CS2610 Lab 6 Report

Objective: To implement and understand various dram row buffer management policies.

Setting up Environment

-> we need to download the usimm simulator (latest version v1.3) , which is publicly available from their website.
->This simulator has a user-friendly interface.
-> In the usimm/src directory there is a scheduler.c file, that is the file where we need to implement our buffer management policy.(Memory Scheduler)
->To run the simulator , we need to use the makefile already given in usimm/src directory.
->Then we can start the simulator by giving appropriate input files to the executable file.

Row-Buffer Management policies

1) Open-page policy:
   In this policy, we keep the row buffer open till the next request to the bank . If the request is a row buffer hit, then we directly take the data from the sense amplifiers. Else we do a precharge to the bank, and then extract the data from it.

2) Closed-page policy:
   In this policy, we do not expect the requests to be row buffer hits. Hence, we do a precharge on idle cycles, so that precharge time for memory requests can be saved.

3)Adaptive-page policy:
   This is an intermediate between open-page and closed-page.Here we maintain an Counter which is modified accordingly. When this counter value crosses a low mark, it implements an open-page policy . when this counter value crosses a upper mark, it implements a closed -page policy.

   While in open-page policy , counter is only incremented when an page miss occurs. While in closed-page policy, counter is decremented only when page hit takes places with the last closed page.

   Implementation of Adaptive page policy:

   Implement a 4 bit saturation counter with variable int.
   Initialise it with some value as required. But while modifying it, if it crosses 15 then make its value 15. Similarly , if it drops below 0 , make its value less than 0.

And shift the policy from open to close and close to open whenever the counter crosses the marks.

Adaptive policy

Now using a if-else construct use open-page request code if its in open page policy else use closed page request code.
Since Memory controller already checks for row buffer hits, we can directly use the scheduler-fcfs.c as open-page policy code.
And closed-page policy can be obtained from the scheduler-closed.c file.

Modifying the counter:
While in open-page policy mode:
If an act command is given, it implies that is a page miss hence increment the counter else do nothing.

While in closed-page policy mode:
Maintaining an array that stores the last closed page for each bank. Now whenever column_rd/column_wr comes we check if that row equals the row in the array corresponding to that bank. If equal decrement the counter else do nothing.

DRAM Performance

Trace file	Closed	Open	Adaptive
Black	218154100	228447648	228447648
Comm1	281429212	302401460	302401460
Comm2	375417625	402959473	402959473
Face	263273636	281165956	281165956
Ferret	278499505	296639309	

Fluid	302044069	316586849
Freq	191565825	200764257
MT0	485160485	518855181
MT1	473645197	504630009
MT2	474193573	505752473
MT3	473971133	505067601
Stream	215208456	226896108
Swapt	320615393	334882841

As we can see , closed page policy is the best policy among the three.

Comparing Memory Address Schemes

By setting Address_Mapping variable in params.h to 0 or 1 we can change it to either Address Mapping 0 or 1.

Trace file	Address Mapping 0	Address Mapping 1
Black	218154100	222379224

Comm1	281429212	270601076
Comm2	375417625	367892056
Face	263273636	333136296
Ferret	278499505	319620333
Fluid	302044069	312591381
Freq	191565825	201057952
MT0	485160485	499728733
MT1	473645197	476759533
MT2	474193573	477114445
MT3	473971133	476606725
Stream	215208456	227609028
Swapt	320615393	333655940