# Logic-compatible Gain Cell eDRAM
# A Real Alternative to SRAM

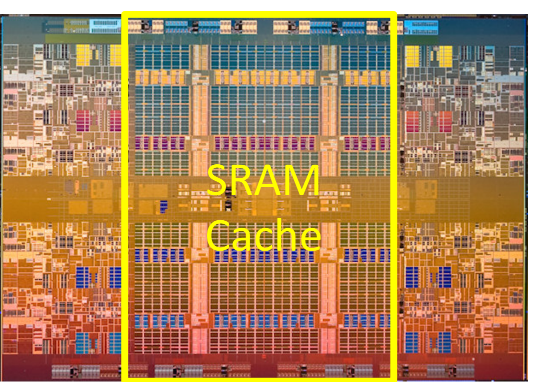
Andrea Bonetti[1], Robert Giterman[2], Adam Teman[2], Alex Fish[2],
Pascal Meinerzhagen[1], **Andreas Burg**[1]
[1]Telecommunications Circuits Laboratory, EPFL
[2]Bar-Ilan University

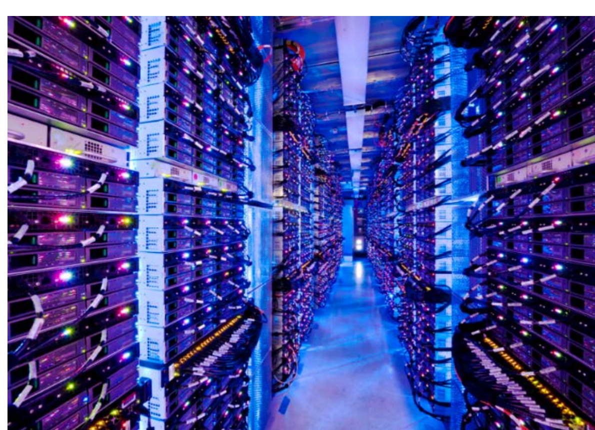
# Memories Are the Limiting Factor for Cost and Energy Efficiency

- **On-chip memories** have a poor area density and **often dominate chip area and cost** in many computing systems
- **Memory** often accounts for **>50% of the active power** and for **100% of the power during sleep/standby periods** in low-power systems

High-Performance Computing in PCs and Data Centers

(Intel 8 core Xeon X7560)



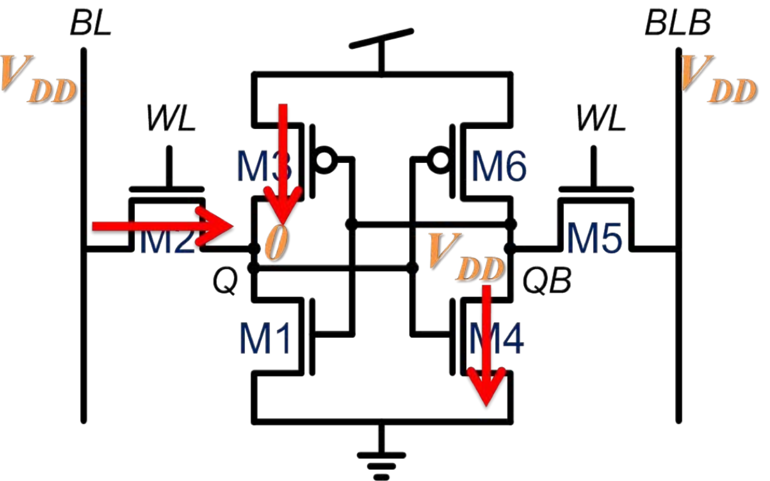
Thousands of cores in a data center consume mega Watts of power

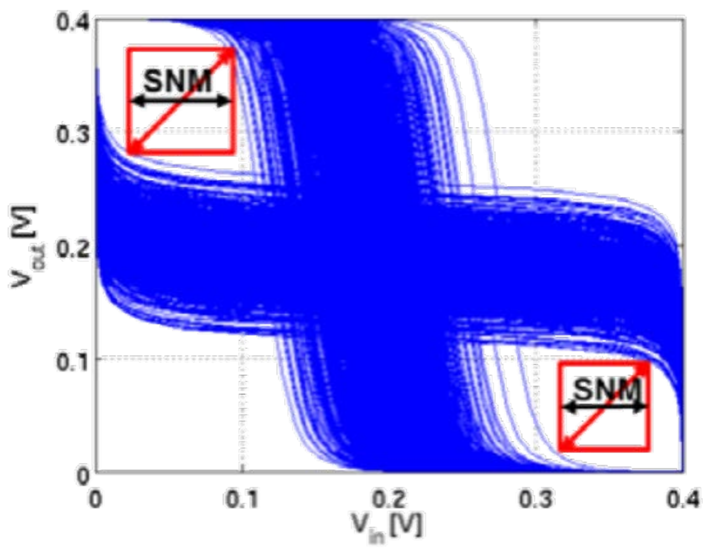
ULP energy-autonomous and wearable devices

Operation for up to 10+ years on single battery charge

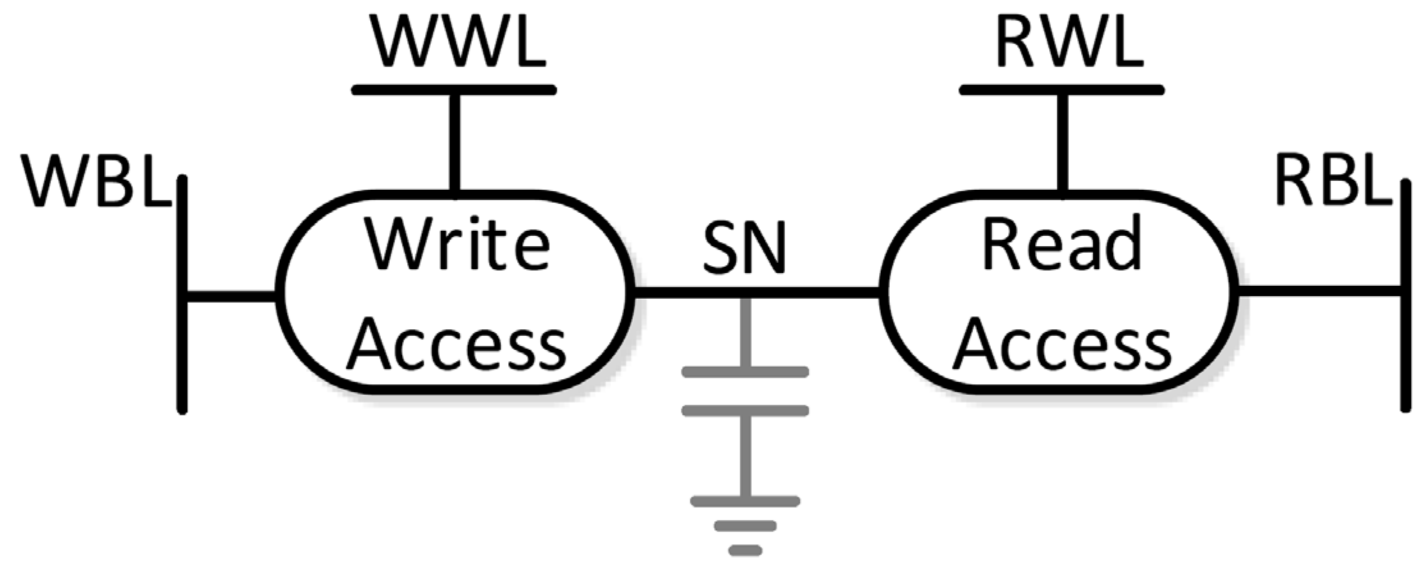
# On-Chip SRAM Limitations

- Almost all integrated circuits rely on the standard 6T-SRAM bitcell for on-chip memory
- However, it incurs several limitations:
  - Six transistors per bit (6T)
  - Considerable static power due to leakage
  - Ratioed operation limits voltage scaling
  - Two port operation requires 8T



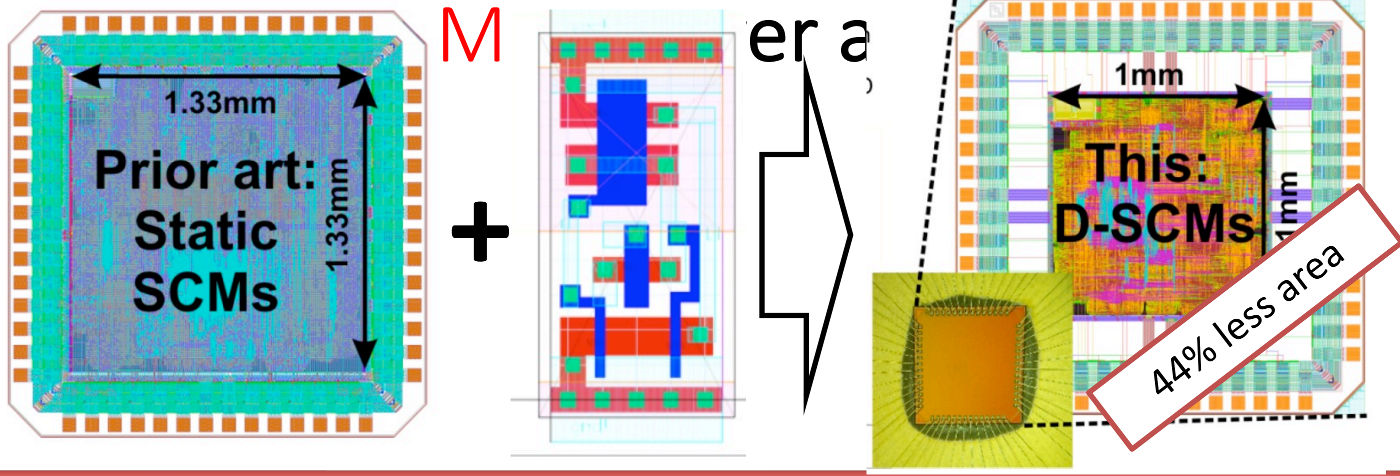
# Basic Concept of Gain Cell eDRAM (GC-eDRAM)

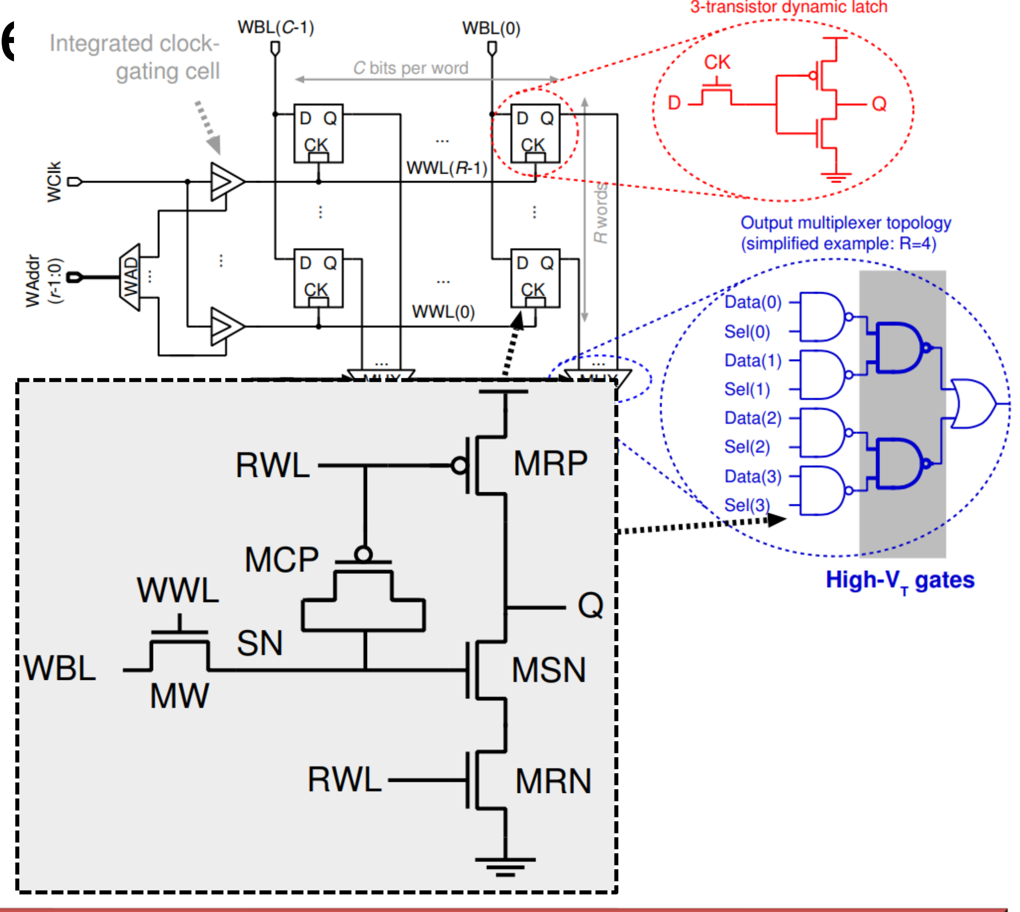



Data is stored as charge with read- and write-access networks with 1-2 transistors each

# Refresh-free GC-eDRAM for DSP Applications

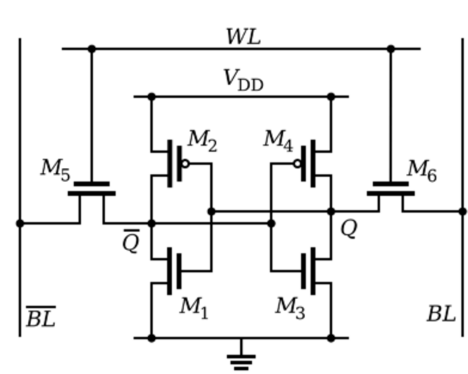
- **DSP applications require many small memories** for short-term storage
- **Standard-cell memories** offer ~**40% powe**
  macros and enable voltage-scaling
  - However, with considerable area overhead
- Replacing latches with dynamic



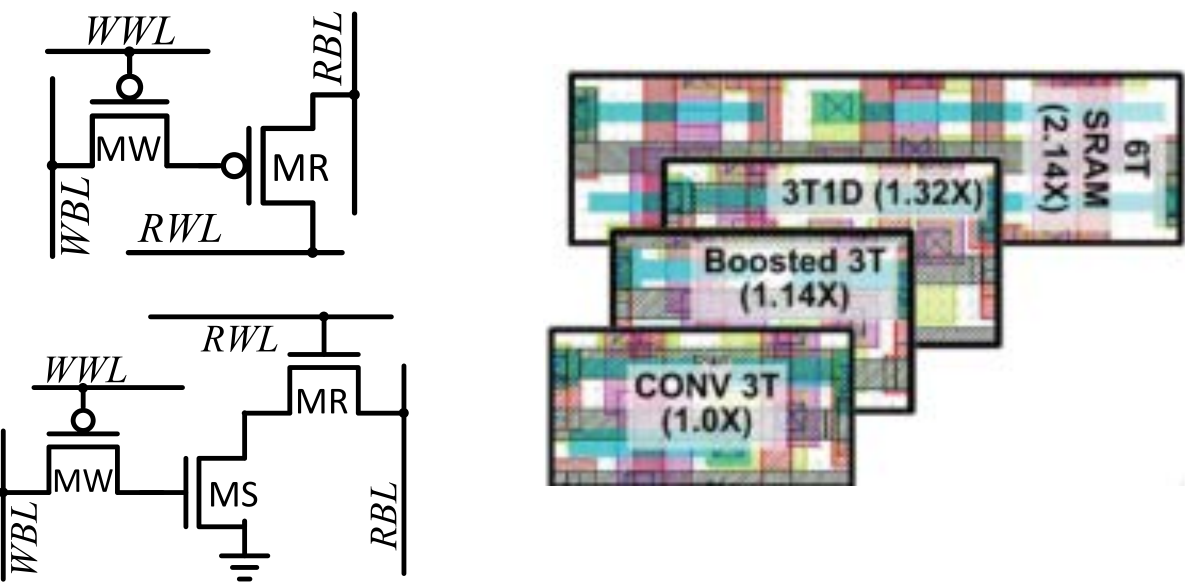



# Gain Cell Embedded DRAM (eDRAM)

- Most compact realization: 2-3 transistors (2T and 3T gain cell)
  - Bit-cell area offers almost a 2x advantage over SRAM



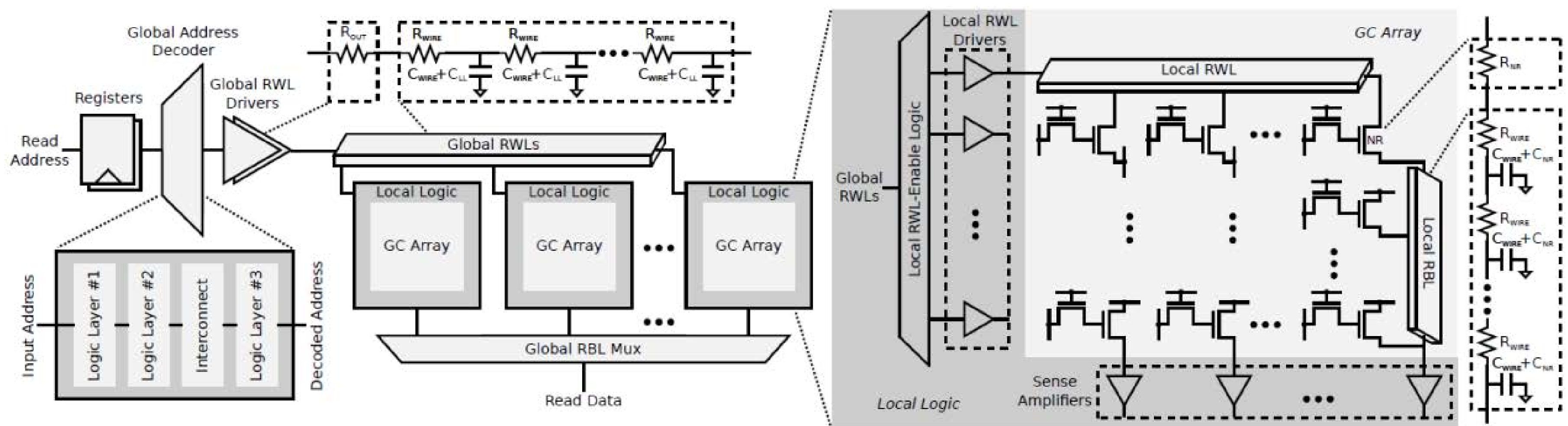

6T-SRAM macros with an area of **130F$^2$ per bit**




2T and 3T Gain Cell with **~70F$^2$ per bit**

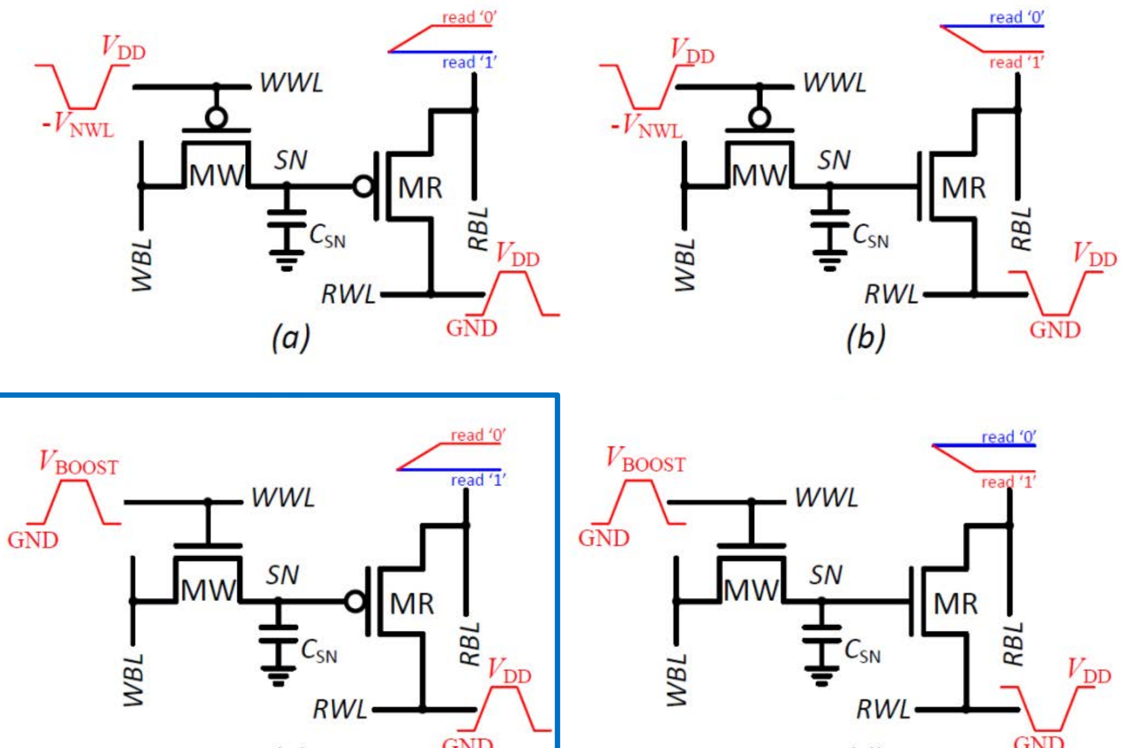
# Gain Cell Embedded DRAM (eDRAM)

- Gain cells can be arranged in compact arrays with simple read- and write-access circuitry
  - Hierarchical arrangement of sub-arrays for large memories
  - Simple area-efficient peripherals
- Gain cell arrays are not more complex than SRAM arrays



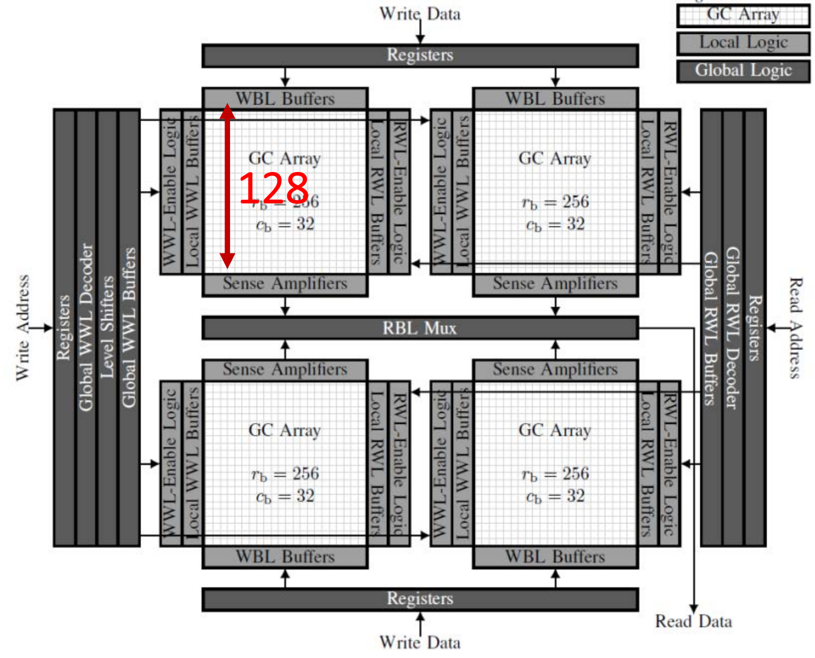
# 2T Gain Cell eDRAM: Basic Operating Principle

- Write port (WWL & WBL), storage cap, and read port (RWL & RBL)
  - Different combinations of PMOS and NMOS transistors
  - Use of different threshold options
- **Write operation:**
  - Boosted WWL, above VDD for NMOS, below $V_{SS}$ for PMOS
- **Read:**
  - PMOS MR: Pre-discharge RBL, raise RWL
  - NMOS MR: Precharge RBL Lower RWL



# Gain Cell eDRAM Requires Periodic Refresh

- Dynamic storage mechanism: data deteriorates over time
- Need for periodic refresh cycles (read/write)
  - Data arranged in sub-arrays
  - Parallel refresh in all sub-arrays




- Array availability

$$Availability\ [\%] = 1 - \frac{T_{\text{clk}}}{T_{\text{ret}}} N_{\text{r}}$$

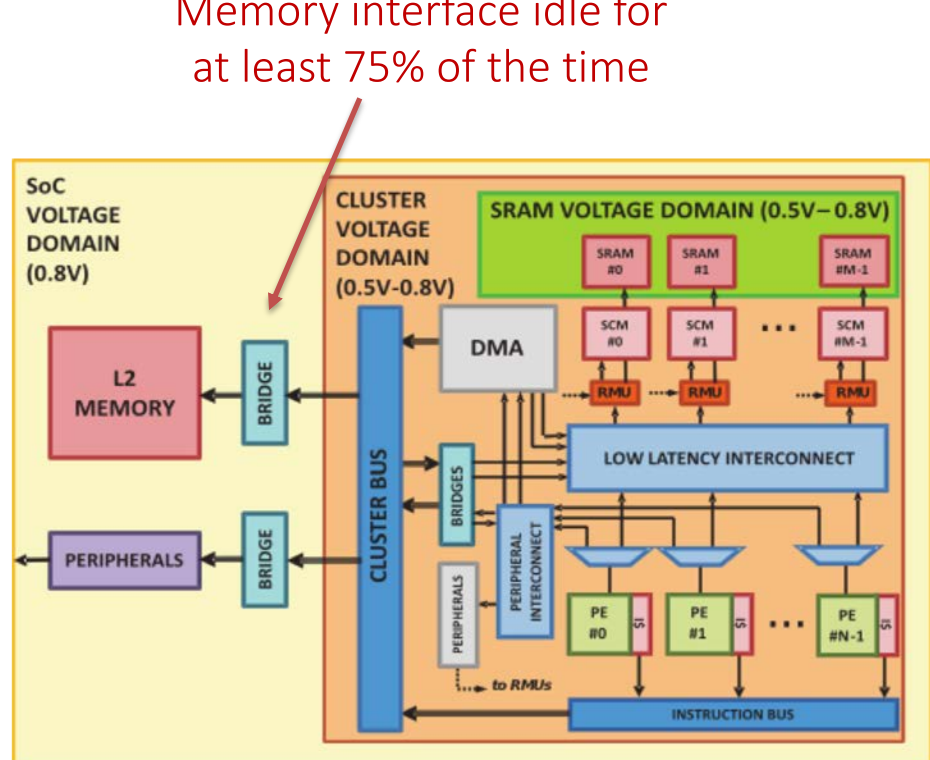
  - Typical retention times: $T_{\text{ret}} =$ 100us – 1ms
  - Typical access/refresh cycle-time: $T_{\text{clk}} =$ 10ns
  - Typical sub-array size $N_r$ =128-256 rows

Typical array availability: ~98%

# Gain Cell eDRAM Requires Periodic Refresh

- Typical application of eDRAM: IoT SoCs:
  - Multi-level memory hierarchy with a high-capacity L2 memory
  - Connected to processor through a shared bus (stalls are not unusual)
  - Processor supported by at least one level of cache
  - L2 memory supported by a dedicated memory controller

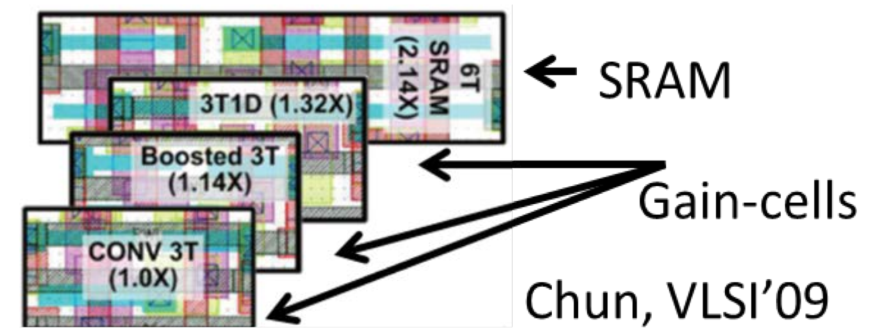
Memory interface idle for at least 75% of the time

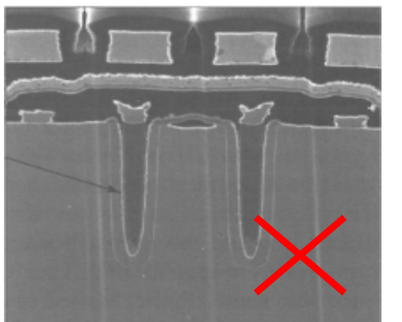



- **L2 memory** access in short bursts **with long idle periods** (>> 75%) that **allow to hide refresh with almost no overhead**

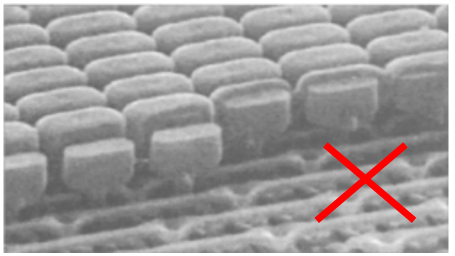
# Gain-Cell Embedded DRAM Advantages

- Gain Cells have several advantages over conventional 6T SRAM and over 1T-1C eDRAM
  - **Smaller** cell size than SRAM, **less bitcell leakage**
  - Compared to 1T-1C eDRAM:
    - **Logic-compatible**, i.e., no special processing steps, and no extra cost
    - **Non-destructive read** operation
  - Naturally **suport two-port operation**
  - Can be **optimized for read-ability AND write-ability**
  - Operational under **near/sub-threshold voltages**
  - Often **lower retention power** (leakage+refresh) than SRAM static power
  - **In many systems refresh can be hidden with almost no overhead**



**Trench cap**

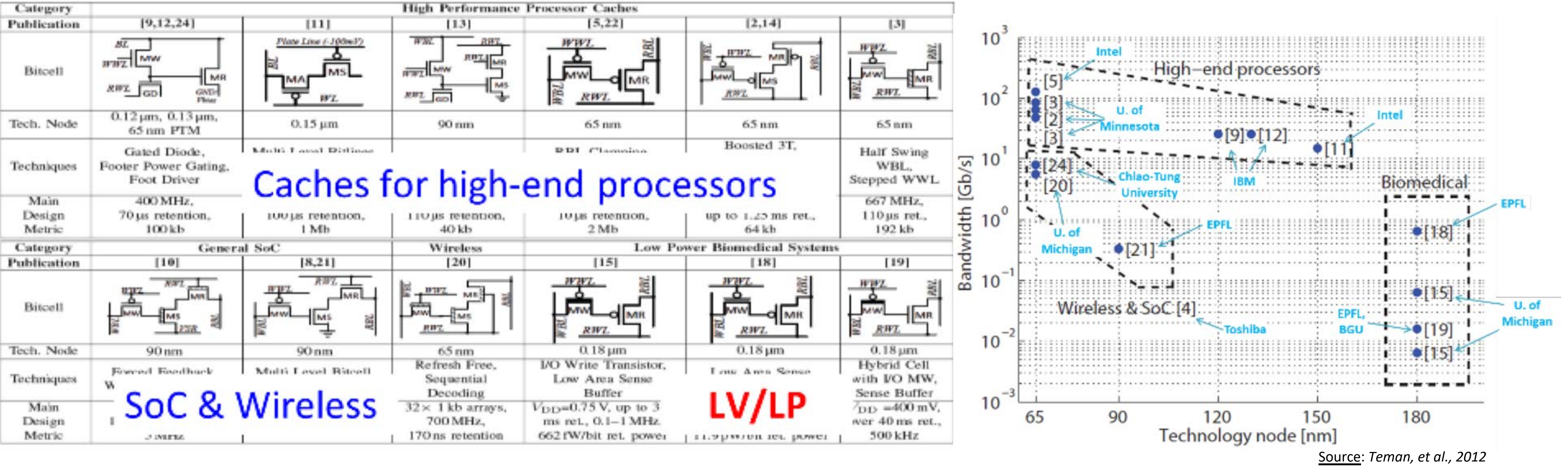


**Stacked caps**

Kang, McGraw-Hill

# GC eDRAM is an Attractive Alternative for Many Applications

- GC-eDRAM is a class of memory with many different flavors
- **Different bit-cells offer a wide range of design trade-offs** between area, retention time, access delay, and power consumption



| Category | High Performance Processor Caches | | | | | |
|---|---|---|---|---|---|---|
| Publication | [9,12,24] | [11] | [13] | [5,22] | [2,14] | [3] |
| Bitcell | | | | | | |
| Tech. Node | 0.12 µm, 0.13 µm, 65 nm PTM | 0.15 µm | 90 nm | 65 nm | 65 nm | 65 nm |
| Techniques | Gated Diode, Footer Power Gating, Foot Driver | [illegible] | [illegible] | [illegible] | Boosted 3T, [illegible] | Half Swing WBL, Stepped WWL |
| Main Design Metric | 400 MHz, 70 µs retention, 100 kb | 100 µs retention, 1 Mb | 110 µs retention, 40 kb | 10 µs retention, 2 Mb | up to 1.25 ms ret., 64 kb | 667 MHz, 110 µs ret., 192 kb |

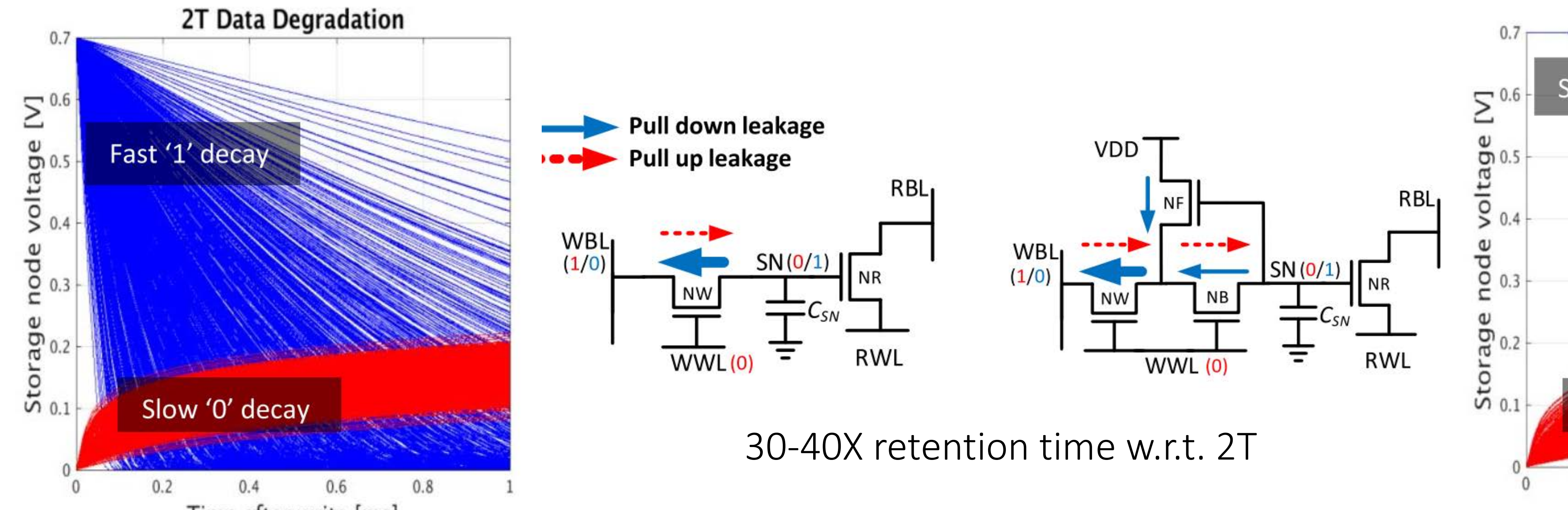
| Category | General SoC | | Wireless | Low Power Biomedical Systems | | |
|---|---|---|---|---|---|---|
| Publication | [10] | [8,21] | [20] | [15] | [18] | [19] |
| Bitcell | | | | | | |
| Tech. Node | 90 nm | 90 nm | 65 nm | 0.18 µm | 0.18 µm | 0.18 µm |
| Techniques | [illegible] | [illegible] | Refresh Free, Sequential Decoding | I/O Write Transistor, Low Area Sense Buffer | [illegible] | Hybrid Cell with I/O MW, Sense Buffer |
| Main Design Metric | [illegible] | [illegible] | 32× 1 kb arrays, 700 MHz, 170 ns retention | $V_{DD}$=0.75 V, up to 3 ms ret., 0.1–1 MHz, 662 fW/bit ret. power | [illegible] | $V_{DD}$=400 mV, [illegible] 40 ms ret., 500 kHz |

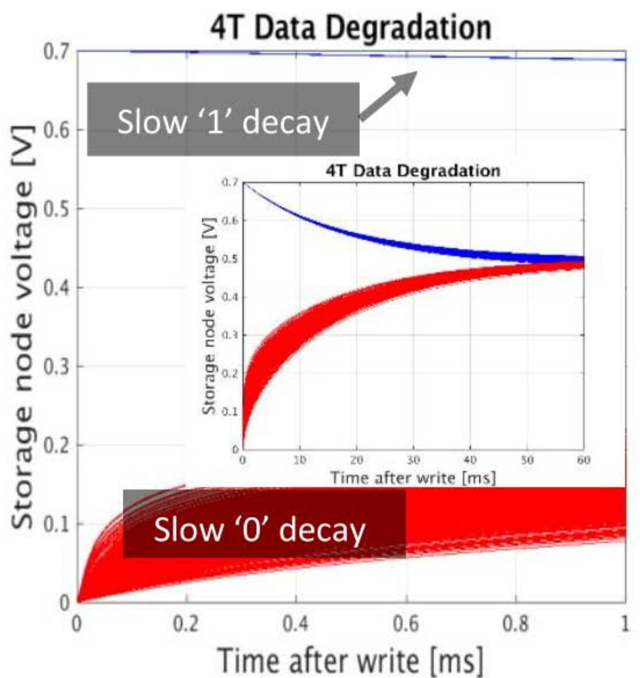


Source: *Teman, et al., 2012*

# GC-eDRAM is also feasible for nm-CMOS Nodes

- Advanced process nodes suffer from high leakage currents
- eDRAM has been **considered infeasible below 65nm**
- 4T bit-cell topology provides feedback for weak data level and **enables eDRAM** with milliseconds retention time **in 28nm CMOS**



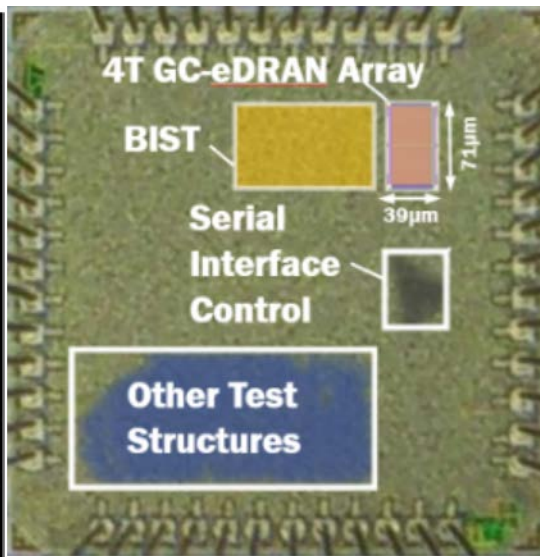



# High Density Low Power GC-eDRAM in 28nm FD-SOI

- Clear area reduction compared to SRAM with sufficient retention

| | 6T SRAM | 2T Gain-Cell | 3T Gain-Cell | Our 4T Gain-Cell |
|---|---|---|---|---|
| Cell Structure | VDD WL WL BL BLB | WWL RBL WBL RWL | RBL WWL WBL RWL | WWL RBL WBL VDD RWL |
| Technology Node | 28nm FD-SOI | 28nm FD-SOI | 28nm FD-SOI | 28nm FD-SOI |
| Cell Size ($\mu m^2$) (Non-pushed logic design rules) | 0.325$\mu m^2$ [1X] | 0.152$\mu m^2$ [0.47X] | 0.186$\mu m^2$ [0.57X] | 0.23$\mu m^2$ [0.71X] |
| 8Kbit Macro Size ($\mu m^2$) | 3321$\mu m^2$ [1X] Pushed SRAM design rules | N/A | N/A | 2769$\mu m^2$ [0.83X] Non-pushed logic design rules |
| Supply Voltage | 700mV | 700mV | 700mV | 700mV |
| Data Retention Time | Static | 32$\mu s$ @ 27C * 3.1$\mu s$ @ 85C * | 51$\mu s$ @ 27C * 4.14$\mu$ @ 85C * | 1691.4$\mu s$ @ 27C ** 154.95$\mu s$ @ 85C ** |
| Array Retention Power* | 74.3nW/8Kb @ 27C 1.36$\mu$W/8Kb @ 85C | 2.1$\mu$W/8Kb@ 27C 22$\mu$W/8Kb @ 85C | 1.4$\mu$W/8Kb@ 27C 16.8$\mu$W/8Kb @ 85C | 57.37nW/8Kb@ 27C 909.34nW/8Kb@ 85C |

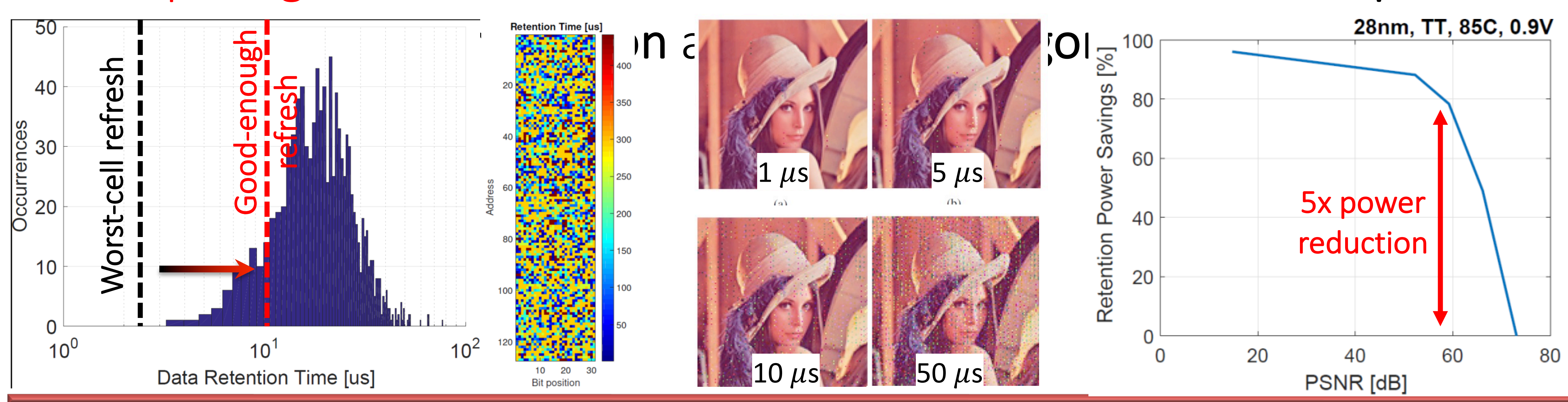
*Simulated **Measured




Source: R. Giterman , A. Fish, A. Burg and A. Teman, IEEE Transactions on Circuits and Systems I (TCAS-I), August 2017

# Approximate Computing with Unreliable (eDRAM) Memory

- **Manufacturing inaccuracies** and **different operating conditions lead to variations in the circuit behavior** within each chip and between chips
  - Conventional solution: refresh with **guardbands for** 100% **reliable operation**
- Computing with unreliable memories: **relax refresh and accept errors**



# Conclusions

- Memory is the limiting factor in area and power for almost all integrated circuits
- 6T SRAM is well established, but consumes large are and power and limits voltage scaling
- **GC-eDRAM is a real alternative for SRAM**
  - Area and energy efficient
  - Refresh can be hidden in many applications
  - Is feasible in advanced technology nodes (28nm and below)
  - GC eDRAM provides further potential when combined with new fault tolerant computing paradigms such as approximate computing or AI