-Mention your doubts in this document. We will address these questions during the session.

1. Sir, not a doubt just a suggestion, it seems that Assignments and class teaching have become two different things. Please make it a little similar to what is being taught in class. It reduces the burden.
2. Sir implementing the features from what's taught in class are getting more extensive and require more of implementing skills rather than understanding concepts of computer architectures. If possible please reduce implementation load in assignments and increase content relevant to comp arch.
3. Also People With Good Implementation Skills In C++ are getting benefit rather than who understands comp arch More (I disagree, I disagree +1)
4. Sir sessions like this regularly would have been a good idea it now seems I was doing everything wrong till now
5. Do we have to do some sort of pipelining implementation in the Assignment?
6. What optimizations do we need to perform in the assignment?
7. Can you release the rubric of the assignment beforehand? Because we can know what you are expecting, and we can work on those features. -> we provided basic distribution of marks for assignment-5 in the statement. We can not provide rubric now itself. We discussed what we are expecting from the assignment. You may not find out all the possibilities to improve upon your implementation. But come up with optimizations to improve Throughput.
8. Do we need to modify only one register in one clock cycle for assignment-5? -> Yes, that is how MIPS processor works. Register file has 2 read ports and 1 write port. Refer chapter 4 of textbook. Page 247
9. Should we implement starvation in the assignment? Is it necessary? -> Ignore
10. What optimisations are done in the compiler that we don't need to implement in the Request manager? Like our TA told us that you don't need to ignore some redundant lw instructions as this is compiler optimisation. Please clarify this as it will greatly affect the implementation of the request manager. -> continuous lw instructions to same register will be treated by compiler as redundant code and removes such instructions from the program. - Divya Praneetha Ma'am but we are making an interpreter, can we do this?
    -> Usually compiled code won't have such instructions. As you are writing a program like that, you would consider any case. Think of cases which occur in reality.
11. For request manager in Assignment 4, we used maps in C++. Now implementing a map-like structure in hardware can take many cycles. So should we change the data structure we are using or I am misinterpreting the meaning of clock cycles? -> addressed duing session.
12. At the time of Row Writeback, should the MRM have decided what would be the next row? For example I can also do a row writeback now, and just before starting row activation I decide which row to activate. -> You have to think about MRM logic yourself.
13. If the MRM takes multiple cycles, then it is possible that during that time we execute some lw/sw statements and put them into the DRAM buffer. So, do we stop execution while MRM takes a decision or we make another buffer to temporarily store the lw/sw requests? -> We giving a solution will limit your implementation to only one solution. Advantage of giving an

open ended problem will dissolve. We advise you to think to yourself how it should be implemented. There would be trade-offs in doing certain way than doing in other ways. You can report what are the limitations in your implementation and if they can be handled with simple changes, we advise you to implement them instead of reporting as weakness.

14. During A4 Demo, our TA told us that we cannot perform two write operations on two different registers (one due to a completed DRAM request and one due to a main instruction like addi). Can you please elaborate on this? I was of the understanding that DRAM and CPU are completely independent entities and they can independently read/write on registers. -> Refer chapter 4 of textbook. Page 247
15. Can we implement a seperate queue for every core? <- <- new -> You can implement separate queues. Finally your implementation should be feasible and efficient.
16. Can you explain how starvation is a better optimisation technique? I mean if a request is not executed for a long time, then it is because the registers used in that instruction are not used anywhere else in the program. In that case, there is no reason to execute an instruction just because it is pending for a long time. In my opinion, starvation will increase the total cycles and decrease throughput.
17. We have been asked not to implement starvation but since we are asked to use finite sized buffer so won' t starvation automatically stop if we use finite sized buffer. -> Not really. You can have requests like 1000, 3000, 1004, 1008, 1012, 1016…. All accessing same row-0 except access to 3000. When ever one such request leaves the finite wait buffer, a new request comes which may need to access row-0 only. In such a case access to 3000 location is starved. Ignore implementation for starvation handling.

18. Let us say a load word instruction took 12 cycles, then the actual register write is only in the last cycle? Only in the last cycle are we forbidden to write from another instruction say addi? -> You are giving a case like you know lw takes 12 cycles. But in reality we don't know when will it complete. Don't make implementation by counting on 12 cycles since the load has started. Hardware doesn't work that way. When 2 register writes comes in same cycle, then delay one of the operations. -> You didn't understood my question. Ignore the "12" cycles. Just wanted to ask whether we are forbidden to write the two registers only in the last cycle of the completion of a load instruction? Or in simpler words, does the DRAM request which writes into a register in the register file, this write operation is occurring only at the last cycle of the complete DRAM operation?
19. Ma'am in point 10, you have asked us to remove redundant statements. But we are designing an interpreter and not a compiler, so can we do this? -> answered in point 10 Ma'am but in Assignment 3 and 4 we were told we are making an interpreter and we can't do this. But it seems now that we have to make a compiler
20. Sir please consider deadline extension -> We will make a separate post regarding this on Piazza.
    Reasons:
    A lot of clarifications

Extensive testing is part of the assignment. Flaws in your implementation will be observed only by running various tests. We want you to perform many tests, observe drawbacks in your implementation and improve upon it.