COMP3710, Computer Microarchitecture

Final Exam (Weight: 40%)
Due date: November 14, 2021 (13:00 pm)
Total Points: 100

Important Instructions: (1) Write down your name and UID on the first page of your submission. (2) Submit the submission as a single pdf file.

Logistics: This exam is an individual effort and submission.

Note: Please practice concision. Be specific. Note that a "yes/no" style of answering would lead to no credit. Avoid ambiguity. Hiding the most crucial information deep in the text would lead to reduced credit. Use good headings, colors, and bold text to your advantage. Attempt all questions and do not be intimidated by open-ended questions.

Advice for open-ended questions: Be succinct and to the point. Use itemized lists. Use diagrams whenever necessary. Clearly explain the diagrams in one or two sentences. It is crucial to provide/explain the high-level ideas of your approach first. Then explore in detail one or two aspects of your proposed approach. Positively ignore tedious implementation details and corner cases.

Submission: Please submit a single Pdf file to the email below:
comp3710arch2022@gmail.com

Design Questions (45 points)

(10 points)

Q1. A processor manufacturer is considering an ARF+ROB pipeline due to ease of implementation and low verification cost. However, the manufacturer wants to implement the no-data-capture policy, a feature that is native to the PRF pipeline. Design an ARF+ROB pipeline that implements a no-data-capture policy, i.e., instructions read their operands after the issue stage. Specifically, when the instruction is selected for execution and sent to the execution unit, it reads the operand values from ARF or ROB just prior to execution.

1. Briefly explain your "big idea" and the proposed technique(s) to tackle this problem.
2. What changes does your proposal require to the ARF, the ROB, or any other structures?
3. What additional actions does an instruction need to perform in the issue stage, the execute stage, the writeback stage and any of the front-end stages?
4. Explain the new instruction commit policy from the head of the ROB?

(20 points)

Q2. It has been noticed that register writes in many programs exhibit a significant amount of value locality. This finding opens new horizons for the microarchitect. If the results of many instructions can be predicted before they are executed and issued, then dependent instructions no longer need to wait for parent instructions to finish execution. Value prediction attacks RAW hazards. Therefore, a processor manufacturer is considering the addition of a value prediction unit to boost their processor's performance. The manufacturer wants to limit the scope of the value predictor to load instructions because they take a long time to resolve (e.g., an on-chip data cache miss takes 100 cycles to resolve). A load value predictor exploits value locality: most of the times, a load instruction retrieves a value from memory that matches a previously seen value for the same address. Your task in this question is to design an OOO pipeline with load value prediction.

Explore the problem of adding value prediction to the ARF+ROB pipeline from as many angles as possible, including context selection, predictor placement, splitting the load operation into generating the address and performing the actual load from memory, support for speculative execution of dependent instructions, verifying predictions, and a recovery mechanism. Briefly explain what changes your approach requires to the major structures of an ARF+ ROB pipeline, including the issue queue, the functional units, load/store queues, and the ROB. Provide brief arguments for why your proposed approach will reduce the penalty of long-latency load instructions.

(5 points)

Q3. Consider a PRF pipeline with no back-end architectural map table (AMT). The active list (AL) contains the previous logical-to-destination register mappings. The front-end register map table (RMT) is in a speculative state. A microarchitect is considering different rollback strategies to repair the front-end register map table (RMT) on an exception. Explain why these two strategies are both legitimate candidates for repairing the RMT: (1) walking the AL backwards from tail to head and incrementally installing the previous mappings into the RMT (2) walking the AL forwards from head to tail. Briefly explain which of the two strategies is simpler to implement. Which one requires maintaining additional state during the rollback process?

(5 points)

Q4. A 32-bit system uses a virtually indexed, physically tagged 4-way set-associative Level-1 cache with a total (data) capacity of 32 KB. The system uses a page size of 4 KB. Provide two ways in which the OS virtual to physical page mapping can lead to abnormal cache behavior. How should an architect design a correctly operating cache without changing the 32 KB cache capacity?

(5 points, 2.5, 2.5)

Q5. Consider a 16 KB PIPT Level-1 2-way set-associative cache. The system has a page size of 4 KB. The low-order thirteen bits of the physical address are used to index the cache. Unfortunately, due to a manufacturing defect, the thirteenth bit for indexing the set in the cache is permanently stuck at ground (bit is always zero). An architect is trying to gain insight into the consequences of deploying the faulty system in real-life. Multiple users will share the system. Explain the implications of deploying this system in practice from the viewpoint of (1) program correctness, and (2) cache utilization and overall performance.

Analytical Questions (55 points)

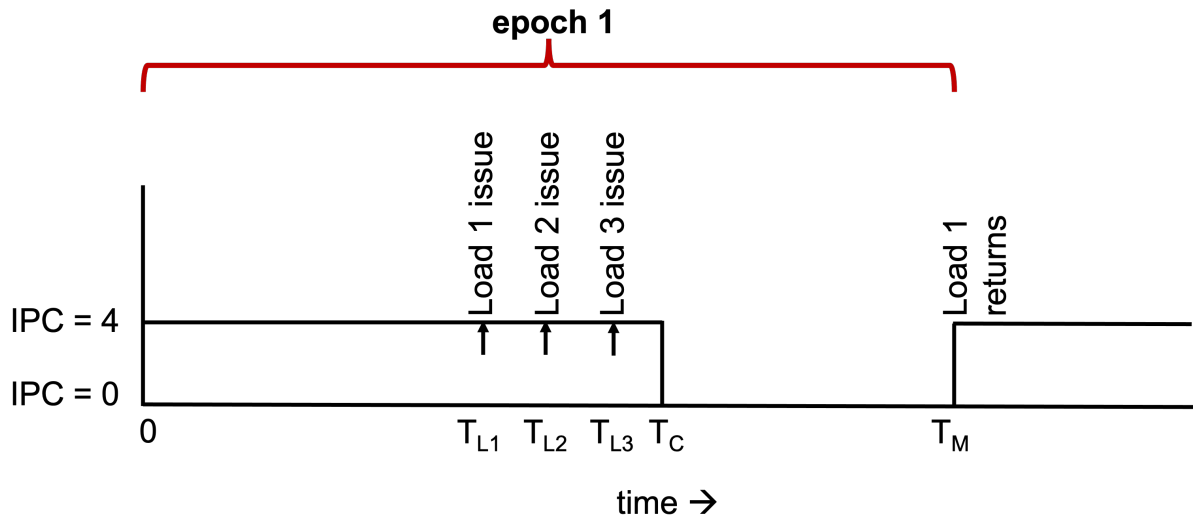
(5 points)

Q1. A processor manufacturer is struggling to choose the size of the active list (active/instruction window) for their upcoming processor. They know that the processor will be used in conjunction with a memory system with the following property: each memory access (i.e., cache miss) will take 100 cycles. Moreover, the processor is expected to run a critical workload with sparse memory accesses, i.e., one long-latency memory operation every few hundreds of instructions. The manufacture has already decided to use an issue width of four, i.e., 4-wide superscalar. What is a reasonably sized active list for this processor?

(5 points)

Q2. A microarchitect is trying to estimate the potential reduction in execution time for a code sequence by doubling the frequency of their 4-issue superscalar processor (from 1 GHz to 2 GHz). They have made the following observations by executing the code sequence at 1 GHz: (1) In the beginning, the OOO (ARF+ROB) pipeline executes instructions at full capacity (four instructions) (2) At some point, the pipeline issues three independent memory (load) operations to the memory system one after the other (3) The pipeline continues operation for some time after the third load is issued to the memory system (4) the pipeline eventually comes to a halt (5) When the first load returns from memory, the OOO pipeline restarts execution. The following diagram depicts this scenario. Each epoch relates to one iteration of the code sequence. You should focus only on the first (iteration) epoch. The X-axis represents time, and the Y-axis represents two modes of the pipeline (1) working at full capacity (IPC = 4) and (2) waiting for a response from memory (IPC = 0). Which following *estimation* of the execution time of the first epoch at 2 GHz is correct? Briefly justify your choice.

1. $T_{L1}/2 + (T_M - T_C)$
2. $T_{L2}/2 + (T_M - T_C)$
3. $T_{L3}/2 + (T_M - T_C)$
4. $T_C/2 + (T_M - T_C)$
5. $T_{L1}/2 + (T_M - T_{L1})$



(5 points)

Q3. A processor has a split load store queue (LSQ) with 32 entries each. The processor executes load and store operations whenever their addresses are ready. A store operation (S1) is dispatched in this out of order pipeline with an SQ_index (store's entry in the store queue) of 17. The current LQ_tail is positioned at index 14. By the time S1 is ready to execute, ten new loads have been dispatched to the load queue. The load addresses at the following LQ indices are aliases of the S1 address: 15, 18, 21, and 23. Which of the speculative loads does S1 needs to consider as part of its execution? Which specific load is canceled (i.e., have its mispredict bit set in the active list)? Briefly explain how the mispredict bit of the active list entry is set.

(5 points, 2.5, 2.5)

Q4. A program executes 2,000,000 memory references. When run on a system containing a particular cache, the cache has a miss rate of 7 percent, of which 1/4 are compulsory misses, 1/4 are capacity misses, and 1/2 are conflict misses.

a) Suppose the only change an architect is allowed to make to the cache is to increase the associativity. What is the maximum number of misses that they can hope to eliminate?

b) If the architect is allowed to both increase the cache size and increase the associativity, what is the maximum number of misses that they can hope to eliminate?

(5 points)

Q5. Consider a 16 MB 16-way Level-3 cache that is shared by two programs A and B. There is a mechanism in the cache that monitors cache miss rates for each program and allocates 1-15 ways to each program such that the overall number of cache misses is reduced. This technology in recent Intel processors is called the Cache Allocation Technology (CAT). Assume that program A has an MPKI of 100 when it is assigned 1 MB of the cache. Each additional 1 MB assigned to the program A reduces the MPKI by 1. Program B has an MPKI of 50 when it is assigned 1 MB of cache; each additional 1 MB assigned to program B reduces its MPKI by 2. What is the best allocation of ways to programs A and B if minimizing the overall MPKI is the goal? Is MPKI the correct metric to best exploit a technology such as CAT? Which metric must the cache controller consider in addition to MPKI?

(5 points, 2.5, 2.5)

Q6. A system has 48-bit virtual addresses, 36-bit physical addresses, and 128 MB of main memory. If the system uses 4096-byte pages, how many virtual and physical pages can the address spaces support? How many page frames of memory are there?

(5 points)

Q7. A processor has 32-bit virtual and physical addresses. The page size is 4 KB, and the processor's TLB has 128 entries and is a 4-way set-associative. How much storage is required for the TLB?

(5 points, 2.5, 2.5)

Q8. What is the maximum memory capacity supported by the following two servers with a single processor socket? Assume a 64-bit word size.

a) Two memory controllers per processor die, two memory channels per controller, two dual-ranked DIMMs per channel, and x4 4 Gb chips.
b) One memory controller per processor die, four memory channels per controller, two dual-ranked DIMMs per channel, and x16 4 Gb chips.

(10 points, 3, 3, 4)

Q9. For the following access stream, estimate the time it takes for the memory request to finish for three scheduling policies: (1) Open-page, (2) Closed-page, and (3) Oracular. The Oracular policy dynamically switches between open and closed policies based on prior knowledge of the access stream. X, X+1, X+2, X+3 map to the same row, and Y, Y+1 map to a different row in the same bank. Access to an open row (row buffer hit) takes 20 ns, access to a closed row if another row is already open (row buffer conflict) takes 60 ns, and access to an empty row buffer (bit lines are precharged already) takes 40 ns.

Request	Arrival Time	Open	Closed	Oracular
X	0 ns			
Y	30 ns			
X+1	100 ns			
X+3	210 ns			
Y+1	250 ns			
X+2	330 ns			

(5 points, 2.5, 2.5)

Q10. A high-performance computing application iterates over contiguous words in a virtual page. The pattern repeats for adjacent virtual pages over time. For such an application, an architect is considering the address mapping scheme at the bank level. Explain the potential advantage and disadvantages of row interleaving over cache block interleaving for the application. Provide a scenario where a high-performance computing application would benefit tremendously from row interleaving.