ECE - 585 - MICROPROCESSOR SYSTEM DESIGN
TEST PLAN - GROUP 6

Simulation of the scheduler portion of a DDR5 Memory Controller

Ashwin Kumar Sivakumar

PSU ID:977214327

Mahidhar Regalla

PSU ID:951927746

Swaroop Chandrashekar

PSU ID: 919564194

Sahil Khan

PSU ID:980980741

Abstract

This report details the simulation of the scheduling part of the Memory Controller DDR5 (Double Data Rate) which serves a 12-core 4.8Ghz processor for a single 16GB PC5-38400 DIMM in a hardware descriptive language System Verilog. The focus was on utilizing a single channel among the two available channels going to 2 different DIMMs, adopting a closed-page policy, and employing in-order execution. The design adheres to specified timing constraints derived from the provided datasheet.

Implementation Overview

The Memory Controller DDR scheduling module was implemented in System-Verilog with the following key features:

1. Single Channel Usage: The design utilizes only one channel out of the two available in the memory system.

2. Closed-Page Policy: The memory controller adopts a closed-page policy, which refers to keeping the page open until a different page is requested.

3. In-Order Execution: The requests are processed in the order they are received, ensuring a sequential and predictable execution flow.

4. Timing Constraints: The implementation adheres to the timing constraints specified in the datasheet. This includes meeting setup and hold times, ensuring proper data transfer rates, and handling various signal timings.

5. Bank-Level Considerations: Requests targeting the same bank or different banks are appropriately managed to address potential conflicts. The status of each request affecting the upcoming requests is tracked in an array.

Execution and Tools

1. The Makefile is utilized for testbench execution, providing a streamlined approach to running test cases. The tool used for program execution is QuestaSim, ensuring compatibility with the System Verilog implementation.

Testbench Implementation

Testbenches were implemented using a combination of Python and spreadsheets. For simple test cases, spreadsheets were employed to manually generate scenarios and expected outcomes. For more complex and randomized test cases, Python scripts were used to generate the testbench.

Test plan for closed-page policy (Level 0)

Scenario 1: Basic Read and Write

Purpose: Validate the fundamental functionality of the closed-page DDR5 design by executing simple read and write operations, ensuring correct data transfer.

testcaseS1T1 : Take a request to read from a particular bank group, bank, row and column which should satisfy tRCD and tRTP timing constraints.

testcaseS1T2: Take a request to write to a particular bank group, bank, row and column which should satisfy tRCD which is from activate to write and tCWL+tBURST+tWR timing constraints which is from write to pre-charge.

Expected Result: The memory controller processes a single CPU request and provides DRAM commands as output to the DIMM for the specified Bank group, Bank, row, and column making sure that all the timing constraints are met.

Scenario 2: Repeated Access within the Same banks

Purpose: Confirm that sequential read and write operations within the same bank exhibit the expected closed-page behavior by accounting the timing constraints from a read to pre-charge and write to pre-charge operations.

testcaseS2T1: take 3 requests of only read where the memory controller needs toaccess the same bank again after closing the page also satisfying 3 consecutive timings constraints of ACT, RD and PRE along with obeying the Trp timing which is specific only within the same bank and tRCD timing constraint.

For example: B0, B1, B0, ...
to the nearest CPU clock cycle.

testcaseS2T2: take 3 requests of only write where the memory controller needs to access thesame bank again after closing the page also satisfying 3 consecutive timings constraints of ACT1, WR1

and PRE along with obeying the tRP timing which is specific only within the same bank and tCWL+tBURST+tWR.

For example: B0, B1, B0, ...
to the nearest CPU clock cycle.

testcaseS2T3: take multiple requests of mixing write and read where the memory controller needs to access the same bank again after closing the page.
For example: B0, B1, B2, B3, B0, ...
to the larger gap of CPU clock cycle where tRP delay is already satisfied before the next request to the same bank enters, the operation should follow the timing constraints as follows, from one ACT1 to RD1 or WR1 which is tRCD and then from RD1 or WR1 to PRE which is tRTP for RD1 or tCWL+tBURST+tWR which is for WR1 then from pre-charge to ACT which is tRP which should be within the same bank. For a larger gap of CPU cycles the timing should be tRP+clock request.
Expected result: Check if the memory controller follows the timing delays mentioned above when accessing the same bank again.
Verify that the scheduler correctly handles the timing constraints.

Boundary Testing:

Testcase4 Test scenarios where the page closes and reopens are at the edges of tRP.

Expected result: It should start processing the ACT 0 in the immediate dimm clock after Trp

testcase5 Test scenarios where the page close and reopen are at the middle of tRP.

Expected result: It should start processing the ACT 0 after satisfying the remaining Trp delay

---

Scenario 3: Sequential Access to Different Banks

Purpose: Assesses the sequential capability of the memory controller by issuing read and write requests to different banks simultaneously to exhibit the expected close pagebehavior without considering Trp delay.

Create test cases to sequentially access different banks without considering Trp delay.
For example: B0, B1, B2, ...
Verify Sequential Bank Access:
Ensure that the memory controller schedules commands for sequential bank accesses correctly. Check the output to confirm that accessing different banks in sequence doesn't introduce Trp delay.

testcaseS3T1: when accessing different banks within the same bank group tRP delay won't be considered and only the instruction delays which are ACT1 To RD1 then from RD1 to PRE will be considered for a read operation. For example: B0, B1, B2, ... for the nearest CPU clock cycle.

testcaseS3T2: when accessing different banks within the same bank group tRP delay won't be considered and only the instruction delays which are ACT1 To WR1 then from WR1 to PRE will be considered for a write operation. For example: B0, B1, B2, ... for the nearest CPU clock

cycle.

testcaseS3T3: take 4 or more instructions that are sequential mixed up with writes and readsfor the different bank group, bank, and row to the nearest CPU clock cycle. Here tRP wont be considered and will follow the timing constraints from ACT1 to RD1 and WR1 instructions then from RD1 and WR1 to PRE and the following request will be in different bank.

Expected result:

Ensure that the memory controller schedules commands for sequential bank access correctly. Check the output to confirm that accessing different banks in sequence doesn't introduce tRP delay.

Scenario 4: Random Bank Access:

Purpose: Created **testcaseS4** with random bank accesses, including both sequential to differentbanks and repeated accesses to same banks.
Mix and match different banks and patterns to simulate real-world scenarios.

Expected result:

Ensure that the memory controller schedules commands for sequential bank accesses to different banks and repeated access to the same bank correctly.
Check the output to confirm that accessing different banks in sequence doesn't introduce tRP delay and accessing to the same bank introduces tRP delay.

Scenario 5: Timing constraints check

Purpose: Evaluate the timings tRP,tRCD, tRTP, ,tburst following read, write and sequential read and write or write and read operations to ensure optimal memory access and prevent unnecessary delays.

testcaseS5T1: read instruction we will check if the timing between activate and read which is tRCD and the timing after the read to pre-charge which is tRTP , the timing between pre-charge to activate which is tRP which should be for the same bank .

testcaseS5T2: write instruction we will check for the timing between activate and write which is tRCD and the timing between write to pre-charge which is Tcwl+tBURST+tWR and the timing between pre-charge to activate which is tRP which should be for the same bank .

testcaseS5T3: successive read and write instruction we will check if the timing between activate and write or read which is tRCD and the timing after the write to pre-charge which is tCWL+tBURST+tWR or timing between the read to pre-charge which is tRTP and timing between pre-charge and activate to another row instruction which tRP for the same bank.

Note: DRAM commands like ACT0, ACT1, RD0, RD1, WR0, WR1 and PRE each will take one

Expected output:

tRCD: The time between the activate command and the start of the read or write operation meets the specified tRCD timing.

tRTP: The time after the read operation to pre-charge meets the specified tRTPtiming.
tCWL+tBURST+tWR: The timing between write to pre-charge meets the specified timing.

tRP: The time between pre-charge and activate to another row instruction meets the specified tRP timing which should be within the same bank.

DRAM commands like ACT0, ACT1, RD0, RD1, WR0, WR1 and PRE each will take oneDIMM clock cycle.

Scenario 6: check for Page empty

Purpose: To check the handling for page empty

testcaseS6T1: to check page empty give 2 requests, issuing first request at 0 CPU clock cycle and do not issue the second request until the first request is complete so that the next ACT will not be issued for long time which makes it a page empty.

Expected result:
To check for the page empty scenario by violating the tRP for the same banks for the next instruction ACT

Scenario 7: Back-to-Back Operations

Purpose: Assess the ability of the memory controller to handle consecutive read or write operations issued in rapid succession, checking for sustained performance.

testcaseS7T1: take 20 consecutive reads to the same bankgroup, bank, row and check for the functionality by the analysis of the obtained data which includes timing constraints within ACT1 to RD1 which tRCD then RD1 to PRE which is tRTP and the timing between pre-charge to activate which is tRP when it is only within the same bank .
for all the 20 consecutive read operations.

testcaseS7T2: take 20 consecutive writes to the same bankgroup, bank, row and check for the functionality by the analysis of the obtained data which includes timing constraints within ACT1 to WR1 which tRCD then WR1 to PRE which is tCWL+tBURST+tWR and the timing between pre-charge to activate which is tRP for all the 20 consecutive write operations.

testcaseS7T3: take consecutive 10 reads to the same address i.e same bankgroup, bank,row andcolumn to check for the functionality by the analysis of the obtained data which includes timing constraints within ACT1 to RD1 which tRCD then RD1 to PRE which is tRTP and the timing between pre-charge to activate which is tRP for all the 20 consecutive write operations.

testcaseS7T4: take consecutive 10 writes to the same address i.e same bankgroup, bank,row andcolumn to check for the functionality by the analysis of the obtained data which includes timing constraints within ACT1 to WR1 which tRCD then WR1 to PRE which is tCWL+tBURST+tWR and timing between pre-charge to activate which is tRP. for all the 20 consecutive write operations.

Expected output: The memory controller successfully manages consecutive read/write requests to the same address and also for the same row without notable performance degradation obeying all the timing constraints.

Analyze the obtained data to confirm that each read/write operation correctly stores the specified data in the designated address.

Sustained performance is demonstrated by the consistent and timely completion of all10 consecutive read/write operations to the same address.

Scenario 8: Boundary Testing

Purpose: Explore edge cases, such as out of boundary for addresses, ensuring the closed-page DDR5 design handles extreme scenarios robustly.

check the constraints including operation and channel are obeying within the defined range of condition.

testcaseS8T1: Operation- Take the value of operation greater than 2 and test whether if this condition is failing as it is exceeding the limit of the designed range.

testcaseS8T2: Channel bit- Check this testcase by giving the channel bit as 1 and check for the failure ofboundary condition as we have defined the design for channel 0.

Expected Output:

The test should result in a failure, indicating that the channel bit or operation input is not within the defined range.

The memory controller should reject the request and may issue an error or exception.

Scenario 9: Queue testing

Testcase1: Check for queue empty

Purpose: Check for queue empty, the expected result is to confirm that, after processing all given input requests, the memory controller accurately recognizes and reports the empty queue condition.

Check for the condition by giving a set of requests to a nearer CPU clock cycle and wait until allthe requests are satisfied and queue becomes empty.

Expected Result: if all the given input requests are satisfied and no request is pending then display-“queue empty”.

Testcase 2: Check for queue full

Purpose: Verify that the memory controller correctly handles the case when the queue is full.

Test Steps: Fill the memory controller queue to its capacity. Attempt to add more operations to the input queue. Verify that the memory controller stalls the input operations until there is space in the queue.

Expected Result: The memory controller correctly stalls input operations when the queue is full.

Testcase3: Check for queue full b y adding and removing request at the same time.

Purpose: The purpose is to assess the system's ability to handle the complexity of both operations occurring concurrently under full queue conditions. The expected result is to verify whether the memory controller correctly recognizes a full queue and effectively manages the simultaneous addition and removal of requests

Expected result: verify if the queue is full and check at the same time for add or remove requests happening simultaneously.

Testcase4: Check for adding and removing request at the same time without queue is full.

Purpose: The purpose is to verify the controller's correct synchronization during concurrent pushing and popping operations at even CPU cycles.

Check this condition by giving the input at the CPU at an even cycle which is at the DIMM clock, then the input scheduled at that is pushed into the queue and then the request at the frontof the queue is popped simultaneously. Here pushing and popping happens at the same CPU cycle when the queue is not full.

Expected output: The memory controller successfully handles the simultaneous addition and removal of requests when the queue is not full.

The requests are pushed into the queue, and the request at the front of the queue is poppedsimultaneously.

The memory controller returns the expected result, indicating correct handling of simultaneous adding and removing when the queue is not full.

Scenario 10: CPU clock instruction

Purpose: To check if the functionality is working properly for request in different CPU clock cycles.

testcaseS10T1: To test if there is new instruction on CPU clock after a long gap.

Check this by taking two consecutive requests scheduled with a larger time interval.

Expected output:

The memory controller should detect the new instruction on the CPU clock after a long gap.

The requests scheduled with a larger time interval should be processed, and the memory controller returns the expected result indicating the successful execution of the instructions.

testcaseS10T2: To test if there is new instruction on every CPU clock without any gap.

Check this by taking two consecutive requests scheduled with consecutive time interval without gap.

Expected output: The memory controller should detect the new instruction on every CPU clock without any gap.

The requests scheduled with consecutive time intervals should be processed efficiently, and the memory controller returns the expected result indicating the correct handling of consecutive instructions.

testcaseS10T3: Test for zero simulation/CPU clock time

This is to be tested by giving a request at the zeroth simulation time and observing the transcript.

Expected output: The memory controller should detect the request at the zeroth simulation time.

The request should start processing without any issues at the first DIMM time, and the memory controller returns the expected result indicating the correct handling.

Testcase4: Test for giving request at the maximum simulation time/CPU clock.

This is to be tested by scheduling a request at the maximum value of CPU clock time.

Expected output: The memory controller should detect the request at the maximum simulation time. The request should be processed without any issues, and the memory controller returns the expected result indicating the correct handling of a request at the maximum simulation time.

Scenario 11: Stress Testing

Purpose: Apply stress to the system by continuously issuing read and write requests, evaluating the design's stability and sustained performance satisfying the applicable timing constraints.

testcaseS11T1: take a very larger number of consecutive read requests to the samebank group, bank, row and check for the functionality by the analysis of the obtained data. check for which includes timing constraints within ACT1 to RD1 which tRCD then RD1 to PRE which is tRTP and the timing between pre-charge to activate which is tRP when it is only within the same bank .

Expected result:

The memory controller should handle the stress by processing the consecutive read requests efficiently.

testcaseS11T2: take a very larger number of consecutive write requests to the samebank group, bank, row and check for the functionality by the analysis of the obtained data. check for which includes timing constraints within ACT1 to WR1 which is tRCD then WR1 to PRE which is tCWL+tBURST+tWR and the timing between pre-charge to activate which is tRP for all the 20 consecutive write operations.

testcaseS11T3: take a very larger number of consecutive successive read and write instruction we will check if the timing between activate and write or read which is tRCD and the timing after the write to pre-charge which is tCWL+tBURST+tWR or timing between the read to pre-charge which is tRTP and timing between pre-charge and activate to another row instruction which tRP only when it is for the same bank.

Expected result:

The memory controller should handle the stress by processing the consecutive write meeting the relevant timings constraints requests efficiently.

Analyze the obtained data to ensure that each read/write operation retrieves the correct data from the relevant timing constraints request efficiently.

Scenario 12: Complete Simulation

Purpose: Run a complete simulation with a sequence of read and write operations, queue full scenarios, and empty queue scenarios.

Test Case steps:

Add a sequence of read and write operations to the input queue.
Introducescenarios where the queue becomes full.
Introduce scenarios where the queue becomes empty.

Verify that the memory controller processes operations correctly and produces the expected output.

Expected Result: The memory controller handles various scenarios correctlythroughout the simulation.

Scenario 13: Debugging Information

Description: Verify that the memory controller produces useful debugging information when debug_en is set.

Test Case Steps:

Set debug_en to 1.

Run a simulation with various operations.

Verify that the debugging information in the output file is informative.

Expected Result: The debugging information in the output file helps identify any issues or unexpected behavior.

Conclusion

The implemented Memory Controller DDR5 scheduling module successfully meets the specified requirements, adheres to timing constraints, and provides robust handling of different types of requests. The combination of manual and automated testbenches ensures a thorough validation of the design's functionality and performance under various scenarios.
