# ECE -585 - MICROPROCESSOR SYSTEM DESIGN
# TEST PLAN -GROUP 10
# Simulation of the Scheduler for a DDR5 Memory Controller


Team Members:

| | |
|---|---|
| Gagana Hirenallur Mohan Kumar<br>PSU ID: 939272712<br>Mail id: gagana@pdx.edu | Raksha Mairpady<br>PSU ID: 965104471<br>Mail id: raksham@pdx.edu |
| Rahul Jnanesh Gowda<br>PSU ID:979476354<br>Mail id: rahulj@pdx.edu | Shashi Kirana Chinchandlahalli Ravanappa<br>PSU ID: 910530955<br>Mail id: shashiki@pdx.edu |


1. Successive read request in the same row of the same bank group, bank but different column address

   Input format: 1 6 0 00012050A
   2 6 0 00013F50A

| For address : 0x 00012050A | |
|---|---|
| Byte select | 2 |
| Lower column | 2 |
| Channel | 0 |
| Bank group | 2 |
| Bank | 1 |

| For address:00013F50A | |
|---|---|
| Byte select | 2 |
| Lower column | 2 |
| Channel | 0 |
| Bank group | 2 |
| Bank | 1 |

| | |
|---|---|
| Upper column | 32 |
| Row | 4 |

| | |
|---|---|
| Upper column | 63 |
| Row | 4 |

Expected output:

ACT0 2 1 4
ACT1 2 1 4
RD0 2 1 202
RD1 2 1 202
RD0 2 1 3F2
RD1 2 1 3F2

2. Successive read requests in the different rows of the same bank group. Input format: 38 7 0 00019050A
   39 7 0 000E2050A

| For address: 0x00019050A | |
|---|---|
| Byte select | 2 |
| Lower column | 2 |
| Channel | 0 |
| Bank group | 2 |
| Bank | 1 |
| Upper column | 32 |
| Row | 6 |

| For address:0X000E2050A | |
|---|---|
| Byte select | 2 |
| Lower column | 2 |
| Channel | 0 |
| Bank group | 2 |
| Bank | 1 |
| Upper column | 32 |
| Row | 56 |

Expected output:

ACT0 2 1 6
ACT1 2 1 6
RD0 2 1 202
RD1 2 1 202
PRE 2 1
ACT0 2 1 38
ACT1 2 1 38
RD0 2 1 202
RD1 2 1 202

3. Successive read requests to the same bank group, but different banks

Input format: 40 8 0 00012090A
41 8 0 000120D0A

| For address : 0x 00012090A | |
|---|---|
| Byte select | 2 |
| Lower column | 2 |
| Channel | 0 |
| Bank group | 2 |
| Bank | 2 |
| Upper column | 32 |
| Row | 4 |

| For address: 0x000120D0A | |
|---|---|
| Byte select | 2 |
| Lower column | 2 |
| Channel | 0 |
| Bank group | 2 |
| Bank | 3 |
| Upper column | 32 |
| Row | 4 |

Expected output:

ACT0 2 2 4
ACT1 2 2 4
RD0 2 2 202
RD1 2 2 202
PRE 2 3
ACT0 2 3 4
ACT1 2 3 4
RD0 2 2 202
RD1 2 2 202

4. Successive read requests to the different bank groups, but the same bank

Input format: 40 9 0 00010C200
41 9 0 00010C380

| For address : 0x 00010C200 | |
|---|---|
| Byte select | 0 |
| Lower column | 0 |
| Channel | 0 |
| Bank group | 4 |
| Bank | 0 |
| Upper column | 12 |
| Row | 4 |

| For address: 0x00010C380 | |
|---|---|
| Byte select | 0 |
| Lower column | 0 |
| Channel | 0 |
| Bank group | 0 |
| Bank | 0 |
| Upper column | 12 |
| Row | 4 |

Expected output:

ACT0 4 0 4
ACT1 4 0 4
RD0 4 0 C0
RD1 4 0 C0
PRE 0 0
ACT0 0 0 4
ACT1 0 0 4
RD0 0 0 C0

5. Read requests to the row of the same bank group and bank, followed by write to the different row of the same bank groups and bank

Input format: 40 4 0 000817084
41 4 1 000C97084

| For address : 0x 000817084 | |
|---|---|
| Byte select | 0 |
| Lower column | 1 |
| Channel | 0 |
| Bank group | 1 |
| Bank | 0 |
| Upper column | 23 |
| Row | 32 |

| For address: 000C97084 | |
|---|---|
| Byte select | 0 |
| Lower column | 1 |
| Channel | 0 |
| Bank group | 1 |
| Bank | 0 |
| Upper column | 23 |
| Row | 42 |

Expected output:

ACT0 1 0 20
ACT1 1 0 20
RD0 1 0 171
RD1 1 0 171
PRE 1 0
ACT0 1 0 2A
ACT1 1 0 2A
WR0 1 0 171
WR1 1 0 171

6. Write to a row of the same bank group and bank followed by a request to read in the same row of the same bank and bank group

Input format: 40 4 1 00016A000
41 4 0 00016A000

| For address : 0x 00016A000 | |
|---|---|
| Byte select | 0 |
| Lower column | 0 |
| Channel | 0 |
| Bank group | 0 |
| Bank | 0 |
| Upper column | 42 |
| Row | 5 |

Expected output:

ACT0 0 0 5
ACT1 0 0 5
WR0 0 0 2A0
WR1 0 0 2A0
RD0 0 0 2A0
RD1 0 0 2A0

7. Write to a row within a specific bank and bank group followed by write to a distinct row within the same bank group and bank.

Input format: 40 5 1 000301210
41 5 1 000FC1210

| For address: 0x 000301210 | |
|---|---|
| Byte select | 0 |
| Lower column | 4 |
| Channel | 0 |
| Bank group | 4 |
| Bank | 0 |
| Upper column | 1 |
| Row | 12 |

| For address: 000FC1210 | |
|---|---|
| Byte select | 0 |
| Lower column | 4 |
| Channel | 0 |
| Bank group | 4 |
| Bank | 0 |
| Upper column | 1 |
| Row | 42 |

Expected output:

ACT0 4 0 C
ACT1 4 0 C

WR0 4 0 14
WR1 4 0 14
PRE 4 0
ACT0 4 0 3F
ACT1 4 0 3F
WR0 4 0 14
WR1 4 0 14

8. Write from a row within the same bank group and bank followed by read to distinct bank groups and bank.

Input format: 40 3 1 000510230
41 3 0 000510EB0

| For address : 0x 000510230 | |
|---|---|
| Byte select | 0 |
| Lower column | 12 |
| Channel | 0 |
| Bank group | 4 |
| Bank | 0 |
| Upper column | 16 |
| Row | 20 |

| For address: 000510EB0 | |
|---|---|
| Byte select | 0 |
| Lower column | 12 |
| Channel | 0 |
| Bank group | 5 |
| Bank | 3 |
| Upper column | 16 |
| Row | 20 |

Expected output:

ACT0 4 0 14
ACT1 4 0 14
WR0 4 0 10C
WR1 4 0 10C
PRE 5 3
ACT0 5 3 14
ACT1 5 3 14
RD0 5 3 10C
RD1 5 3 10C

9. Write from a row within a bank of a bank group followed by request to write to a distinct row and column within the same bank and bank group.

Input format: 40 5 1 00064049F
41 5 1 0009C2493

| For address: 0x00064049F | |
|---|---|
| Byte select | 3 |
| Lower column | 7 |
| Channel | 0 |
| Bank group | 1 |
| Bank | 1 |
| Upper column | 0 |
| Row | 25 |

| For address: 0X0009C2493 | |
|---|---|
| Byte select | 3 |
| Lower column | 7 |
| Channel | 0 |
| Bank group | 1 |
| Bank | 1 |
| Upper column | 2 |
| Row | 39 |

Expected output:

ACT0 1 1 19
ACT0 1 1 19
WR0 1 1 7
WR1 1 1 7
PRE 1 1
ACT0 1 1 27
ACT1 1 1 27
WR0 1 1 27
WR1 1 1 27

10. Consecutive reads to different banks in the same bank group.

Input format: 41 4 0 00084789A
41 4 0 000847C9A
41 4 0 00084709A

| For address: 0x00084789A | |
|---|---|
| Byte select | 2 |
| Lower column | 6 |
| Channel | 0 |
| Bank group | 1 |
| Bank | 2 |
| Upper column | 7 |
| Row | 33 |

| For address: 0X000847C9A | |
|---|---|
| Byte select | 2 |
| Lower column | 6 |
| Channel | 0 |
| Bank group | 1 |
| Bank | 3 |
| Upper column | 7 |
| Row | 33 |

| For address: 0X00084709A | |
|---|---|
| Byte select | 2 |
| Lower column | 6 |
| Channel | 0 |
| Bank group | 1 |
| Bank | 0 |
| Upper column | 7 |
| Row | 33 |

Expected output:

ACT0 1 2 21
ACT1 1 2 21
RD0 1 2 76
RD1 1 2 76
PRE 1 3
ACT0 1 3 21
ACT1 1 3 21
RD0 1 3 76
RD1 1 3 76
PRE 1 0
ACT0 1 0 21
ACT1 1 0 21
RD0 1 0 76
RD1 1 0 76

11. Access to same row in same bank and same bank group at the same time.
Input format: 1 6 0 00012050A
1 6 0 00013F50A

| For address : 0x 00012050A | |
|---|---|
| Byte select | 2 |
| Lower column | 2 |
| Channel | 0 |
| Bank group | 2 |
| Bank | 1 |
| Upper column | 32 |
| Row | 4 |

| For address:00013F50A | |
|---|---|
| Byte select | 2 |
| Lower column | 2 |
| Channel | 0 |
| Bank group | 2 |
| Bank | 1 |
| Upper column | 63 |
| Row | 4 |

Expected output:
ACT0 2 1 4 `

ACT1 2 1 4
RD0 2 1 202
RD1 2 1 202
RD0 2 1 3F2
RD1 2 1 3F2

12. Access to different row in same bank and bank group at the same time.

Input format: 38 7 0 00019050A
38 7 0 000E2050A

| For address: 0x00019050A | |
|---|---|
| Byte select | 2 |
| Lower column | 2 |
| Channel | 0 |
| Bank group | 2 |
| Bank | 1 |
| Upper column | 32 |
| Row | 6 |

| For address:0X000E2050A | |
|---|---|
| Byte select | 2 |
| Lower column | 2 |
| Channel | 0 |
| Bank group | 2 |
| Bank | 1 |
| Upper column | 32 |
| Row | 56 |

Expected output:

ACT0 2 1 6
ACT1 2 1 6
RD0 2 1 202
RD1 2 1 202
PRE 2 1
ACT0 2 1 38
ACT1 2 1 38
RD0 2 1 202
RD1 2 1 202

13. Access to same bank group and same bank at the same time.

Input format: 38 7 0 00015A240
38 7 0 00015A240

| For address: 0x00015A200 | |
|---|---|
| Byte select | 0 |

| For address:0X00015A200 | |
|---|---|
| Byte select | 0 |

| | |
|---|---|
| Lower column | 0 |
| Channel | 0 |
| Bank group | 4 |
| Bank | 0 |
| Upper column | 26 |
| Row | 5 |

| | |
|---|---|
| Lower column | 0 |
| Channel | 0 |
| Bank group | 4 |
| Bank | 0 |
| Upper column | 26 |
| Row | 5 |

Expected output: ACT0 4 0 5
ACT1 4 0 5
RD0 4 0 1A0
RD1 4 0 1A0

14. Access to different bank groups at the same time.

Input format: 40 8 0 00012090A
40 8 0 000120D0A

| For address : 0x 00012090A | |
|---|---|
| Byte select | 2 |
| Lower column | 2 |
| Channel | 0 |
| Bank group | 2 |
| Bank | 2 |
| Upper column | 32 |
| Row | 4 |

| For address: 0x000120D0A | |
|---|---|
| Byte select | 2 |
| Lower column | 2 |
| Channel | 0 |
| Bank group | 2 |
| Bank | 3 |
| Upper column | 32 |
| Row | 4 |

Expected output:

ACT0 2 2 4
ACT1 2 2 4
RD0 2 2 202
RD1 2 2 202
PRE 2 3
ACT0 2 3 4
ACT1 2 3 4
RD0 2 2 202
RD1 2 2 202

15. Reading the same column at different time.

Input format: 1 6 0 00012050A
2 6 0 00012050A

| For address : 0x 00012050A | |
|---|---|
| Byte select | 2 |
| Lower column | 2 |
| Channel | 0 |
| Bank group | 2 |
| Bank | 1 |
| Upper column | 32 |
| Row | 4 |

| For address:00012050A | |
|---|---|
| Byte select | 2 |
| Lower column | 2 |
| Channel | 0 |
| Bank group | 2 |
| Bank | 1 |
| Upper column | 32 |
| Row | 4 |

Expected output: ACT0 2 1 4 `

ACT1 2 1 4
RD0 2 1 202
RD1 2 1 202

16. Writing to same column at the same time

Input format: 1 6 1 00012050A
1 6 1 00012050A

| For address : 0x 00012050A | |
|---|---|
| Byte select | 2 |
| Lower column | 2 |
| Channel | 0 |
| Bank group | 2 |
| Bank | 1 |
| Upper column | 32 |
| Row | 4 |

| For address:00012050A | |
|---|---|
| Byte select | 2 |
| Lower column | 2 |
| Channel | 0 |
| Bank group | 2 |
| Bank | 1 |
| Upper column | 32 |
| Row | 4 |

Expected output:

ACT0 2 1 4 `
ACT1 2 1 4
WR0 2 1 202
WR1 2 1 202

17. Prioritizing read over write for different address at the same time.
    Input format: 40 4 0 000817084
                  40 4 1 000C97084

| For address : 0x 000817084 | |
|---|---|
| Byte select | 0 |
| Lower column | 1 |
| Channel | 0 |
| Bank group | 1 |
| Bank | 0 |
| Upper column | 23 |
| Row | 32 |

| For address: 000C97084 | |
|---|---|
| Byte select | 0 |
| Lower column | 1 |
| Channel | 0 |
| Bank group | 1 |
| Bank | 0 |
| Upper column | 23 |
| Row | 42 |

Expected output:

ACT0 1 0 20
ACT1 1 0 20
RD0 1 0 171
RD1 1 0 171
PRE 1 0
ACT0 1 0 2A
ACT1 1 0 2A
WR0 1 0 171
WR1 1 0 171

18. Out of bound checking:
    Check whether the address, bank group, bank, row, channel, column is within the boundary value specified in the spec.
    Take the value of the operations greater than 2 and check whether the condition fails or exceeds the limit specified in the spec. Similarly, testing for bank group, bank, channel, row and column.

19. Testing full and empty conditions of the queue:
    i. Check for the condition by giving the values for input CPU clocks less than the DIMM clocks. So, the queue would be empty
    ii. Check for the condition by giving values for input CPU clocks continuously up to a particular range where the frequency of the popping requests will

be less than the frequency of the inputs at CPU clock cycles. Hence, Queue would be full.

iii. Check whether the queue is full by adding and removing the request at the same time.

iv. Adding and removing the request at the same time when the queue is not full.