Lecture 5: Refresh, Chipkill

- Topics: refresh basics and innovations, error correction
Refresh Basics

• A cell is expected to have a retention time of 64ms; every cell must be refreshed within a 64ms window

• The refresh task is broken into 8K refresh operations; a refresh operation is issued every $t_{REFI} = 7.8$ us

• If you assume that a row of cells on a chip is 8Kb and there are 8 banks, then every refresh operation in a 4Gb chip must handle 8 rows in each bank

• Each refresh operation takes time $t_{RFC} = 300$ns

• Larger chips have more cells and $t_{RFC}$ will grow
More Refresh Details

- To refresh a row, it needs to be activated and precharged.

- Refresh pipeline: the first bank draws the max available current to refresh a row in many subarrays in parallel; each bank is handled sequentially; the process ends with a recovery period to restore charge pumps.

- “Row” on the previous slide refers to the size of the available row buffer when you do an Activate; an Activate only deals with some of the subarrays in a bank; refresh performs an activate in all subarrays in a bank, so it can do multiple rows in a bank in parallel.
Fine Granularity Refresh

- Will be used in DDR4

- Breaks refresh into small tasks; helps reduce read queuing delays (see example)

- In a future 32Gb chip, tRFC = 640ns, tRFC_2x = 480ns, tRFC_4x = 350ns – note the high overhead from the recovery period
What Makes Refresh Worse

• Refresh operations are issued per rank; LPDDR does allow per bank refresh

• Can refresh all ranks simultaneously – this reduces memory unavailable time, but increases memory peak power

• Can refresh ranks in staggered manner – increases memory unavailable time, but reduces memory peak power

• High temperatures will increase the leakage rate and require faster refresh rates (> 85 degrees C  3.9us tREFI)
Refresh Innovations

- Smart refresh (Ghosh et al.): do not refresh rows that have been accessed recently
- Elastic refresh (Stuecheli et al.): perform refresh during periods of inactivity
- Flikker (Liu et al.): lower refresh rate for non-critical pages
- Refresh pausing (Nair et al.): interrupt the refresh process when demand requests arrive
- Preemptive command drain (Mukundan et al.): prioritize requests to ranks that will soon be refreshed
Refresh Innovations II

• Concurrent refresh and accesses (Chang et al. and Zhang et al.): makes refresh less efficient, but helps reduce queuing delays for pending reads

• RAIDR (Liu et al.): profile at run-time to identify weak cells; track such weak-cell rows in retention time bins with Bloom Filters; refreshes are skipped based on membership in a retention time bin

• Empirical study (Liu et al.): use a custom memory controller on an FPGA to measure retention time in many DIMMs; shows that retention time varies with time and is a function of data in neighboring cells
Basic Reliability

• Every 64-bit data transfer is accompanied by an 8-bit (Hamming) code – typically stored in a x8 DRAM chip

• Guaranteed to detect any 2 errors and recover from any single-bit error (SEC-DED)

• Such DIMMs are commodities and are sufficient for most applications

• 12.5% overhead in storage and energy

• For a BCH code, to correct $t$ errors in $k$-bit data, need an $r$-bit code, $r = t \times \text{ceil}(\log_2 k) + 1$
Terminology

• Hard errors: caused by permanent device-level faults

• Soft errors: caused by particle strikes, noise, etc.

• SDC: silent data corruption (error was never detected)

• DUE: detected uncorrectable error

• DUE in memory caused by a hard error will typically lead to DIMM replacement

• Scrubbing: a background scan of memory (1GB every 45 mins) to detect and correct 1-bit errors
Field Studies

• Memory errors are the top causes for hw failures in servers and DIMMs are the top component replacements in servers

• Study examined Google servers in 2006-2008, using DDR1, DDR2, and FBDIMM
### Table 1: Memory errors per year:

<table>
<thead>
<tr>
<th>Platf.</th>
<th>Tech.</th>
<th>Per machine</th>
<th>CE Incid. (%)</th>
<th>CE Rate Mean</th>
<th>CE Rate C.V.</th>
<th>CE Median Affct.</th>
<th>UE Incid. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>DDR1</td>
<td></td>
<td>45.4</td>
<td>19,509</td>
<td>3.5</td>
<td>611</td>
<td>0.17</td>
</tr>
<tr>
<td>B</td>
<td>DDR1</td>
<td></td>
<td>46.2</td>
<td>23,243</td>
<td>3.4</td>
<td>366</td>
<td>–</td>
</tr>
<tr>
<td>C</td>
<td>DDR1</td>
<td></td>
<td>22.3</td>
<td>27,500</td>
<td>17.7</td>
<td>100</td>
<td>2.15</td>
</tr>
<tr>
<td>D</td>
<td>DDR2</td>
<td></td>
<td>12.3</td>
<td>20,501</td>
<td>19.0</td>
<td>63</td>
<td>1.21</td>
</tr>
<tr>
<td>E</td>
<td>FBD</td>
<td></td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>0.27</td>
</tr>
<tr>
<td>F</td>
<td>DDR2</td>
<td></td>
<td>26.9</td>
<td>48,621</td>
<td>16.1</td>
<td>25</td>
<td>4.15</td>
</tr>
<tr>
<td>Overall</td>
<td>–</td>
<td></td>
<td>32.2</td>
<td>22,696</td>
<td>14.0</td>
<td>277</td>
<td>1.29</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Platf.</th>
<th>Tech.</th>
<th>Per DIMM</th>
<th>CE Incid. (%)</th>
<th>CE Rate Mean</th>
<th>CE Rate C.V.</th>
<th>CE Median Affct.</th>
<th>UE Incid. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>DDR1</td>
<td></td>
<td>21.2</td>
<td>4530</td>
<td>6.7</td>
<td>167</td>
<td>0.05</td>
</tr>
<tr>
<td>B</td>
<td>DDR1</td>
<td></td>
<td>19.6</td>
<td>4086</td>
<td>7.4</td>
<td>76</td>
<td>–</td>
</tr>
<tr>
<td>C</td>
<td>DDR1</td>
<td></td>
<td>3.7</td>
<td>3351</td>
<td>46.5</td>
<td>59</td>
<td>0.28</td>
</tr>
<tr>
<td>D</td>
<td>DDR2</td>
<td></td>
<td>2.8</td>
<td>3918</td>
<td>42.4</td>
<td>45</td>
<td>0.25</td>
</tr>
<tr>
<td>E</td>
<td>FBD</td>
<td></td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>0.08</td>
</tr>
<tr>
<td>F</td>
<td>DDR2</td>
<td></td>
<td>2.9</td>
<td>3408</td>
<td>51.9</td>
<td>15</td>
<td>0.39</td>
</tr>
<tr>
<td>Overall</td>
<td>–</td>
<td></td>
<td>8.2</td>
<td>3751</td>
<td>36.3</td>
<td>64</td>
<td>0.22</td>
</tr>
</tbody>
</table>
Field Studies

Schroeder et al., SIGMETRICS 2009

• A machine with past errors is more likely to have future errors
• 20% of DIMMs account for 94% of errors
• DIMMs in platforms C and D see higher UE rates because they do not have chipkill
• 65-80% of uncorrectable errors are preceded by a correctable error in the same month – but predicting a UE is very difficult
• Chip/DIMM capacity does not strongly influence error rates
• Higher temperature by itself does not cause more errors, but higher system utilization does
• Error rates do increase with age; the increase is steep in the 10-18 month range and then flattens out
Chipkill

• Chipkill correct systems can withstand failure of an entire DRAM chip

• For chipkill correctness
  ➢ the 72-bit word must be spread across 72 DRAM chips
  ➢ or, a 13-bit word (8-bit data and 5-bit ECC) must be spread across 13 DRAM chips
RAID-like DRAM Designs

• DRAM chips do not have built-in error detection

• Can employ a 9-chip rank with ECC to detect and recover from a single error; in case of a multi-bit error, rely on a second tier of error correction

• Can do parity across DIMMs (needs an extra DIMM); use ECC within a DIMM to recover from 1-bit errors; use parity across DIMMs to recover from multi-bit errors in 1 DIMM

• Reads are cheap (must only access 1 DIMM); writes are expensive (must read and write 2 DIMMs)

Used in some HP servers
RAID-like DRAM

- Add a checksum to every row in DRAM; verified at the memory controller
- Adds area overhead, but provides self-contained error detection
- When a chip fails, can re-construct data by examining another parity DRAM chip
- Can control overheads by having checksum for a large row or one parity chip for many data chips
- Writes are again problematic
SSC-DSD

* The cache line is organized into multi-bit symbols

* Two symbols are required for error detection and 3/4 symbols are used for error correction (can handle complete failure in one symbol, i.e., each symbol is fetched from a different DRAM chip)

* 3-symbol codes are not popular because it leads to non-standard DIMMs

* 4-symbol codes are more popular, but are used as 32+4 so that standard ECC DIMMs can be used (high activation energy and low rank-level parallelism) (16+4 would require a non-standard DIMM)
Virtualized ECC

• Also builds a two-tier error protection scheme, but does the second tier in software

• The second-tier codes are stored in the regular physical address space (not specialized DRAM chips); software has flexibility in terms of the types of codes to use and the types of pages that are protected

• Reads are cheap; writes are expensive as usual; but, the second-tier codes can now be cached; greatly helps reduce the number of DRAM writes

• Requires a 144-bit datapath (increases overfetch)
LoT-ECC

- Use checksums to detect errors and parity codes to fix
- Requires access of only 9 DRAM chips per read, but the storage overhead grows to 26%
Title

• Bullet