Test Plan

This test plan was created to test the programs ability to handle various situations that may break the program in addition to testing the timing constraints of our simulation.

Potential Input Problems

Potential Problem	Should Test?	Test Trace Filename	Solution	Test Case
Combination of Spaces/Tabs between the time, command, and address	Yes	extraWhitespace_test.txt	Using fscanf to gather the arguments and ignore the whitespace.	Have input test use different spacing and tabs.
Empty Lines in Input Files	No		The design document describes very specific input formatting and describes the fields and that they are separated by one or more spaces or tabs. It does not indicate that the user can include empty lines in their input, so we assume that this should generate an error.	
Comments in Trace File?	No		The design document describes very specific input formatting and describes the fields and that they are separated by one or more spaces or tabs. It does not indicate that the user can include comment lines or comment out traces, so we assume that this should generate an error.	
Out of range addresses? This could be an issue if the row is too large. The address is given in hexadecimal, we can pull out all of the organization bits, but the row could be too large where it would reference an invalid address. We can either ignore this and mask the extra bits, or end the simulation and provide an error.	Yes	rowOutOfRange_test.txt	Add a check to the parser where it looks to see if the upper address bits that would be assigned to the 15 bits of the row address are out of range (larger than 0x7FFF or 32767) as this would be larger than the number of rows in the memory. Exits with an error.	A simple test case has an address of 0x11FFFFFFFF which is larger than the possible address.
Negative time inputs	No		Dr. Faust said that the trace file will be monotonically increasing and never negative, so we don't need to test for this condition.	

Potential Problem	Should Test?	Test Trace Filename	Solution	Test Case
Time overflow	Yes	timeOverflow_test.txt	Dr. Faust said that we can decide what occurs if the elapsed CPU time overflows the size we have given it which is an unsigned long long int. We have decided that this should generate an error if the time overflows.	A simple test case tries to read at time 18446744073709551616 which is larger than an unsigned long long integer can hold. This should trigger an end simulation message.

Timing Constraint Tests

Potential Problem	Should Test?	Test Trace Filename	Solution	Test Case
Multiple commands received at the same CPU clock cycle	Yes	sameClockCycle_test.txt	Per Dr. Faust, we decided to have only one item be added to the queue per CPU clock cycle. So any duplicate times will be added on the following clock cycle and not before.	Multiple requests on the same clock cycle twice. Fill the queue, should continue operating.
Access successive bank groups and banks in the same row and column	Yes	allBanksBankGroups_test.txt	Timing should have all of the possible bank groups and banks accessed sequentially and should have no precharge commands.	16 requests all on the same byte select, row, column, with only bank group and bank changing through all possible combinations.
Read	Yes	readSingle_test.txt	Basic read, checking timing. Should be Activate, Read.	Single read to confirm basic timing. Confirms tRCD = 24.
Write	Yes	writeSingle_test.txt	Basic write, checking timing. Should be Activate, Write.	Single write to confirm basic timing. Confirms tRCD = 24.
Instruction Fetch	Yes	ifetchSingle_test.txt	Basic instruction fetch, should be the same as the Read.	Single instruction fetch to confirm basic timing. Confirms tRCD = 24.
Read from an Open Page Later	Yes	readOpenRowLater_test.txt	Basic read and then another read later from that open row	Two reads, one that happens much later from the open page. Tests open page policy.
Write from an Open Page Later	Yes	writeOpenRowLater_test.txt	Basic write and then another write later from that open row	Two writes, one that happens much later from the open page. Tests open page policy.

Potential Problem	Should Test?	Test Trace Filename	Solution	Test Case
Two successive Reads from the same Bank and BG	Yes	readSameBGB_test.txt	Read from the same bank group's bank twice.	Should confirm that delay is happening correctly when accessing same row twice. Tests tCCD_L = 8.
Two successive Writes from the same Bank and BG	Yes	writeSameBGB_test.txt	Write from the same bank group's bank twice.	Should confirm that delay is happening correctly when accessing same row twice. Tests tCCD_L = 8.
Two successive iFetches from the same Bank and BG	No			Same as reading
Two successive Reads from different Bank Groups	Yes	readDiffBG_test.txt	Read from two different bank groups	Should be act, act, read, read. Tests tRRD_S = 4 and tCCD_S = 4.
Two successive Writes from different Bank Groups	Yes	writeDiffBG_test.txt	Write from two different bank groups	Should be act, act, write, write Tests tRRD_S = 4 and tCCD_S = 4.
Two successive iFetches from different Bank Groups	No			Same as reading
Two successive Reads from the same BG different bank	Yes	readSameBGDiffB_test.txt	Read from the same bank group two different banks.	Should confirm that delay is happening correctly when trying to activate the same bank group again. Tests tRRD_L = 6 and tCCD_L = 8.
Two successive Writes from the same BG different bank	Yes	writeSameBGDiffB_test.txt	Write from the same bank group two different banks.	Should confirm that delay is happening correctly when trying to activate the same bank group again. Tests tRRD_L = 6 and tCCD_L = 8.
Two successive iFetches from the same BG different bank	No			Same as reading
Three successive Reads from the same Bank and BG but different rows	Yes	readThreeDiffRows_test.txt	Read from the same bank and bank group 3 times but different rows	Tests precharge to precharge delays. Tests tRAS = 52, tRP = 24 and tRCD = 24 and tRC = tRAS + tRP = 76
Three successive Writes from the same Bank and BG but different rows	Yes	writeThreeDiffRows_test.txt	Write from the same bank and bank group 3 times but different rows	Tests precharge to precharge delays. Tests tCWD(CWL) + tBurst + tWR = 20 + 4 + 20 = 44 which is the minimum time after a write that a precharge can be issued which is longer than tRAS. Also tests tRP = 24 and tRCD = 24.
Three successive iFetches from the same Bank and BG but different rows	No			Same as reading

Potential Problem	Should Test?	Test Trace Filename	Solution	Test Case
Three successive reads. One, then a pause, second to same row, bank, bank group, then a third to same bank, bank group, but a different row.	Yes	readTwoLongWaitOneRTP_test.txt	Check to ensure tRTP is satisfied	Tests tRTP = 12 and tRP = 24. tRTP is a much shorter time period that needs to be satisfied than tRAS, but tRAS has already been satisfied because there is a long time before the second read.
Three successive writes. One, then a pause, second to same row, bank, bank group, then a third to same bank, bank group, but a different row.	Yes	writeTwoLongWaitOneRTP_test.txt	Check to ensure tRTP is satisfied	Tests tRTP = 12 and tRP = 24. tRTP is a much shorter time period that needs to be satisfied than tRAS, but tRAS has already been satisfied because there is a long time before the second write.
Read then write on same row	Yes	readWriteSameRow_test.txt	Check to ensure read then write timing.	Tests read to write latency difference between CL and WL where WL is smaller so there could be bus contention. Extra padding done to ensure that tCCD_L is long enough to allow the read to complete before the write starts using the bus.
Read then write on different rows	Yes	readWriteDiffBGB_test.txt	Check to ensure read then write timing and that bus contention doesn't occur.	Tests read to write latency difference between CL and WL where WL is smaller so there could be bus contention. Extra padding done to ensure that tCCD_L is long enough to allow the read to complete before the write starts using the bus.
Read then write then read on same row	Yes	readWriteReadSameRow_test.txt	Check to ensure read, write, read timing and that tWTR is satisfied.	Tests same as above and tWTR_L + tBurst + tCWD = 12 + 4 + 20 = 36. Can't read again on the same row until the write is finished and the write to read delay is satisfied.
Read then write then read on same row	Yes	readWriteReadDiffBGB_test.txt	Check to ensure read, write, read timing across banks and that tWTR is satisfied.	Tests same as above and tWTR_S + tBurst + tCWD = 4 + 4 + 20 = 28. Can't read again on the same row until the write is finished and the write to read delay is satisfied.
Read then write then read across all banks and bank groups.	Yes	readWriteReadAllBanksBankGroups_test.txt	Check to ensure read write read timing across banks all works.	Tests above read write read conditions over all banks and bank groups for validation

Table of Timing Constraints and the Traces that Test Them

Timing Parameter	Traces that test this Parameter
tRC (time between successive row accessed to different rows = tRAS + tRP)	readThreeDiffRows_test.txt writeThreeDiffRows_test.txt

Timing Parameter	Traces that test this Parameter
tRAS (time between ACTIVATE command and end of restoration of data in DRAM array which allows a precharge to then be issued)	readThreeDiffRows_test.txt writeThreeDiffRows_test.txt
tRRD_L (row to row delay, for ACTIVATEs to the same bank group)	readSameBGDiffB_test.txt writeSameBGDiffB_test.txt
tRRD_S (row to row delay, for ACTIVATEs to a different bank group)	readDiffBG_test.txt writeDiffBG_test.txt
tRFC (Refresh cycle time/recovery delay or time to complete a refresh command)	
CWL or tCWD (CAS Write Latency is the delay between WRITE and the availability of the first bit of input data)	writeThreeDiffRows_test.txt
tCAS or CL (Time between READ command and the first data valid)	readWriteSameRow_test.txt readWriteReadSameRow_test.txt
tRCD (minimum time between ACT command and READ or WRITE command)	readSingle_test.txt writeSingle_test.txt ifetchSingle_test.txt
tWR (write recovery time, minimum time between the end of write data burst and the start of a precharge command)	writeThreeDiffRows_test.txt
tRTP (read to precharge time, must also satisfy tRAS)	readTwoLongWaitOneRTP_test.txt writeTwoLongWaitOneRTP_test.txt
tCCD_L (column to column delay for READ or WRITE to the same bank group)	readSameBGDiffB_test.txt writeSameBGDiffB_test.txt readSameBGB_test.txt writeSameBGB_test.txt
tCCD_S (column to column delay for READ or WRITE to a different bank group)	readDiffBG_test.txt writeDiffBG_test.txt
tBURST (data burst duration)	writeThreeDiffRows_test.txt
tWTR_L (write to read turn around time, minimum time between the last bit of the write data burst before a read command can be issued on the same bank group)	write_to_read_turnaround.txt
tWTR_S (write to read turn around time, minimum time between the last bit of the write data burst before a read command can be issued on a different bank group)	write_to_read_turnaround_different_bank_group.txt
REFI (required average refresh interval of REFRESH commands)	