Behavioral model of Memory Controller scheduler

PROJECT REPORT

For our Behavioral model of MC scheduler , we have implemented a **True open page policy** (we keep track of open rows in all the banks).

Project Overview:

- We are using System Verilog for implementing the scheduler.
- In this code we are not using any priorities, like reads over writes.
- Bank level parallelism is not used.
- Open page policy is employed.

Code Implementation:

- Employed two functions to transition between the FSM states based on the delays generated in the Bank Status Register array.
- The BSR(Bank Status Register) array calculates the time since a previous DRAM request was issued and it keeps note of the bank_group and bank to which that request was issued.
- BSR array is divided into 4 columns. - [ACTIVATE][PRECHARGE][READ][WRITE].
- We call these two functions into our initial block which loops through the one while loop to move from one state to another.
- Once a new request is issued we are moving that corresponding bank group and bank's value to zero for that request and now that becomes the latest transaction of that type.
- The FSM was created to be implemented in such a way to represent the true open page policy where each row is kept open once an activate is issued to it, unless we give a precharge command, where it will be closed for a short time.

Verification

- Wrote a python script to generate trace files.
- Takes inputs as time ,number of entries in trace file,bank group and row
- According to the input format, we can generate entries to be page hit ,page miss and page empty.
- Can also check for jumps in timing by issuing requests at different times.