Notes from the meeting 02/24/2016

**Policies and Design Decisions:**

1. Out of Order, access scheduling
2. Starvtion = 50 cycles : Whatever is starving the most gets prioritized
3. Keep everything in CPU Cycle : 1 DRAM = 4 CPU Clock ticks
4. If write command is issued to the same bank/row don't prioritize.

**Program Flow:**

* Use enumeration for the Bank name, Commands and time constraints
* Use 2D array to keep track of the elapsed time :
  + Use counters that saturates at a maximum value
  + Touch each of the array every single time
  + Compare for the clock ticks, and either update the counter or issue next command
    - **if (Tick\_Since\_Last [Bank][Command]) >=time\_constraint)**

**send next command**

**else**

**++Tick\_Since\_Last[Bank][Command]**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **PRE** | **ACT** | **RD** | **RDAP** | **WR** | **WRAP** |
| **Bank0** |  |  |  |  |  |  |
| **Bank1** |  |  |  |  |  |  |
| **Bank2** |  |  |  |  |  |  |
| **Bank3** |  |  |  |  |  |  |
| **Bank4** |  |  |  |  |  |  |
| **Bank5** |  |  |  |  |  |  |
| **Bank6** |  |  |  |  |  |  |
| **Bank7** |  |  |  |  |  |  |

**For open page policy:**

Hit: RD (Same bank, same row)

Miss: PRE--> ACT--> RD (Same bank, different row)

**Masks:**

**Row bits: 15**

**Column bits: 11 bits**

**Bank: 3 bits**

**No\_Care : 3 bits**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bits | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  | Row bits | | | | | | | | | | | | | | | Banks | | | Column bits | | | | | | | | | | | Byte Se | | |

Row bit-mask : 0xFFFE0000 >> 17 bits to get row address

Bank bit-mask: 0x1C000 >> 14 bits to get bank address

Column bit-mask: 0x3FF8 >> 3 bits to column address

**Scheduling:**

**Case 1:**  Based on current timing, and DIMM status, check if the command (Read/Write/Fetch) is

ready to execute

1. Check for starvation
2. Check if there is a command that is trying to access open bank/row
   * If yes,
     + Check if Read-Write is issues
       - Yes- Don't prioritize
       - No-Prioritize
   * No—access next oldest command that is not starved

More on Out of order execution:

Prioritize in the following higher order (Left to right)

|  |  |  |
| --- | --- | --- |
| Read  Different Row | Read  Different Column | Read  Different column only |



