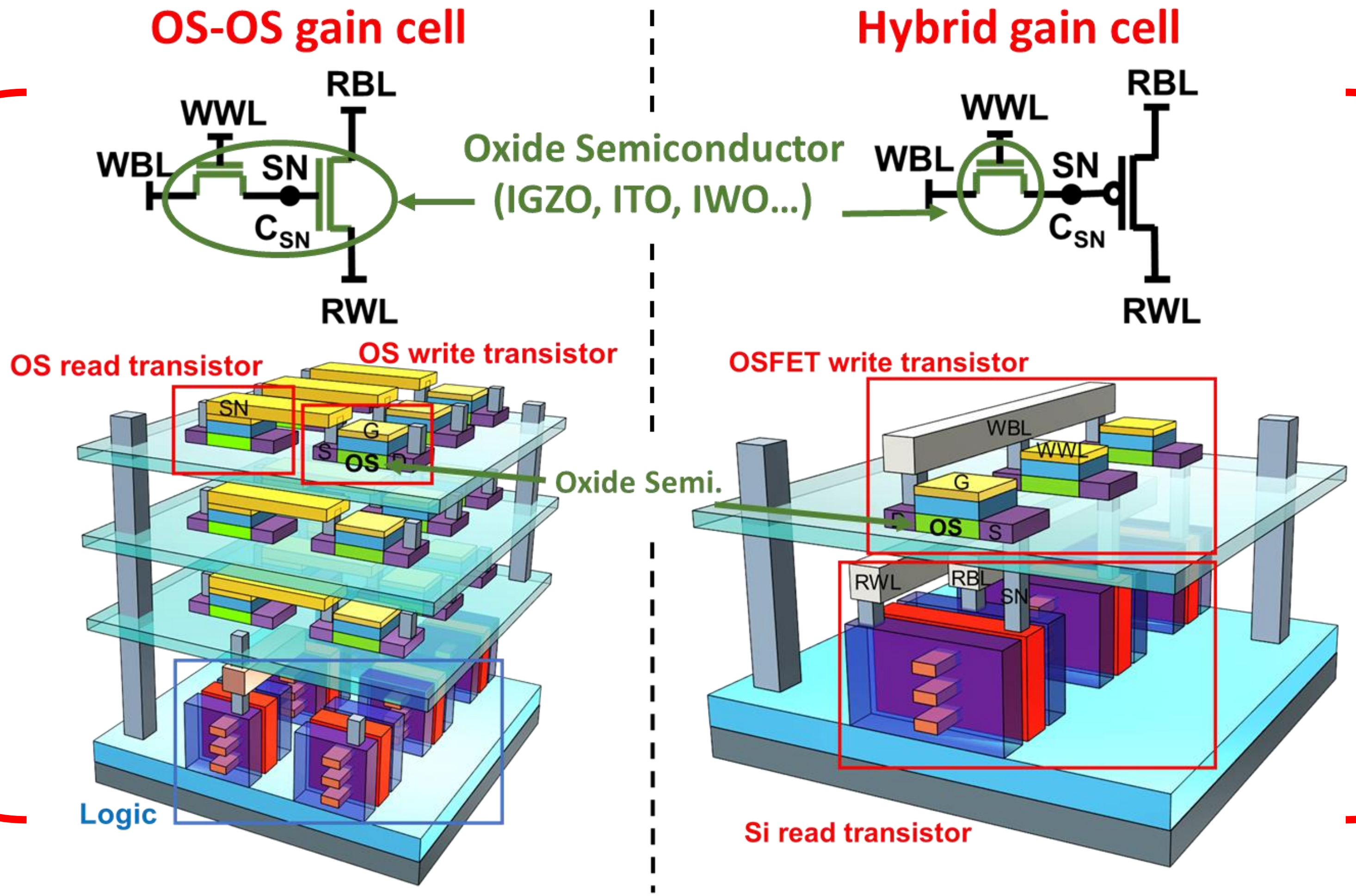




Koustav Jana,… H.-S. P. Wong, *ITC*, poster # 05_1003, 2024, S. Liu …et al., *EDL*, 2024.

Gain Cell memory: BEOL implementation

Higher capacity



Higher speed

Shuhan Liu, …, H.-S. Philip Wong, IEDM 2023, T-ED 2024, VLSI 2024

Slide by Shuhan Liu

To enable fast, accurate, customizable, and optimized Gain Cell bank generation and performance simulation:

We need a Gain Cell compiler

M. R. Guthaus, et al., “OpenRAM: An Open-Source Memory Compiler”, ICCAD, 2016

Related work

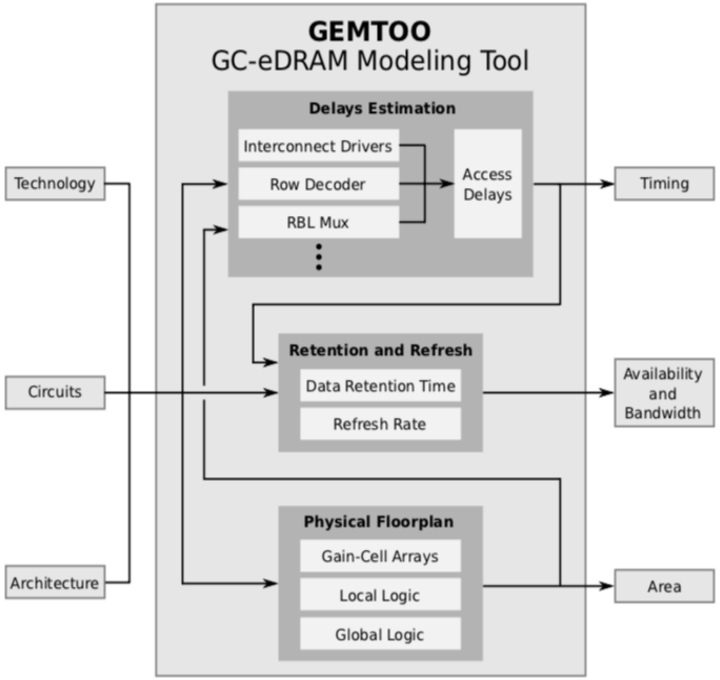
- RAAAM GCRAM

- Up-to **2X Higher density** vs. SRAM
- Up-to **50% area reduction** vs. SRAM
- Standard **SRAM interface**
- **Extended interface options** vs. SRAM
- **Single/two** ported
- Up to **2Mbit** instances
- **Standard CMOS** process
- **Single cycle** operation
- **Customizable**

Limitations:

- 3T GCRAM
- commercialized
- not open-access

- GEMTOO Simulator



Limitations:

- No netlist and layout
- No power evaluation
- Very rough delay estimation

- OpenRAM Compiler

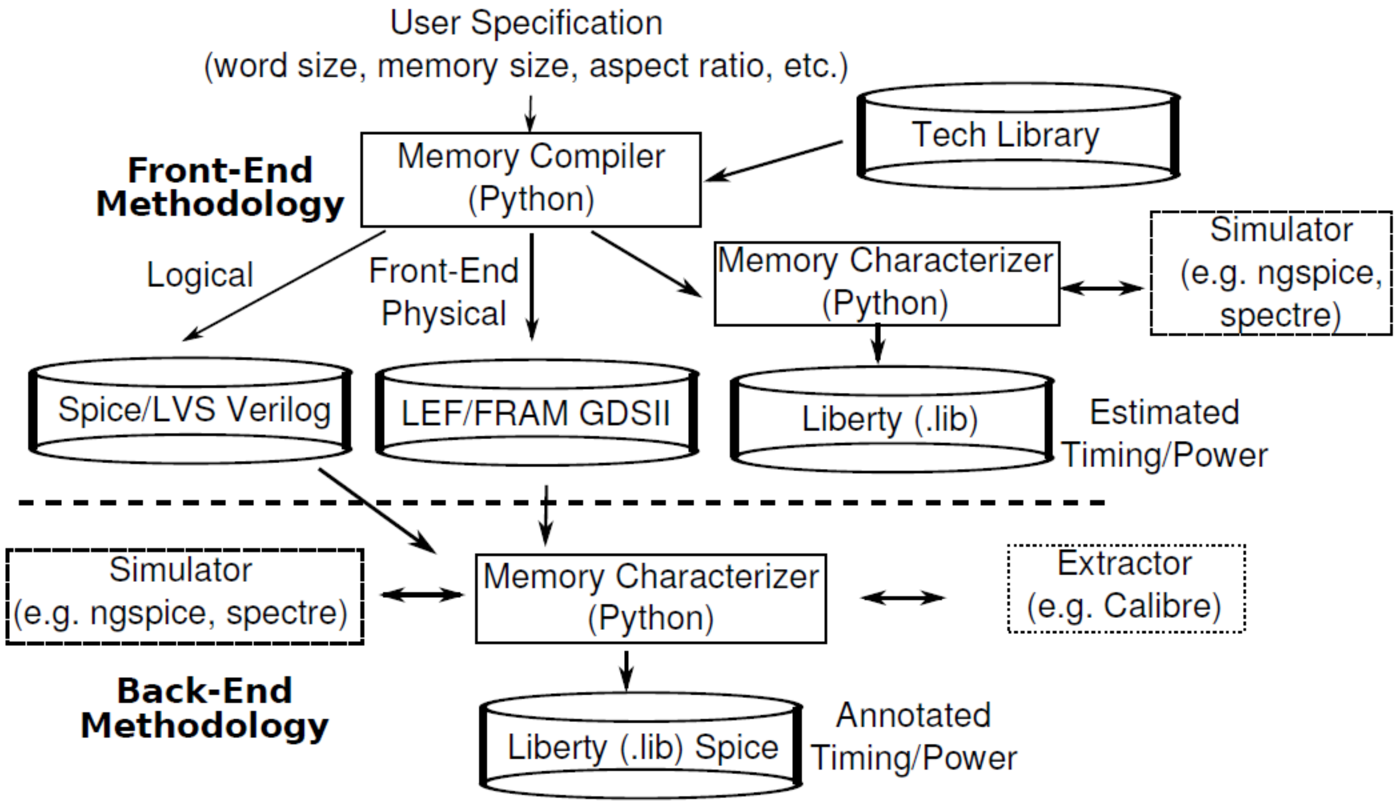
- Open-source SRAM compiler
- Support open-source PDKs
- SRAM bank netlist & layout generation
- Performance simulation

Limitations:

- No Gain Cell support
- No commercial PDK support

OpenRAM SRAM compiler

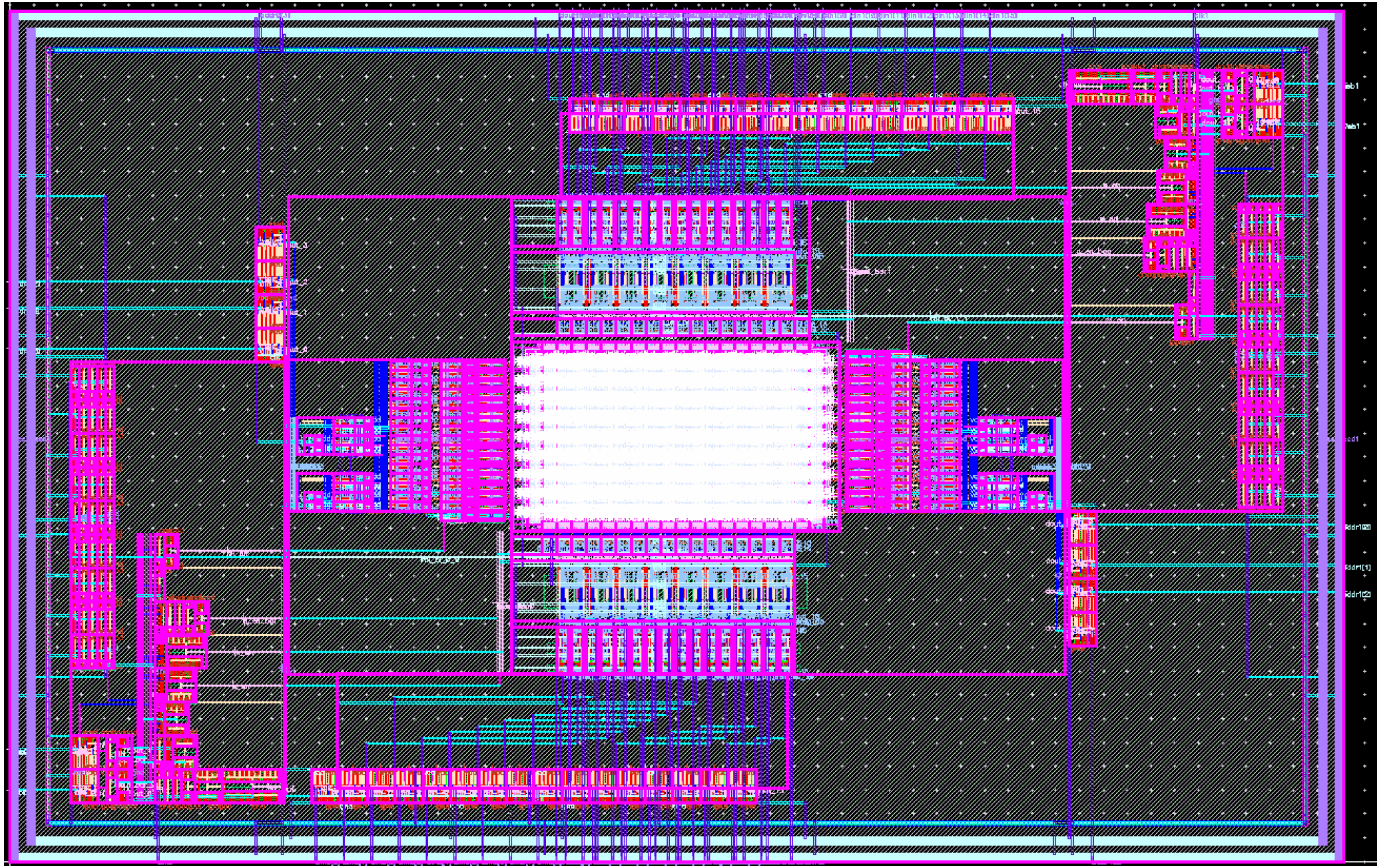
- Front-end: Create Netlist and Layout, DRC & LVS check;
- Back-end: Perform simulations on these generated files



M. R. Guthaus, et al., “OpenRAM: An Open-Source Memory Compiler”, ICCAD, 2016

OpenRAM SRAM compiler

- Example: 16x16 SRAM macro (file:///E:/OneDrive%20-%20Stanford/CNT-ITO/OpenRAM/sram_16x2.html)



M. R. Guthaus, et al., “OpenRAM: An Open-Source Memory Compiler”, ICCAD, 2016

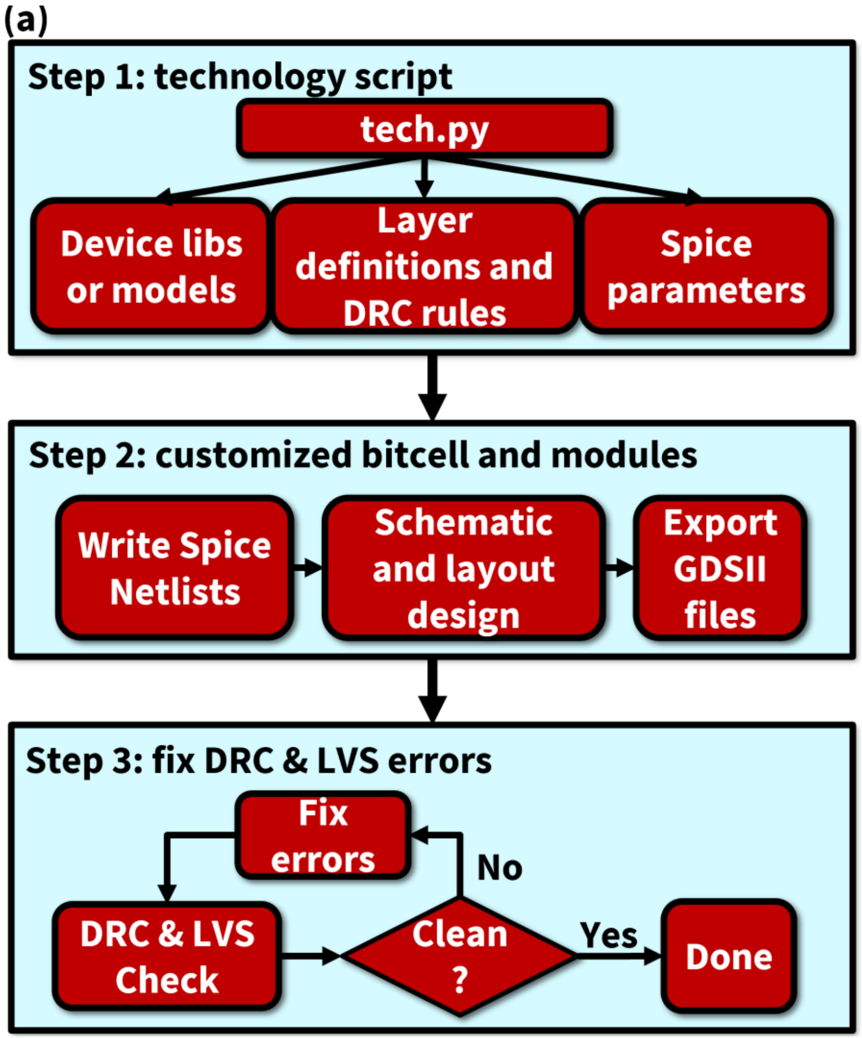
OpenRAM SRAM compiler

- Limitations:
  - Support NCSU FreePDK 45nm, MOSIS 0.35um (SCN4M_SUBM), Skywater 130nm (sky130), no advanced tech nodes
  - Only SRAM is supported
  - Only one macro architecture is supported

M. R. Guthaus, et al., “OpenRAM: An Open-Source Memory Compiler”, ICCAD, 2016

Methodology to extend OpenRAM functionality

- Step 1: Porting to new PDKs



(a)

Step 1: technology script

tech.py

Device libs or models

Layer definitions and DRC rules

Spice parameters

Step 2: customized bitcell and modules

Write Spice Netlists

Schematic and layout design

Export GDSII files

Step 3: fix DRC & LVS errors

Fix errors

DRC & LVS Check

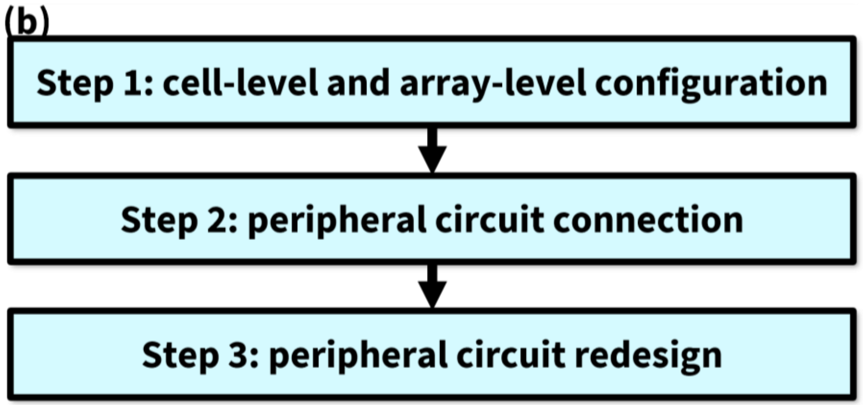
Clean ?

No

Yes

Done

- Step 2: Adding new memory technologies



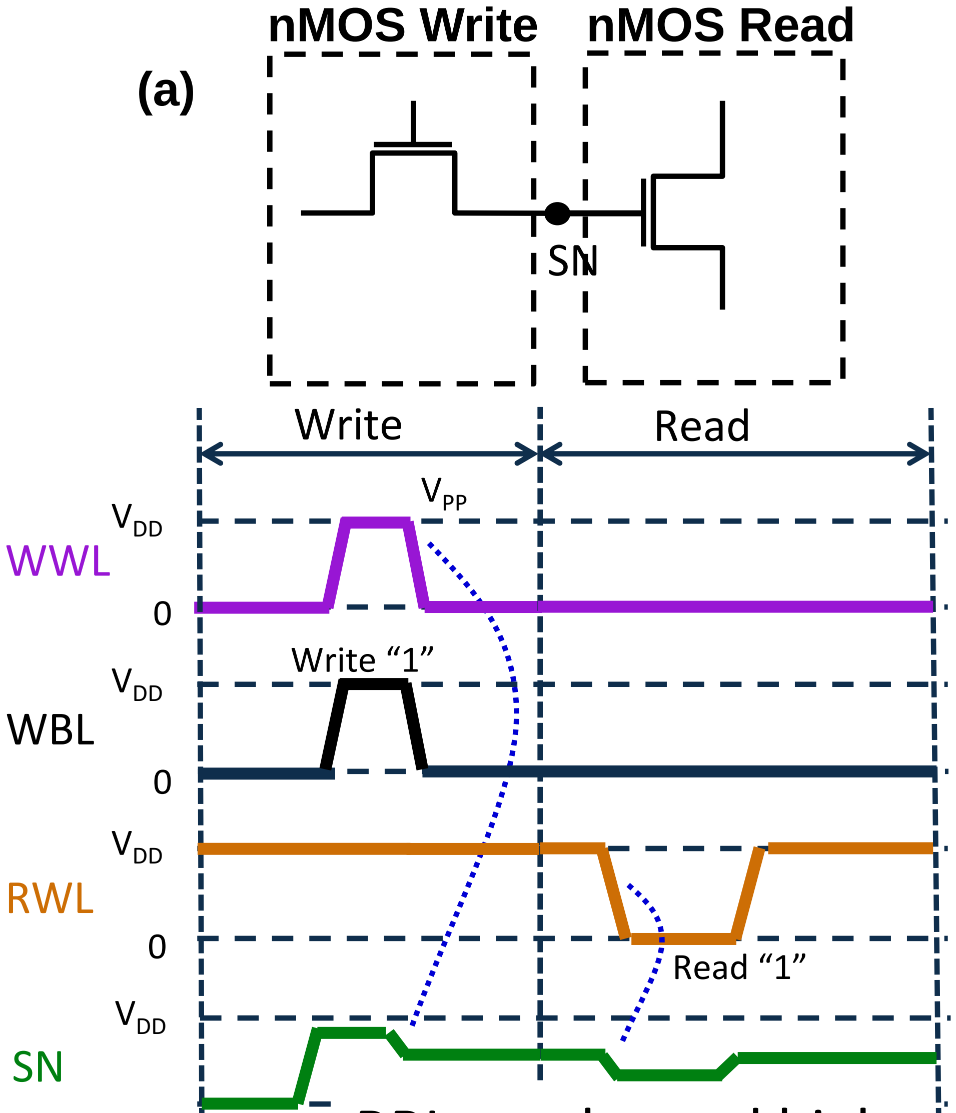
(b)

Step 1: cell-level and array-level configuration

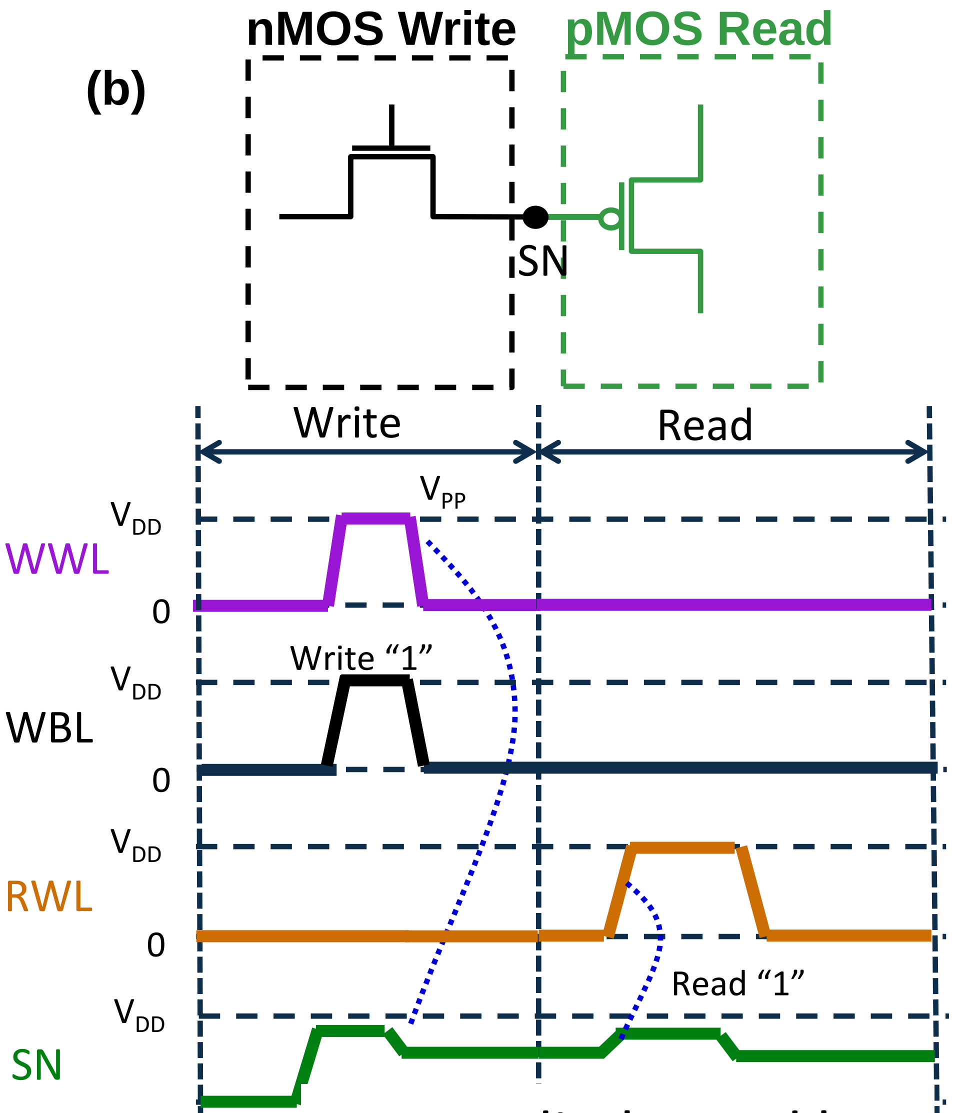
Step 2: peripheral circuit connection

Step 3: peripheral circuit redesign

Gain Cell memory operations



RBL precharged high

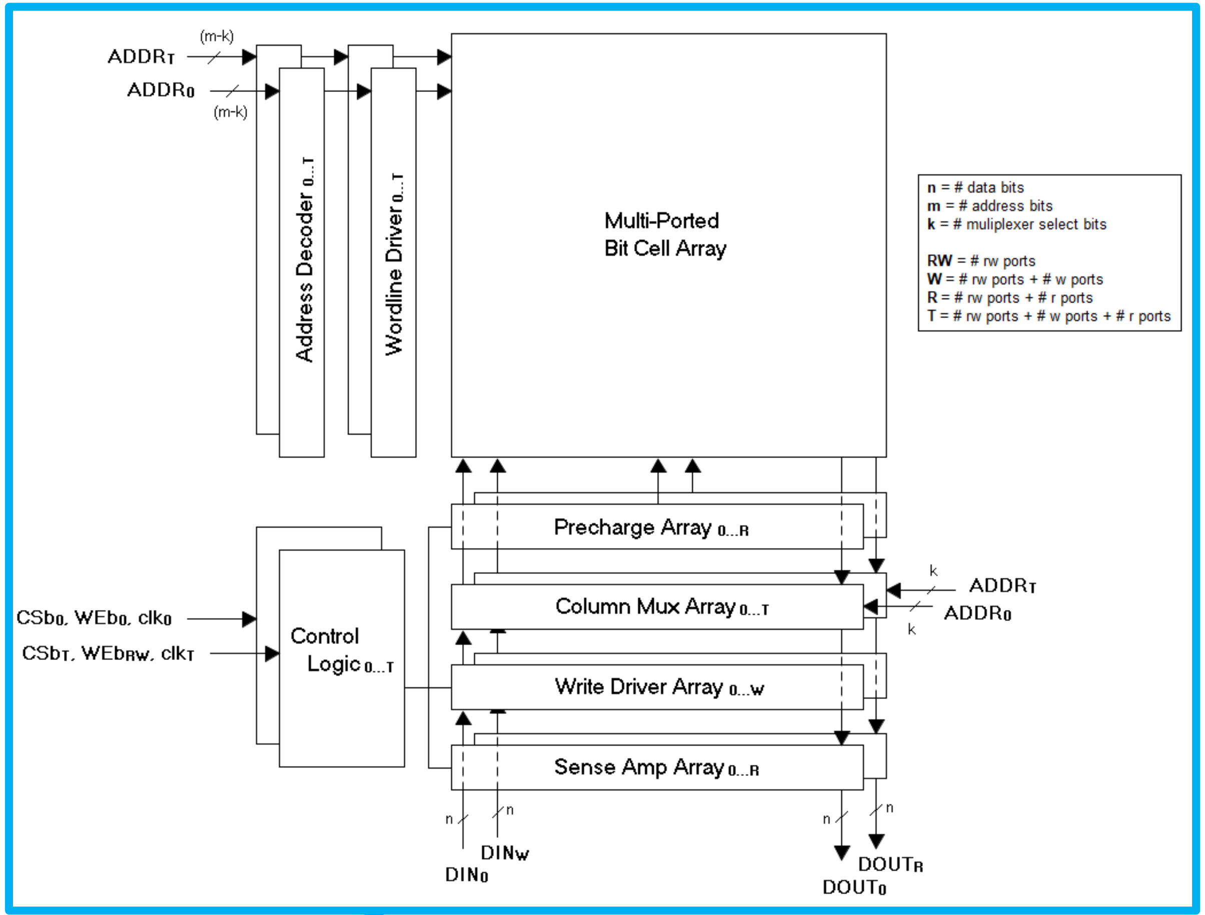


RBL predischarged low

Koustav Jana,… H.-S. P. Wong, *ITC*, poster # 05_1003, 2024, S. Liu …et al., *EDL*, 2024.

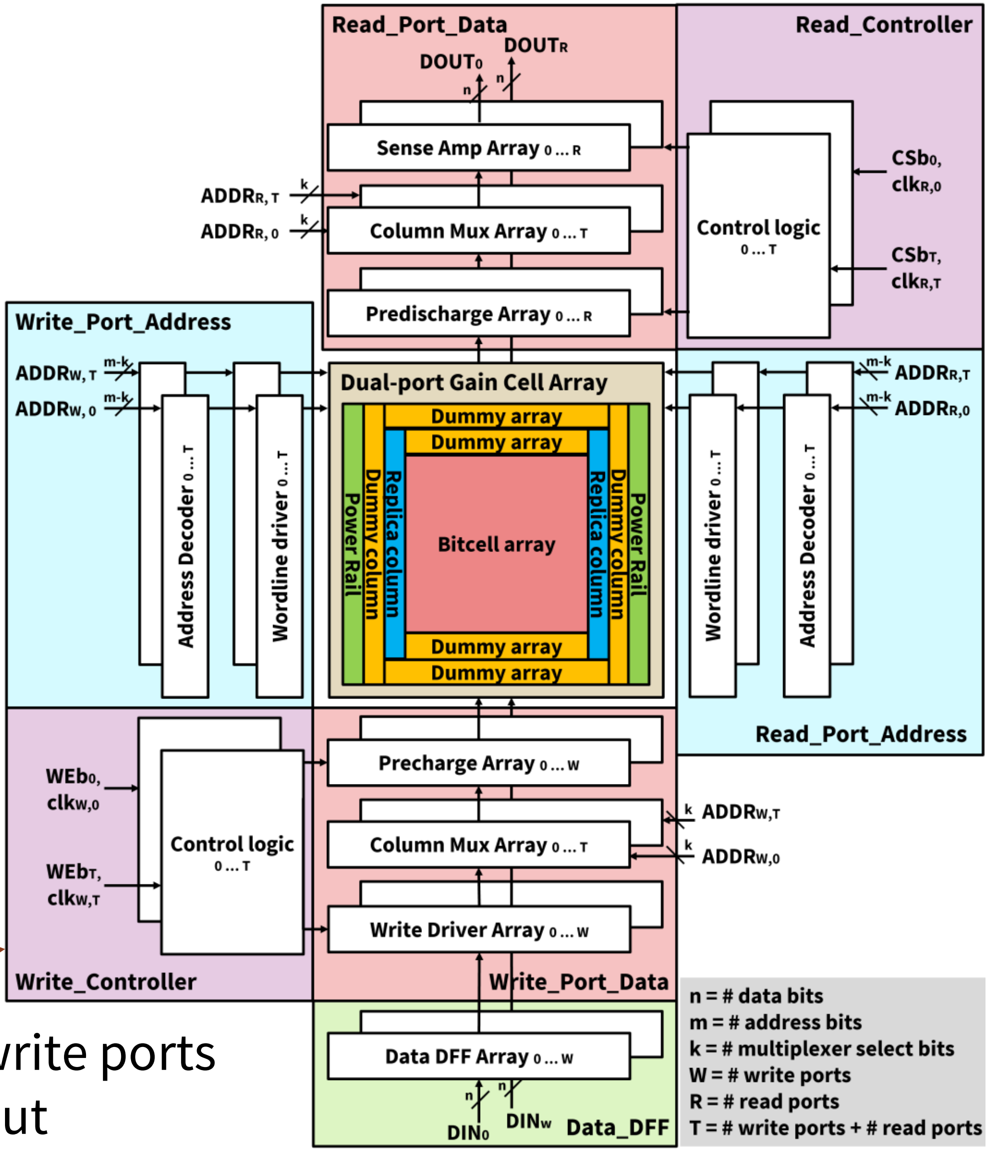
Stanford University

OpenGC: bank architecture



n = # data bits
m = # address bits
k = # muliplexer select bits

RW = # rw ports
W = # rw ports + # w ports
R = # rw ports + # r ports
T = # rw ports + # w ports + # r ports



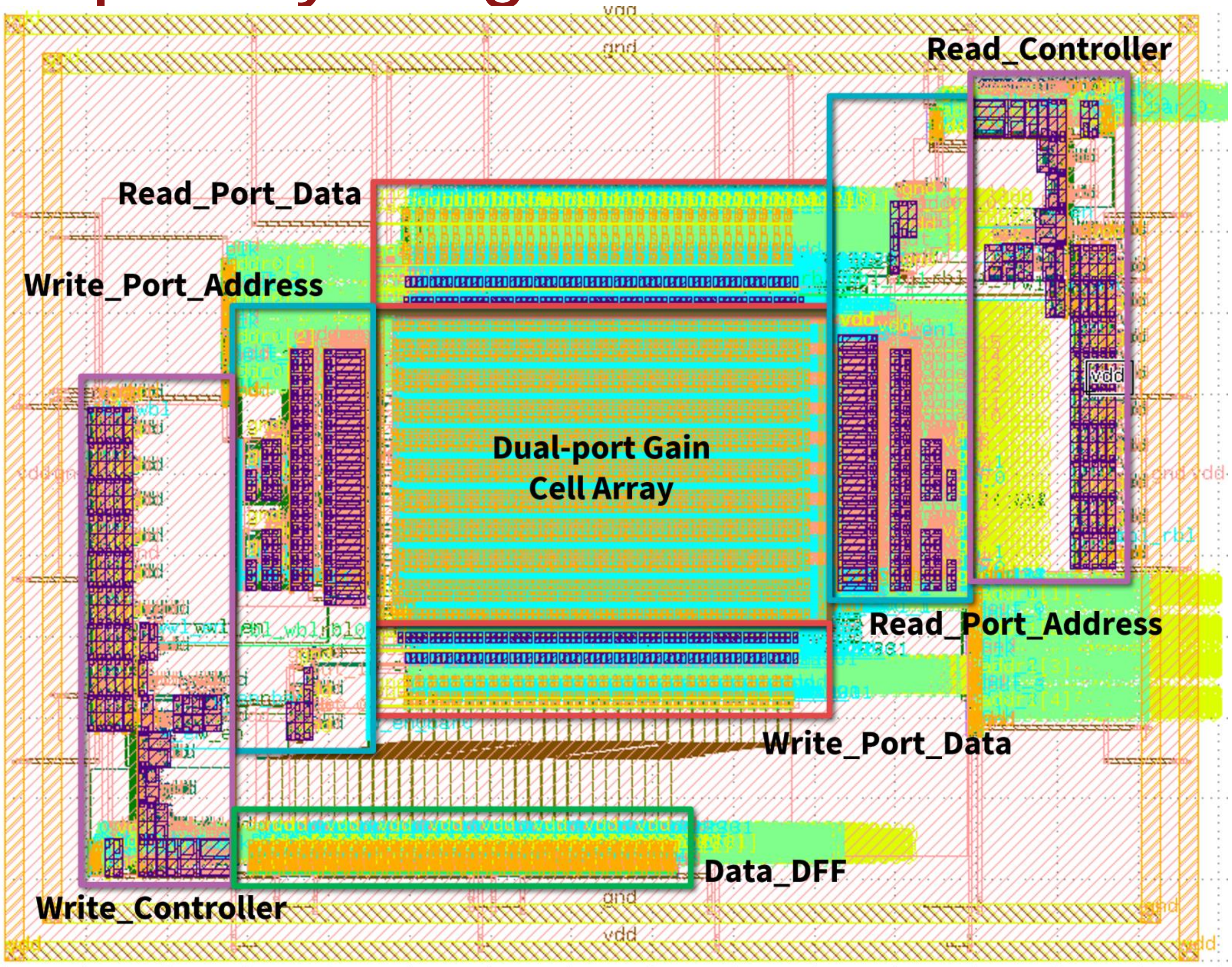
n = # data bits
m = # address bits
k = # multiplexer select bits
W = # write ports
R = # read ports
T = # write ports + # read ports

SRAM bank → **Gain Cell bank**

- separate read and write ports
- single-ended read out
- predischarge array for read port

OpenGC: Example layout generation

A 32x32 Gain Cell bank generated by OpenGC



Gain Cell vs. SRAM: area

- GC array area < SRAM array area

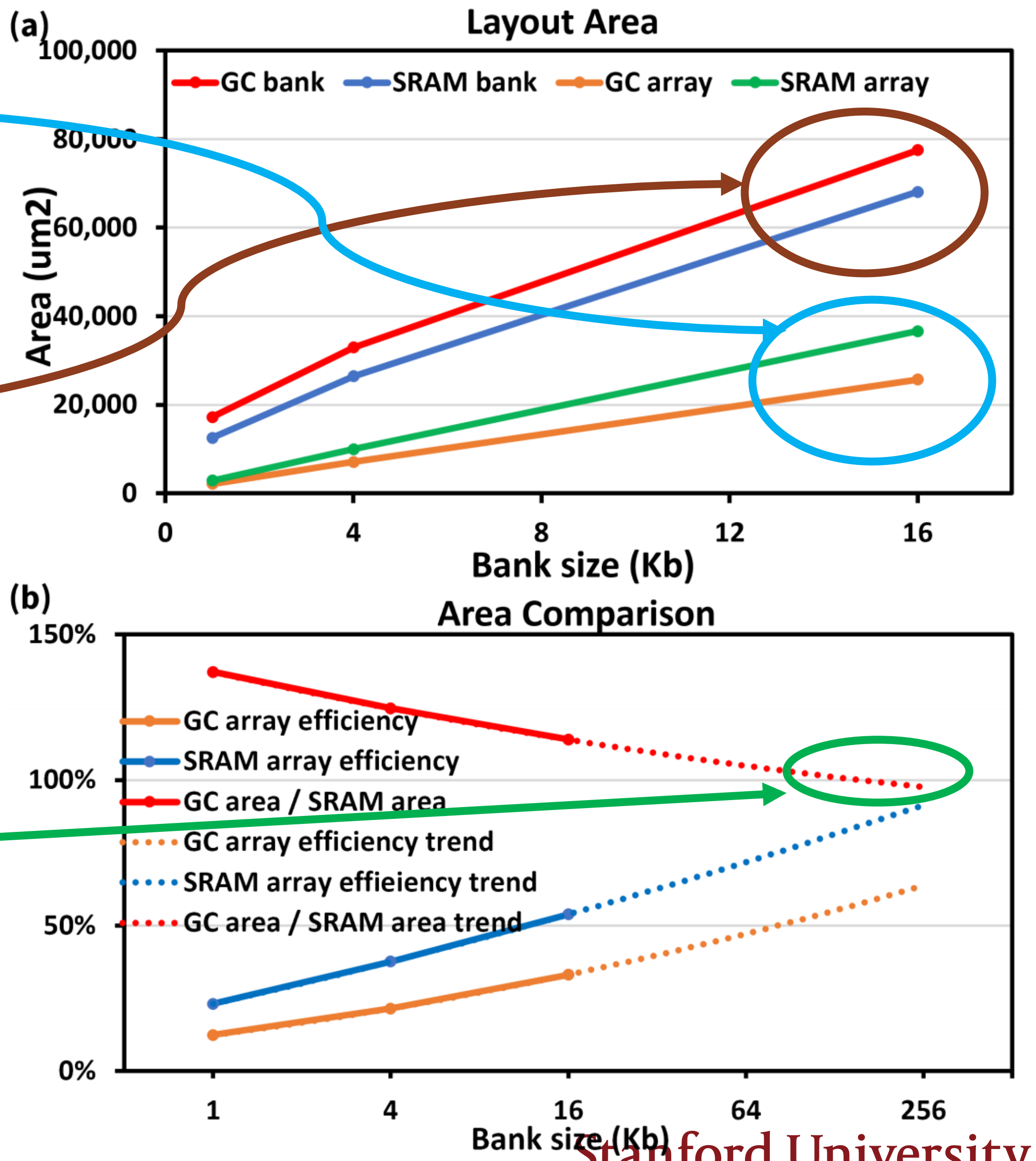
  → GC cell size < SRAM cell size

- GC bank area > SRAM bank area for small bank size

  → GC has separate peripheral circuits for different ports

- GC bank area < SRAM bank area for large bank size

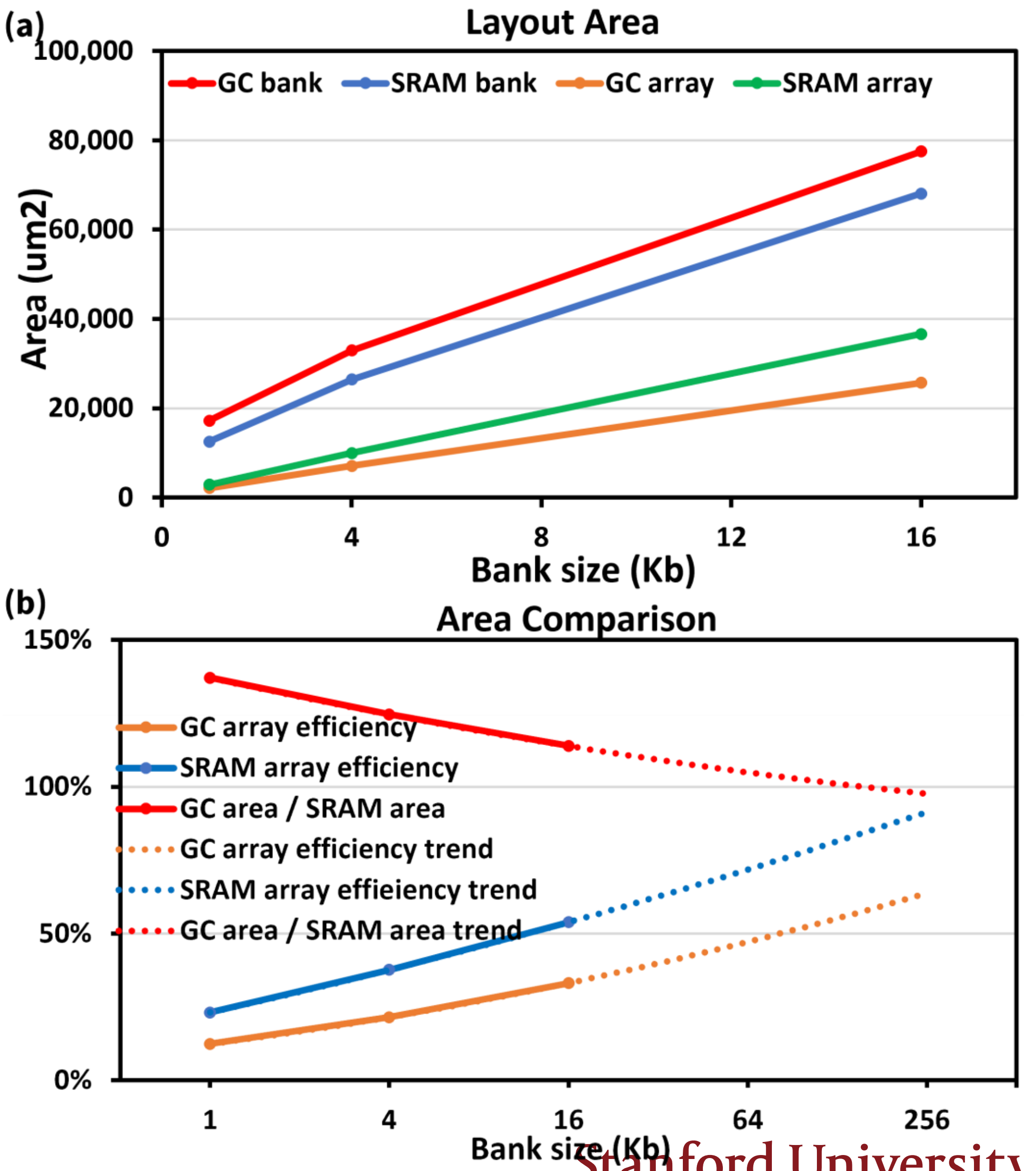
  → Peripheral circuits are amortized



Gain Cell vs. SRAM: area

- Dual-port peripheral of GC bank allows simultaneous read and write operations, resulting in high bandwidth

- Dual-port SRAM bank area is ~2x of single-port SRAM bank area, which is larger than GC bank.



Gain Cell vs. SRAM: delay and power

- GC frequency < SRAM frequency

  → GC uses single-ended readout

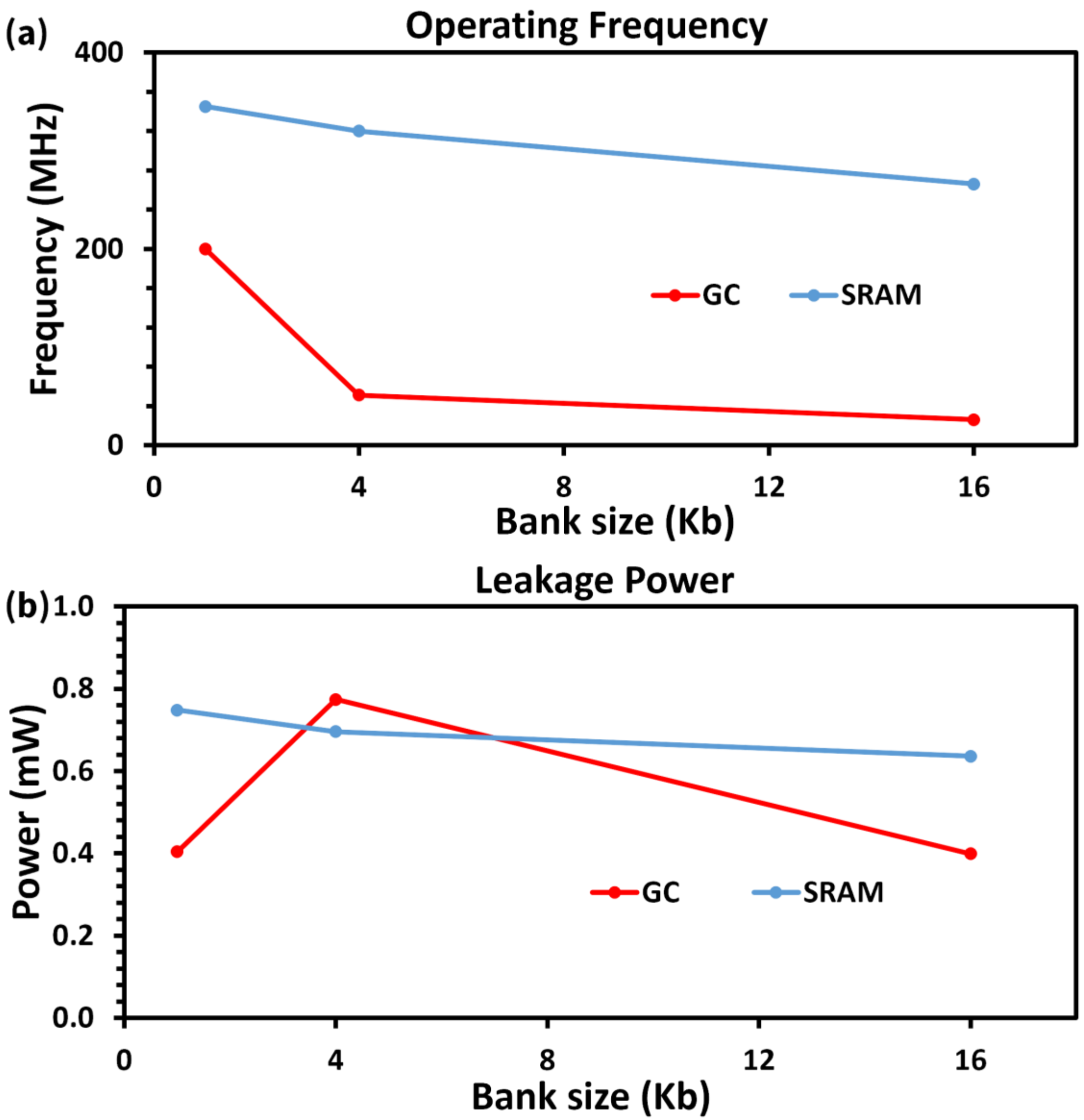
  → GC storage node voltage is Vdd-Vth when writing “1”

- GC leakage power < SRAM leakage power

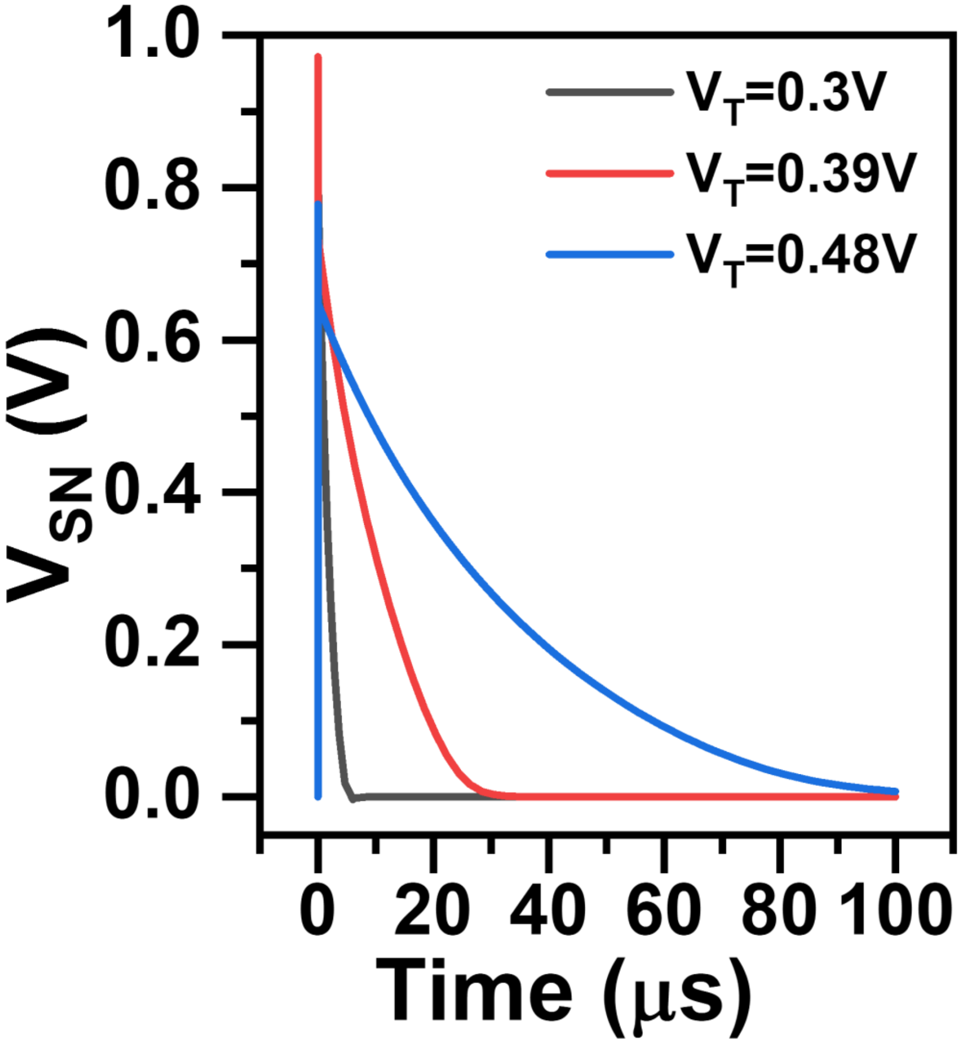
  → There is no direct path from Vdd to GND in GC

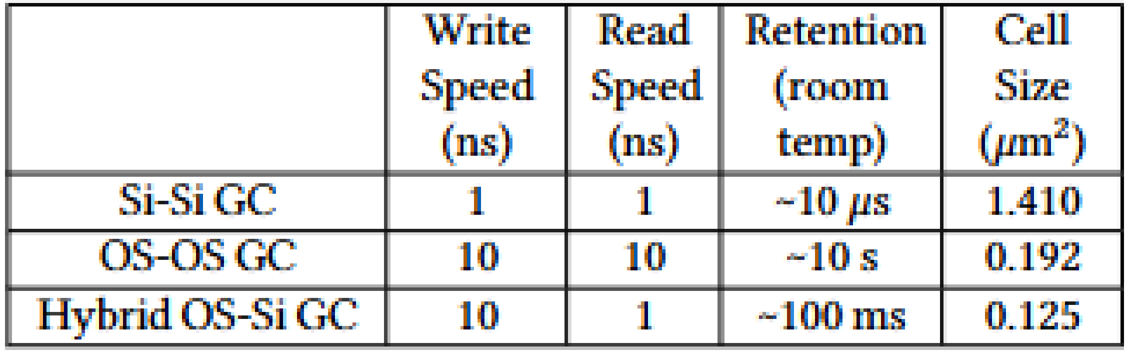
- An abrupt change of GC freq and power from 1 Kb to 4 Kb

  → Increase in delay chain stages



Gain Cell retention modulation





	Write Speed (ns)	Read Speed (ns)	Retention (room temp)	Cell Size (μm^2)
Si-Si GC	1	1	~10 μs	1.410
OS-OS GC	10	10	~10 s	0.192
Hybrid OS-Si GC	10	1	~100 ms	0.125

Table 1: Simulation results on 40 nm node 2T GC variants.

By adjusting the transistor design (like Vth and channel material), the retention can be tuned to accommodate for different applications (such as activation caches and weight memory in AI inference).

Next Steps

- Add BEOL layers to PDK tech files
- Add support for hybrid Gain Cell and OS-OS Gain Cell
- Add level shifters and multiple variants of sense amp & drivers

Summary

- We developed OpenGC to enable fast, accurate, customizable, and optimized Gain Cell bank design as high-density on-chip memory
- We introduced a standard methodology for porting OpenRAM compiler to new PDKs and memory technologies
- OpenGC supports Gain Cell bank design generation with TSMC N40 PDK and precise Spice simulations for performance evaluation
- By following the proposed methodology, OpenGC can be extended to support additional types of Gain Cell memory and other PDKs