**Problem**

- DRAM requires periodic refresh to avoid data loss
- Refreshes interfere with memory accesses and waste energy
- Refresh overhead limits DRAM scaling

**Our Mechanism: RAIDR**

1. Profile retention times (how frequently each row needs to be refreshed to avoid losing data)

<table>
<thead>
<tr>
<th>Row 1</th>
<th>Row 2</th>
<th>Row 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initially</td>
<td>Row 1</td>
<td>Row 2</td>
</tr>
<tr>
<td>11111111...</td>
<td>11111111...</td>
<td>11111111...</td>
</tr>
<tr>
<td>After 64 ms</td>
<td>11011111...</td>
<td>11111111...</td>
</tr>
<tr>
<td>After 128 ms</td>
<td>(64–128ms)</td>
<td>11111011...</td>
</tr>
<tr>
<td>After 256 ms</td>
<td>11111111...</td>
<td>11111111...</td>
</tr>
</tbody>
</table>

2. Group rows into different bins based on their retention time using Bloom filters for scalability and efficiency

   - Memory controller chooses each row as a refresh candidate every 64 ms
     - Row in 64-128 ms bin? (First Bloom filter: 256 B)
     - Refresh the row
     - Every other 64 ms window, refresh the row
     - Every 4th 64 ms window, refresh the row

   - Row in 128-256 ms bin? (Second Bloom filter: 1 KB)
   - Every other 64 ms window, refresh the row

3. Refresh rows in different bins at different rates

**Key Observation and Idea**

- Most DRAM cells can be refreshed infrequently without losing data
- All cells are refreshed at the same worst-case rate
- High refresh rate imposed by few weak cells
- Key idea: Refresh rows containing weak cells more frequently; refresh other rows less frequently

**Results**

- 1.5 KB storage overhead in a 32 GB DRAM system
- 74.6% refresh reduction resulting in 8.6% performance gain and 16.1% energy efficiency improvement on average
- Benefits increase as DRAM scales in density