Microprocessor System Design Fall 2023

ECE 585

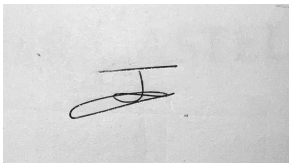
Project (Fall 2023)

Test-Plan

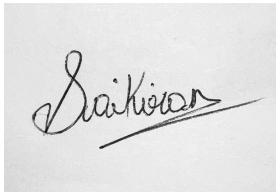
Behavioural Simulation of DDR5 Memory Controller Scheduler

TEAM-25:

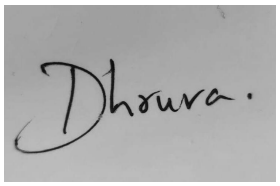
- **Jonathan Thomas**
  (PSU ID : 958484004)
  jonthom@pdx.edu



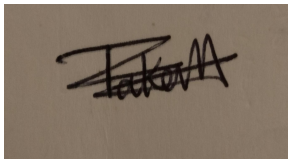
- **Saikiran Ranga -**
  (PSU ID: 925738827)
  sranga@pdx.edu



- **Dhruva Teja Koppisetty -**
  (PSU ID: 952138674)
  dhruva@pdx.edu



- **Rakesh Gurajala-**
  (PSU ID: 930251476)
  gurajala@pdx.edu



ABSTRACT:

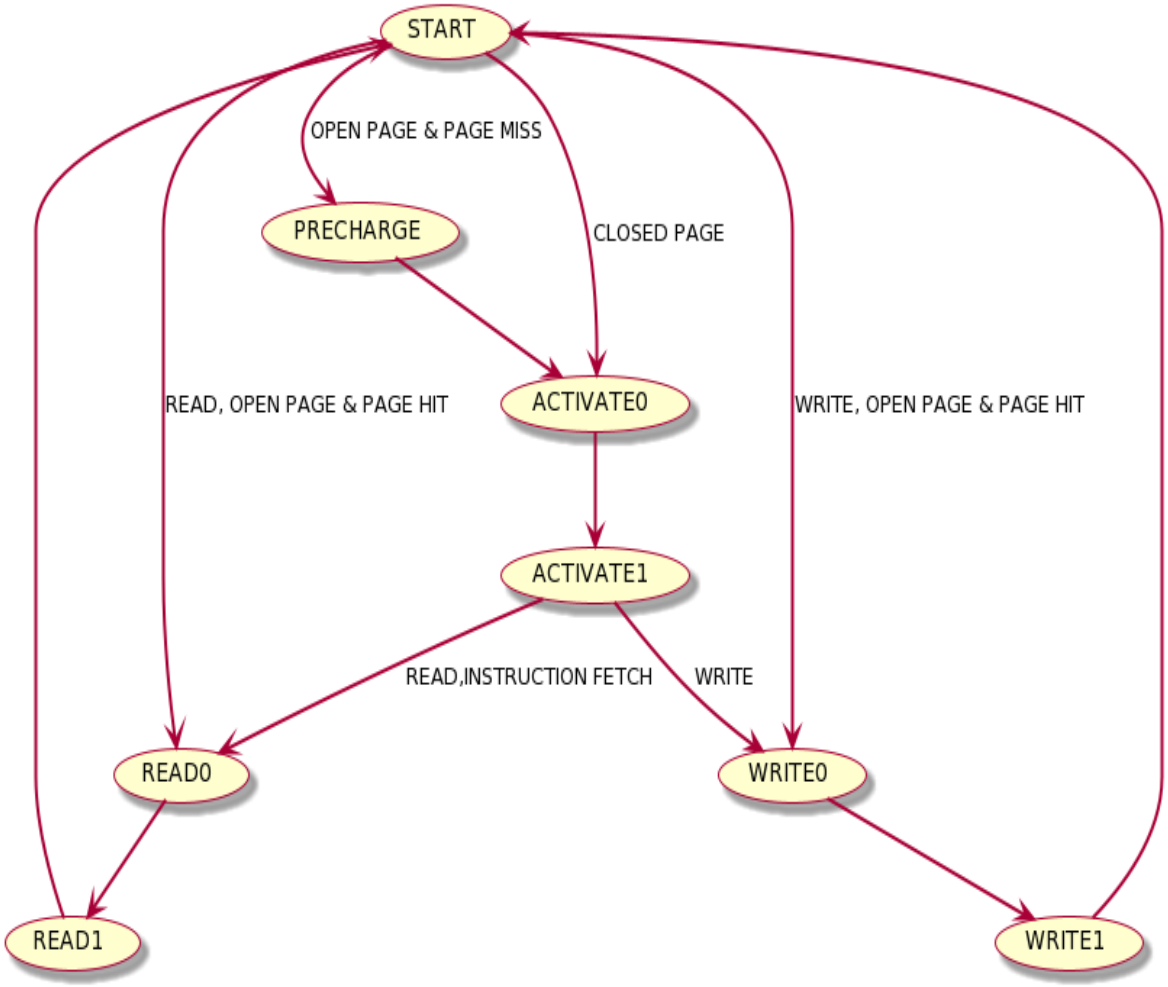
The Memory Controller Scheduler Project focuses on the development of a memory controller module for a 12-core 4.8 GHz processor employing a single 16GB PC5-38400 DIMM with DDR5 technology. The project involves implementing a scheduler capable of efficiently managing memory requests in compliance with DDR5 timing specifications. The simulation considers a relaxed consistency model with two independent channels in the DIMM, each organized with x8 devices having a 1 KB page size and 40-39-39-76 timing.

The simulation, not intended for synthesizable design, involves tracking DRAM cycle counts and adhering to various DDR5 timing parameters, including tRC, tRAS, tRRD_L, tRRD_S, tRP, tRFC, CWL, tCAS, tRCD, tWR, tRTP, tCCD_L, tCCD_S, tCCD_L_WR, tCCD_S_WR, tBURST, tCCD_L_RTW, tCCD_S_RTW, tCCD_L_WTR, and tCCD_S_WTR. Input traces in a specified format drive the simulation, and the output provides a trace of all issued DRAM commands.

The project offers flexibility in choosing programming languages or hardware description languages for implementation. A key aspect is the inclusion of a scheduling algorithm that can range from basic (Level 0) to more advanced (Level 3) with variations in bank level parallelism, open page policy, and out-of-order execution. In this project we are implementing Level 1 open page policy.

The evaluation criteria encompass adherence to external specifications, completeness, correctness, quality of design decisions, implementation, and the clarity of the project report. Additional points can be earned by simulating more aggressive scheduling policies and implementing optional features such as adaptive open page policies and out-of-order scheduling. The abstract provides an overview of the project's scope, objectives, and evaluation criteria.

STATE TRANSITION DIAGRAM :



Timing Parameters Spread Sheet:

Sno	Timing parameter	From	To	Time	Condition(same bank or not)	Comment / Notes
1	Trc	Activate	Activate	115	same bank	Trc = Tras+ Trp
2	Trrd_l	Activate	Activate	12	same bank group, different bank	
3	Trrd_s	Activate	Activate	8	different bank group	
4	Trp	Precharge	Activate	39		
5	Tras	Activate	precharge	76	different bank	
6	Trtp	Read	precharge	18		
7	Tcwd+Tburst+Twr	Write	Precharge	38+8+30 = 76		
8	Twr	Data end	precharge	30		write to precharge delay = Tcwd+Tburst+Twr
9	Tcas / CL	Read	Data start	40		Tcwd = Tcas-2
10	Tcwd / CWL	Write	Data start	38		Other commands maybe valid at these times
11	Trcd	Activate	Write /read	39		
12	TCCD_s	Read	Read	8	different bank group	
13	TCCD_l	Read	Read	12	same bank group,same bank	
14	TCCD_l_wr	Write	Write	48	same bank group, different bank	
15	TCCD_s_wr	Write	Write	8	different bank group	
16	TCCD_s_RTW	Read	Write	16	different bank group	
17	TCCD_l_RTW	Read	Write	16	same bank Group	
18	TCCD_s_WTR	Write	Read	52	different bank group	
19	TCCD_l_WTR	Write	Read	70	Same bamk group	

1. For checking the queue:

- Using debug statements ($display) to check for the queue size for every iteration
- Sending requests at every clock cycle and check if the queue is able to push and pop all the requests
- Test to check the emptying condition of the queue. Here the requests are given at larger intervals , so as to check the number of entries in the queue.
- Test which provides inputs at varying times - from every input clock cycle to flood the queue to overflow condition, to inputs after a long time to check if the queue is emptying properly.

Trace Input

Time	Core	Operation	Bank	Bank-group	Row	Column
2	1	0	0	1	2	7
4	1	0	1	2	3	8
500	1	2	2	1	6	12

2.For checking the open page policy:

- Test with multiple back to back reads to an open page to see if the timing parameter for reads are satisfied. The purpose of this test is to simulate page hits.
- Test with reads to the same page coming after different row addresses to the same bank coming in between to simulate page misses.
- Test with requests coming to different banks to check closed page policy.
- Test to flood the same bank with all the requests to measure the performance(How the timing parameters add up).

Generated Test cases using Python

Time	Core	Operation	Address
0	6	1	00075c92d
1	2	2	000a90187
2	11	0	000ca9514
3	3	1	00075493e
4	11	1	000aa318d
5	9	0	000cb9531

Closed Page

- Hits all the different banks

Generated Test cases using Python

Examples for Above Scenarios:

Time	Core	Operation	Address
0	4	0	000ca343f
1	5	1	000f2c89c
2	8	2	003232d2e
3	0	2	00c8211ad
4	1	1	00144349e
5	10	0	00062b903
6	0	1	0001e7c32
7	0	2	000198534

Open Page/Page miss:

- **We used Python script to generate a trace file.**

Time	Core	Operation	Bank	Bank-group	Row	Column
2	1	0	0	1	2	7
4	1	0	1	2	3	8
5	1	2	0	1	3	12

Expected Output

4	0	ACT0	1	0	0x2
6	0	ACT1	1	0	0x2
84	0	RD0	1	0	0x7
86	0	RD1	1	0	0x7
182	0	ACT0	2	1	0x3
184	0	ACT1	2	1	0x3
262	0	RD0	2	1	0x8
264	0	RD1	2	1	0x8
360	0	PRE	1	0	
438	0	ACT0	1	0	0x6
440	0	ACT1	1	0	0x6
518	0	RD0	1	0	0xC
520	0	RD1	1	0	0xC

Open Page/Page Hit:

- **We used Python script to generate a trace file.**

Time	Core	Operation	Bank	Bank-group	Row	Column
2	1	0	0	1	2	7
4	1	0	1	2	3	8
5	1	2	0	1	2	12

Expected Output:

4	0	ACT0	1	0	0x2
6	0	ACT1	1	0	0x2
84	0	RD0	1	0	0x7
86	0	RD1	1	0	0x7
182	0	ACT0	2	1	0x3
184	0	ACT1	2	1	0x3
262	0	RD0	2	1	0x8
264	0	RD1	2	1	0x8
362	0	RD0	1	0	0xC
364	0	RD1	1	0	0xC

- **<u>To Check for DRAM operations (Activate, Read, Write, Precharge)</u>**
- **We used Python script to generate a trace file.**

Time	Core	Operation	Bank	Bank-group	Row	Column
2	1	1	0	1	2	7
4	1	0	1	2	3	8
5	1	2	0	1	3	12

Expected Output:

4	0	ACT0	1	0	0x2
6	0	ACT1	1	0	0x2
84	0	WR0	1	0	0x7
86	0	WR1	1	0	0x7
146	0	ACT0	2	1	0x3
148	0	ACT1	2	1	0x3
226	0	RD0	2	1	0x8
228	0	RD1	2	1	0x8
324	0	PRE	1	0	
402	0	ACT0	1	0	0x6
404	0	ACT1	1	0	0x6
482	0	RD0	1	0	0xC
484	0	RD1	1	0	0xC
