4 DRAM Refresh [60 points]

4.1 Basics [15 points]

A memory system is composed of eight banks, and each bank contains 2^{15} rows. Every DRAM row refresh is initiated by a command from the memory controller, and it refreshes a single row. Each refresh command keeps the command bus busy for 5 ns. We define *command bus utilization* as the fraction of total execution time during which the command bus is occupied.

1. [5 points] Given that the refresh interval is 64ms, calculate the command bus utilization of refresh commands. Show your work step-by-step.

   Command bus is utilized for $8 \times 2^{15} \times 5ns$ at every 64ms.
   $Utilization = (2^{18} \times 5ns)/(2^6 \times 10^6 ns) = 2^{12}/(2 \times 10^5) = 2^{11} \times 10^{-5} = 2.048\%$

2. [10 points] If 60% of all rows can withstand a refresh interval of 128 ms, how does the command bus utilization of refresh commands change? Calculate the reduction in bus utilization. Show your work step-by-step.

   At every 128 ms:

   - 60% of the rows are refreshed once.
     Command bus is busy for: $0.6 \times 8 \times 2^{15} \times 5ns = 3 \times 2^{18} ns$
   - 40% of the rows are refreshed twice.
     Command bus is busy for: $0.4 \times 8 \times 2^{15} \times 5ns \times 2 = 4 \times 2^{18} ns$

   $Utilization = (3 + 4) \times 2^{18} ns/128ms = 0.7 \times 2^{11} \times 10^{-5}$

   $Reduction = 1 - (0.7 \times 2^{11} \times 10^{-5})/(2^{11} \times 10^{-5}) = 30\%$

4.2 VRL: Variable Refresh Latency [45 points]

In this question, you are asked to evaluate "Variable Refresh Latency," proposed by Das, A et al. in DAC 2018[1]

The paper presents two key observations:

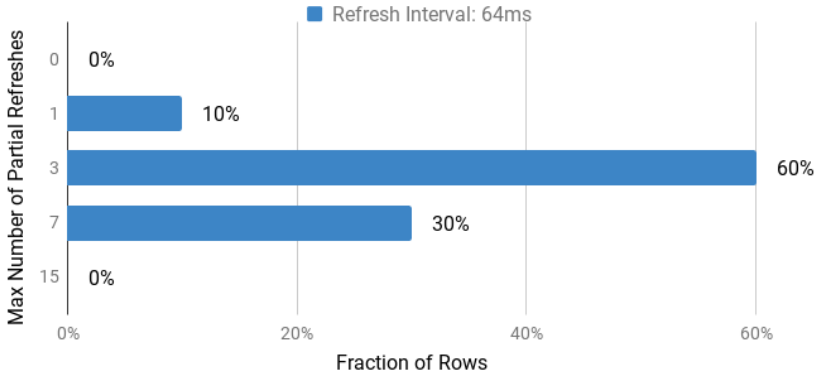
- First, a cell's charge reaches 95% of the maximum charge level in 60% of the nominal latency value during a refresh operation. In other words, the last 40% of the refresh latency is spent to increase the charge of a cell from 95% to 100%. Based on this observation, the paper defines two types of refresh operations: (1) *full refresh* and (2) *partial refresh*. Full refresh uses the nominal latency and restores the cell charge to 100%, while the latency of partial refresh is only 60% of the nominal value and it restores 95% of the charge.
- Second, a fully refreshed cell operates correctly even after multiple partial refreshes, but it needs to be fully refreshed again after a finite number of partial refreshes. The maximum number of partial refreshes before a full refresh is required varies from cell to cell.

[1] Das, A. et al., *"VRL-DRAM: Improving DRAM Performance via Variable Refresh Latency."* In Proceedings of the 55th Annual Design Automation Conference (DAC), 2018.

The **key idea** of the paper is to apply a *full refresh* operation **only when necessary** and use *partial refresh* operations at all other times.

(a) [15 points] Consider a case in which:

- Each row must be refreshed every 64 ms. In other words, the refresh interval is 64 ms.
- Row refresh commands are evenly distributed across the refresh interval. In other words, all rows are refreshed exactly once in any given 64 ms time window.
- You are given the following plot, which shows *the distribution of the maximum number of partial refreshes* across all rows of a particular bank. For example, if the maximum number of refreshes is three, those rows can be partially refreshed for at most three refresh intervals, and the fourth refresh operation must be a full refresh.
- If all rows were always fully refreshed, the time that a bank is busy, serving the refresh requests within a refresh interval would be T.



How much time does it take (in terms of T) for a bank to refresh all rows within a refresh interval, after applying Variable Refresh Latency?

Full refresh latency = T, partial refresh latency = 0.6T.

10% of the rows are fully refreshed at every other interval:
$0.1 \times (0.5 \times 0.6T + 0.5 \times T)$
60% of the rows are fully refreshed after every three partial refresh:
$0.6 \times (0.75 \times 0.6T + 0.25 \times T)$
30% of the rows are fully refreshed after every seven partial refresh:
$0.3 \times (0.875 \times 0.6T + 0.125 \times T)$

Then, new refresh latency of a bank would be 0.695T.

(b) [15 points] You find out that you can relax the refresh interval, and define your baseline as follows:

- 90% of the rows are refreshed at every 128ms; 10% of the rows are refreshed at every 64ms.
- Refresh commands are evenly distributed in time.
- All rows are always fully refreshed.
- A single refresh command costs $0.2/N$ $ms.$, where N is the number of rows in a bank.
- *Refresh overhead* is defined as the fraction of time that a bank is busy, serving the refresh requests over a very large period of time.

Calculate the refresh overhead for the baseline.

At every 128ms:

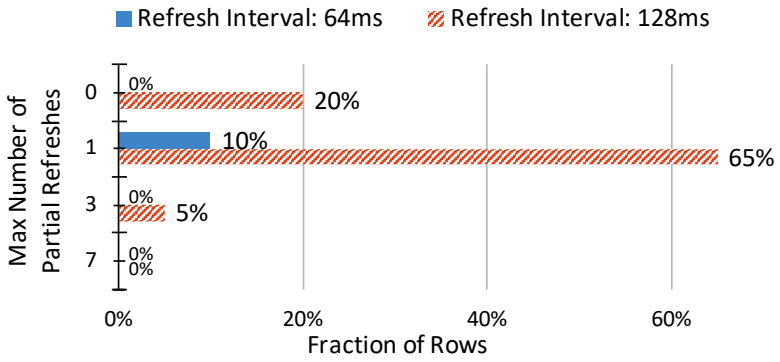
10% of the rows are refreshed twice, 90% of the rows are refreshed once.

Total time spent for refresh in a 128 ms. interval is $(0.9N + 2 \times 0.1N) \times 0.2/N = 0.22ms$.

Then refresh overhead is 0.22/128

(c) [15 points] Consider a case where:

- 90% of the rows are refreshed at every 128ms; 10% of the rows are refreshed at every 64ms.
- Refresh commands are evenly distributed in time.
- You are given the following plot, which shows *the distribution of the maximum number of partial refreshes* across all rows of a particular bank.
- A single refresh command costs $0.2/N$ *ms.*, where N is the number of rows in a bank.
- *Refresh overhead* is defined as the fraction of time that a bank is busy, serving the refresh requests over a very large period of time.



Calculate the refresh overhead. Show your work step-by-step. Then, compare it against the baseline configuration (the previous question). How much reduction do you see in the performance overhead of refreshes?

Full refresh of a row costs 0.2/N ms. Then, partial refresh of a row costs 0.12/N ms

At every 4×128 *ms*:

- 20% of the rows are refreshed for 4 times:
  4 times *fully refreshed* and 0 times *partially refreshed*.
- 10% of the rows are refreshed for 8 times:
  4 times *fully refreshed* and 4 times *partially refreshed*.
- 65% of the rows are refreshed for 4 times:
  2 times *fully refreshed* and 2 times *partially refreshed*.
- 5% of the rows are refreshed for 4 times:
  1 time *fully refreshed* and 3 times *partially refreshed*.

Total time spent for refresh is:
$= (0.2N \times 4 + 0.1N \times 4 + 0.65N \times 2 + 0.05N \times 1) \times 0.2/N$
$+ (0.2N \times 0 + 0.1N \times 4 + 0.65N \times 2 + 0.05N \times 3) \times 0.12/N$

$= (0.8 + 0.4 + 1.3 + 0.05) \times 0.2 + (0.4 + 1.3 + 0.15) \times 0.12$
$= 2.55 \times 0.2 + 1.85 \times 0.12$
$= 0.51 + 0.222 = 0.732$ ms.
Then, refresh overhead is: $0.732/(4 \times 128)$

So, the reduction is $1 - (0.732/4)/0.22 = 1.7/22 \approx 7.7\%$.