Gain-Cell Embedded DRAMs: Modeling and Design Space

Andrea Bonetti¹, Roman Golman², Robert Giterman¹,²,³, Adam Teman²,³, Andreas Burg¹,³

¹ Telecommunications Circuits Laboratory, EPFL
² EnICS Labs Impact Center, Faculty of Engineering, Bar-Ilan University
³ RAAAM Technologies

IEEE TVLSI 28(3), March 2020
Data-Intensive Applications Call for Large-Size Memories

- **Silicon area** of SoCs often **dominated by memories** (especially 6T SRAM)
- **Memory size** is **forecasted to increase** further: e.g., AI and ML systems [1]
- Not only an area problem: RAMs are often the **bottleneck** of VDD scaling

---

https://irds.ieee.org/
### Features
- Six-transistor bitcell
- Large noise margins
- Symmetric layout
- Fast readout

### Drawbacks
- Large size: ≥ 6 transistors per bit
- High leakage: VDD-to-GND paths
- Poor VDD scaling: due to ratioed op.
- Single-Ported: WR or RD per cycle

6T SRAM is by far the most common on-chip memory in VLSI SoCs
RAAAM’s GC-eDRAM Technology Compared to SRAM

Commodity 6T-SRAM

- Schematic: 6 Transistors / bit
- Layout: 1x Area

RAAAM’s GC-eDRAM

- Schematic: 2-3 Transistors / bit
- Layout: 0.45x-0.55x Area

Gain-Cell technology requires fewer transistors per bit resulting in a 50% smaller layout.
# Dynamic Storage For High Density

<table>
<thead>
<tr>
<th>Storage Mechanism</th>
<th>Static CMOS</th>
<th>Size</th>
<th>Power</th>
<th>Speed</th>
<th>No Refresh</th>
<th>Scaling</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRAM</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>eDRAM</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>RAAAM’s GC-eDRAM</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
</tr>
</tbody>
</table>

GC-eDRAM provides the density advantages of eDRAM in any standard CMOS process.
GC-eDRAM: Large Variety of Design Tradeoffs

Circuit Level

- Read and write peripherals: **power** vs. **access time**
- Different bit-cells: **area** vs. **retention time**

Memory Organization

- Geometry of basic array: **rows/columns**
- Breakdown into **sub-arrays** for larger arrays
Need for a GC-eDRAM Modeling Tool

**Design space of GC-eDRAM is large and complex**

- Data Retention Time
- Refresh Rate
- Bitcell Topology
- Access Time
- Memory Organization
- Silicon Area
- Memory Density
- Memory Bandwidth

**Features**
- **Scale existing memories**
  - Memory Design
    - Predict memory performance
    - Support early design choices
- **Predict future memories**
  - System Architecture
    - Complex SoCs heavily depends on memory performance
    - Support high-level memory integration

**Applications**
- Memory Design
- System Architecture
Prior Art: Modeling Tools for DRAM

**CACTI-D** by HP Labs
- Based on CACTI
- Complete memory hierarchy: from low-level SRAM to higher-level DRAMs
  
  [ Thoziyoor, ISCA 2008 ]

**T. Vogelsang** (Rambus)
- Detailed energy model for DRAMs
- Physical floorplan and wires modeling
  
  [ Volgelsang, MICRO 2010 ]

**DDArt**
- Flexible: based on components definition
- Improved RC delay model

  [ Shih, TCAD 2014 ]

**DRAMSpec**
- Open-source tool
- Bitcell modeled as RC unit
- Integrated in gem5 simulator

  [ Naji, SAMOS 2015 ]

None of them can be used to model GC-eDRAMs
- Not all publicly available
- No GC-eDRAM model
- Refresh rate vs access time
- Large variety of GCs
- Non-destructive read
- Dual-port operation
GEMTOO: Modeling Features

Exploration of various design parameters
- Bit-cell array and peripherals
- Different type of Gain Cells
- Different types of memory organization
- Various geometry and structural transformations

Reporting of key characteristics
- Macro area and array efficiency
- Accurate read and write-access delays
- Interactions between refresh rate and delay
Validation of the GEMTOO model

Timing estimation validated against transistor-level simulations and silicon measurements in 28nm

### Post-layout simulations

- Max 15% error on estimated $F_{\text{MAX}}$
- Different memory sizes and organizations where loads of the interconnects scale accordingly

<table>
<thead>
<tr>
<th>Type</th>
<th>Folding</th>
<th>Size [Kb]</th>
<th>$f_{\text{sim}}^{\text{MAX}}$ [MHz]</th>
<th>$f_{\text{mod}}^{\text{MAX}}$ [MHz]</th>
<th>Error [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monolithic</td>
<td></td>
<td>8</td>
<td>1240</td>
<td>1420</td>
<td>15</td>
</tr>
<tr>
<td>Monolithic</td>
<td></td>
<td>32</td>
<td>780</td>
<td>830</td>
<td>6</td>
</tr>
<tr>
<td>Hierarchical</td>
<td>x1</td>
<td>64</td>
<td>710</td>
<td>760</td>
<td>8</td>
</tr>
<tr>
<td>Hierarchical</td>
<td>x2</td>
<td>64</td>
<td>950</td>
<td>920</td>
<td>4</td>
</tr>
</tbody>
</table>

### Silicon measurements

- 4 Kb GC-eDRAM in 28nm bulk [1]
- Random process variations: $F_{\text{MAX}}$ limited by read delay of GC with smallest RBL driving capability
- ON-resistance of GC read port derived with 10k Monte Carlo runs
- $F_{\text{MAX}}$ error between +7% and -9% for $n_{\sigma}$ of 4 and 6, respectively

Exploration: Optimization for Operating Frequency

$F_{\text{MAX}}$ optimization → critical-path delay → typically read-access delay

RBL delay up to 80% of read delay due to the weak GC driving strength

a) Avg. $2 \times F_{\text{MAX}}$ moving from 2T to 3T
b) Avg. +29% on $F_{\text{MAX}}$ reducing $V_{\text{SN}}$ degradation limit from 30% to 20%

Cutting BLs reduces driving load for GCs: +47% on $F_{\text{MAX}}$ when folding
Exploration: Optimization for Availability and Bandwidth

**Folding the Memory**

- **↑ Folding Factor**
  - GC Arrays
  - $F_{MAX}$
  - Avail. & BW

**Relaxing the Refresh Rate**

- **↑ Availability**
- **↓ Refresh Period**
  - Refresh rate
  - $F_{MAX}$
  - Availability
- **↑ BW monolithic**
- **↓ BW folded**

**GC Topology and Refresh Rate**

*High DRT of 4T ensures > 99% availability and highest BW at highest refresh rate*

<table>
<thead>
<tr>
<th>GC</th>
<th>$t_r$ [µs]</th>
<th>$v_d$ [%]</th>
<th>$f_{max}$ [MHz]</th>
<th>$\alpha$ [%]</th>
<th>$b$ [GB/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>2T</td>
<td>0.48</td>
<td>20</td>
<td>480</td>
<td>72.8</td>
<td>1.4</td>
</tr>
<tr>
<td>2T</td>
<td>3.98</td>
<td>30</td>
<td>390</td>
<td>95.9</td>
<td>1.5</td>
</tr>
<tr>
<td>2T</td>
<td>19.24</td>
<td>40</td>
<td>280</td>
<td>98.8</td>
<td>1.1</td>
</tr>
<tr>
<td>4T</td>
<td>74.6</td>
<td>20</td>
<td>420</td>
<td>99.7</td>
<td>1.6</td>
</tr>
<tr>
<td>4T</td>
<td>156.7</td>
<td>30</td>
<td>340</td>
<td>99.8</td>
<td>1.3</td>
</tr>
<tr>
<td>4T</td>
<td>397.9</td>
<td>40</td>
<td>240</td>
<td>99.9</td>
<td>0.9</td>
</tr>
</tbody>
</table>
**Exploration: Optimization for Memory Density**

- Area efficiency increases with memory size
- > 50% area occupied by GCs for > 4 Kb
- **GC area optimization** is key for high density

**Up to 33% area reduction**
when 2T GC is chosen instead of 4T GC (but less memory bandwidth)

<table>
<thead>
<tr>
<th>Size [Kb]</th>
<th>Memory Area [μm²]</th>
<th>Area Reduction [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>552</td>
<td>474</td>
</tr>
<tr>
<td>4</td>
<td>1523</td>
<td>1267</td>
</tr>
<tr>
<td>16</td>
<td>4910</td>
<td>3994</td>
</tr>
<tr>
<td>64</td>
<td>17464</td>
<td>14019</td>
</tr>
</tbody>
</table>
Exploration: Retention power of GC-eDRAM

GCs have up to **60% lower leakage power** than 6T SRAM bitcell. However, **refresh power overhead** is present in GC-eDRAMs.

Preliminary estimation of **retentive power**:
- Bitcell leakage power
- Refresh power to drive WLs/BLs load

<table>
<thead>
<tr>
<th>Folding Factor</th>
<th>6T SRAM</th>
<th>8T SRAM</th>
<th>2T GC</th>
<th>3T GC</th>
<th>4T GC</th>
</tr>
</thead>
<tbody>
<tr>
<td>pA / bit</td>
<td>411</td>
<td>616</td>
<td>199</td>
<td>224</td>
<td>161</td>
</tr>
<tr>
<td>Deviation from 6T</td>
<td>-</td>
<td>+ 49%</td>
<td>-51%</td>
<td>-45%</td>
<td>-60%</td>
</tr>
</tbody>
</table>

- **Retentive Power vs Folding Factor**
  - Refresh overhead is 70% of total power in “tall” GC-eDRAMs (no folding)
  - Folding reduces power overhead to charge/discharge the shared BLs
  - Twice-folded GC-eDRAM consumes **34% less power** than 6T SRAM

18% reduction on refresh rate gives **8% additional power savings** for 99% yield in x1 folded GC-eDRAM.
Summary and Conclusions

• GC-eDRAM is a high-density, low-leakage, and logic-compatible alternative to conventional SRAM for large on-chip storage

• GEMTOO is the first modeling tool for GC-eDRAMs proposed to facilitate and speed up their design and integration in VLSI systems

• Fast scaling and optimization of GC-eDRAMs is enabled despite their complex design space determined by many interdependent variables

• Memory architecture- and device-level modeling enables an accurate estimation of timing, bandwidth, and area of GC-eDRAMs

• Multiple case studies of design-space exploration are presented to find design choices that fulfill the most critical memory requirements
Thank you

GEMTOO available for download at:

https://raaam-tech.com/
GEMTOO: GC-eDRAM Modeling Tool

Modular Code
- Analytical formulas
- Look-up tables

Inputs
- Technology
- Circuits
- Architecture

Outputs
- Timing
- Availability and Bandwidth
- Area

GEMTOO
GC-eDRAM Modeling Tool

Delays Estimation
- Interconnect Drivers
- Row Decoder
- RBL Mux
- Access Delays

Retention and Refresh
- Data Retention Time
- Refresh Rate

Physical Floorplan
- Gain-Cell Arrays
- Local Logic
- Global Logic
GEMTOO: Timing Modeling of the Read Path

\[ F_{\text{max}} = \frac{1}{\max(t_{wr}, t_{rd})} \]
\[ t_{wr} = \max(t_{wwl}, t_{wbl}) + t_{wsl} \]
\[ t_{rd} = t_{rwl} + t_{rbl} \]

**RWL Delay (t_{rw})**:
- Registers
- Global Address Decoder
  - Array RWL-enable
  - FI2 gates and interconnect layer
- Global RWL Drivers
  - DDM with routing parasitics
  - Based on floorplan
- Local RWL-Enable Logic
- Local RWL Driver

**RBL Delay (t_{rb})**:
- RBL discharge by the Gain Cell
  - DDM + parasitics
  - Degradation on SN voltage limits driving strength
- Voltage-Based Sense Amplifier
- Global RBL Mux
  - Based on floorplan
**GEMTOO: Memory Availability, Bandwidth, and Area**

- GC-eDRAM needs a **periodic refresh** to retain the stored data
- Memory content cannot be accessed during refresh (**stall**)

**Memory Availability (\(\alpha\))**

\[
\alpha = 1 - \frac{t_s}{t_r}
\]

- \(t_s\) : time to refresh memory content
- \(t_r\) : time between two refresh periods
- \(r_b\) : number of words per GC array

**Memory Bandwidth (\(b\))**

\[
b = \alpha f_{\text{max}} c_b
\]

- \(c_b\) : number of columns per GC array (word size)

**Area and Utilization**

- Memory density is **key** in embedded memories
- **Estimation** of physical dimensions based on:
  - Input dimensions of **unit blocks** (GC, cells…)
  - **Pitch-matched** to WLs/BLs of GC array
  - Floorplan and global/local logic based on **architectural transformations**

**Memory Density:** 3b/\(\mu\)m\(^2\)

**Area Efficiency:** 82%