EECS 470 *Final Exam*

Winter 2025

Name: _____ Uniqname: _____

Sign the honor code:

I have neither given nor received aid on this exam nor observed anyone else doing so.

_____

#	Points
1	**/ 10**
2	**/ 15**
3	**/ 18**
4	**/ 12**
5	**/ 19**
6	**/ 13**
7	**/ 13**
8	**/ 0**
Total	**/100**

NOTES:

- Closed book and one double-sided 8.5x11 page of notes
- Calculators are allowed, but no smart watches, cell phones, earbuds, etc.
- Don't spend too much time on any one problem.
- You have about 120 minutes for this exam.
- There are **10** pages including this one.
- **Be sure to show your work and explain what you've done.**

1) Bubbles [10 points]

1) When doing virtual-to-physical address translations, there is aliasing if the
block offset / page offset / physical page number / tag / index / virtual page number
◯ ◯ ◯ ◯ ◯ ◯
bits overlap with the bits of the
block offset / page offset / physical page number / tag / index / virtual page number
◯ ◯ ◯ ◯ ◯ ◯

This question had other ways to fill out the bubbles and still be correct. Since the two rows are the same, you could make the selection of "index" in either of the rows. In the other row, you could select either "virtual page number" or "physical page number". The bits of both the virtual and physical page number start in the same position, so either would work.

2) When doing **speculative load** scheduling, the processor must include age logic to determine the (◯ **oldest** / ◯ **youngest**) (◯ **load** / ◯ **store**) that is (◯ **older** / ◯ **younger**) than the (◯ **store** / ◯**load**) when determining memory order violations

Common mistake here was to answer what you needed for checking store-to-load forwarding. The question wants to know about recovering from **speculative loads,** which is detected when a store executes, and you need to find the load to roll-back to in order to restart execution.

1) Given a cache access pattern of A,B,C,D,A,B,C… where A,B,C and D are addresses chosen with uniform random probability. What is the approximate probability that the second access of A will hit in the cache, if the cache is a 2048 Byte direct-mapped cache with a block size of 32 Bytes?

25%	**32%**	**77%**	**80%**	**95%**	**100%**
◯	◯	◯	◯	◯	◯

Cache block count is 2048/32 = 64. The odds that A and something map to a different line is 63/64. The odds that B, C, and D all map to a different line is 63/64*63/64*63/64=95.4% Could also calculate the odds as 1 - (P(A=B) + P(A=C) + P(A=D) - P(B=C) - P(B=D) - P(C=D)). That is 1-(1/64+1/64+1/64-3*(1/64)^2)=95.4%.

3) Which is most capable of dealing with jumps that occur after functions finish executing?

BHT	**PHT**	**RAS**	**BTB**	**BHR**	**TLB**
◯	◯	◯	◯	◯	◯

It is called a return address stack for a reason. When functions finish, they call return.

4) ◯ **True** **or** ◯ **False**
In a vector style processor a single instruction operates across all the values in the entire register file.

Vector machines perform operations across all elements of the vector (one vector register), not across all the vector registers.

2) Cache Tradeoffs [15 points]

You have landed a job at Jonah's Brothers' Chips, a new semiconductor company specializing in server-oriented computing. Specifically, your job is on the cache team and recently you have noticed that the cache hit rate is not as high as it should be.

a) Your manager Nick has the great idea of reducing your cache miss rate by simply making the on-chip cache larger. What are two advantages and two disadvantages of this approach? **[4 points]**

Advantages: more storage space (so fewer conflict and capacity misses), fewer bits needed for tags, relatively easy, less pressure on memory bus

- Note: "lower miss rate" is counted as one advantage. Capacity vs conflict misses are not different enough to count for different advantages

Disadvantages: slower access time, more power, more area, doesn't help if the majority of misses are compulsory misses, potentially introduce aliasing (with virtual memory)

b) Your coworker, Joe, disagrees with your manager and suggests keeping the cache size the same, but increasing the associativity. What are one pro and two cons of this approach? **[3 points]**

Advantages: fewer conflict misses

Disadvantages: higher power, slower access time, larger tag array (more bits go to tag since there are fewer sets)), increased overhead for replacement logic, more complicated replacement logic

c) Meanwhile, after the collapse of Silicon Valley Bank in 2023, your financial advisor Kevin believes that keeping all your cache data in one bank is too risky, so you suggest splitting the cache into multiple banks. What is the main benefit and main disadvantage that banking the cache gives? **[4 points]**

Advantages: Multiple reads and writes per cycle without needing extra ports on each SRAM

Disadvantages: Potential bank conflicts if you are trying to read/write two different lines that go to the same bank, more logic for bank management, potentially more concurrent memory transactions since evictions can come from more sources

- Any of these answers would've worked as a disadvantage, but for advantage the only correct answer was multiple reads/writes per cycle without extra ports on each bank

d) Finally, your therapist Frankie suggests increasing the cache block size. You bring this idea to your manager, and they say the benefits of this will be application dependent. How does increasing the block size change the performance in a program that does: **[4 Points]**

i) Matrix multiplication using matrices that are stored as contiguous bits in memory?

Helps performance! You are very likely to access all the data in the larger block, so you incur fewer misses.

ii) Graph algorithms where the algorithm traverses a graph based on the data in the current node being processed?

Hurts performance. Since you are not likely to access all the data in the block, you end up fetching more data that doesn't get used.

3) Coherence [18 points]

a) What condition is necessary for a memory system to be coherent? **[2 points]**

There has to be total order among all stores to **any** given address

b) How is this different from consistency? **[2 points]**

There has to be total order among all stores to **all** addresses in the address space

c) Recall that the "owned" (O) state in a coherence protocol is when a cache line is shared and dirty. This state allows for an optimization where a dirty cache line does not need to be written back to memory until the core that modified it evicts the line. Complete the table below showing which states can transition into the O state, and to which states a processor can transition from O. **[6 points]**

	Can transition **to** O	Can transition **from** O
M	X	X
S	◯	◯
I	◯	X

d) Now consider the "exclusive" (E) state. This state corresponds to a cache line that is shared, and the core holding it is the only one with that cache line. Complete the table showing which states can transition into the E state, and to which states a processor can transition from E. **[6 points]**

	Can transition **to** E	Can transition **from** E
M	◯	X
S	◯	X
I	X	X

e) Why would you want to add the Exclusive state to the standard MSI protocol? **[2 points]**

The E state is used when a processor is the only one in the system with that particular cache line but the line remains clean.
This state is beneficial because it reduces bus transactions needed in order to write to the block later (ie silent E -> M transition)

4) Bits of (Pre-)history [12 points]

In the year 40,000 B.C., Grog the Neanderthal won MICRO best paper for his contributions to branch predictor architecture. Because microprocessor design was still in its infancy during the Upper Paleolithic Age, Grog's design was a simple branch history table with 1 bit of history. That is, the prediction is always the last recorded outcome of the branch.

a) Memory was quite limited in 40,000 B.C. If the BHT size were limited to 2KiB, how many PC bits should be used to index the BHT? **[2 points]**

2KiB = 2^11 B = 2^14 bits – each entry is a singular bit, so 14 PC bits used to index.

b) Overcome with jealousy, a rival neanderthal clan decided to write programs with **pathological behavior** in order to decrease the accuracy of Grog's predictor. Consider the following program:

```
for (int i = 0; i < 100000; ++i) {
      if (i % 3 == 0 || i % 4 == 0) { /* Branch A: decision */
            // important mammoth hunting calculations
      } else {
            // berry gathering routine
      }
} /* Branch B: loop backedge */
```

What is the *steady-state accuracy* of Grog's predictor on this program, you may assume that Branch A and Branch B do not alias? **[5 points]**

In the steady state, branch A's outcome pattern repeats every 12 iterations:

N T T **N** N **T N T N** N **T** T

In the pattern, we have 8 transitions from T→N or N→T, each of which causes a misprediction with one bit of history. Outcomes which cause a misprediction are bolded above.

In the steady state, branch B is always predicted accurately (it is taken over and over again).

Therefore, our accuracy is approximately 4/12 + 12/12 = 16/24 ≈ 66.67%

c) Ever innovative, Grog decides to develop a new **local history**-based predictor in order to improve performance. Grog's new predictor introduces a pattern history table (still with a single bit of history) – the PHT is a second level structure indexed by the BHT output for a given branch. How many bits of history would Grog need to have in each BHT entry in order to achieve 100% *steady-state accuracy* for the above program? **[5 points]**

Consider the same pattern as in part b).

With 4 bits of history, we have the following issue:

- Starting on iteration 0, we get N T T N, and should predict N
- Starting on iteration 9, we get N T T N, and should predict T

Therefore it is impossible to get 100% accuracy.

With 5 bits of history, there is no such conflict. So Grog needs 5 bits of history in order to achieve 100% steady-state accuracy

5) Load-Store Queues [19 points]

a) Why is every load potentially dependent on every prior store? **[3 points]**

The store could write to the same address from which the load reads

b) When can a store leave the store queue? Why? **[4 points]**

At retirement, because you don't want to speculatively write a store to the cache.

c) Your company, Stores "Я" Us, wants to break the rule from part (b) and speculatively send stores to the data cache. Even though you tell them this is a bad idea, your manager pushes the idea ahead anyway.

i) What extra data would you need to save/checkpoint before sending the store to the data cache? **[4 points]**

The previous data at the address you're storing to so you can recover this data

ii) Assuming your data cache is limited to one read and write port, is it possible to recover from a branch mispredict in one cycle on every misprediction? Why? **[4 points]**

No, because you could have multiple stores you need to correct, and you can only do one write to the cache per cycle.

iii) Does this change how your cache handles forwarding from stores to loads? Explain why or why not. **[4 points]**

Yes, because in the default scheme when you only send stores to the cache at retirement, all stores in the cache are older than any load that is outstanding. But, now you can have interleaving of stores and loads in the cache, so the dcache needs to check if a store is older than an outstanding load. You could also allow stores to still go to the memory before retirement, but not allow them if there is an older load that doesn't know its address. If it helps, think of this new case as "backwarding" the data.

Bonus: Some students came up with the solution presented in the InvisiFence paper. In that, when you start speculating on a store, you copy all dirty lines back to memory. Now if you put the store into the cache, the main memory will have the old value. When you need to recover, you can do so in a single cycle by just invalidating all the lines in the cache. There are some corner cases that need to be handled (can't evict a dirty line while speculating for example), but this would allow the answer to ii) to be yes, you can recover in a single cycle and iii) Similar answer, you need to check age, and if so you need to grab the old value from memory, as it is no longer in the part (i) answer..

6) Prefetching [13 points]

a) Consider a stream-buffer based data prefetcher. For each of the following statements, indicate whether it is true or false. Briefly explain why.

i) As stream buffer depth increases, so does data cache pollution. **[3 Points]**

False. The data stays in the stream buffer and doesn't move to the cache until it is confirmed to be needed.

ii) Stream buffers perform poorly for linked-list traversals. **[3 Points]**

True. Linked lists do poorly with prefetching in general since they jump around in memory.

iii) A multi-way stream buffer will improve the performance of traversing a large array. **[3 Points]**

Could argue either way as long as valid reasoning is provided.
False: If you have a lot of ways, then your utilization of the multiple ways will be poor, as you only have one streamr. If the total stream buffer size is limited (same number of total entries), then adding multiple ways makes it perform poorly compared to a single ways.
True: Stream buffers do well with traversing large arrays, as they have a unit stride..
In summary: the multi-way part is bad, but the traversing a large array pattern part is good.

b) List one advantage and one limitation of runahead prefetchers over conventional prefetchers (like stream buffers or global history buffers). **[4 Points]**

Advantages: No need for extra memory structures to buffer outside the cache, warms up branch predictor, fetches instructions that will be needed, can capture more complex patterns by actually executing them, can remove dependent instructions from the ROB to allow more instructions not dependent on the L2 miss to be eligible to execute (assumes a poision bit implementation compared to a checkpoint implementation).. (Would not accept vague statements like "it is more accurate" or "increased performance" without valid reasoning.)

Limitations: Wasted energy of running the core, needs to discard the entire ROB/RS and re-execute, whereas a prefetcher would just block the code and allow the pipeline to pick back up part way through.

7) Virtual Memory [13 Points]

a) What is the "synonym problem" when it comes to virtual memory? **[4 points]**

A **synonym** is when 2 or more different virtual addresses map to the same physical address

The **synonym problem** is that when we support synonyms, if we use a virtual address to index a cache, then we can have the same physical address potentially in more than one cache line. This akes it so updates to one line are not seen by the other.

b) The company "Ozone Chips" is investigating L1 cache architectures. Their system has 32-bit virtual addresses, 24-bit physical addresses, and 4 KB page sizes. Assuming you must stay with the specified virtual memory system and must use a virtually-indexed physically-tagged cache, label the following options as "viable" or "non-viable". Briefly explain why.

To avoid the synonym problem, we need to ensure that cache indexing uses only bits from the page offset of the address. In set associative caches, this is accomplished by making page size * associativity >= cache size

i) 4-way set associative 8 KB cache **[3 Points]**

Viable
4 KB (page size) * 4 (associativity) = 16 KB >= 8 KB (cache size)

ii) Direct-mapped 4 KB cache with a 512 B fully-associative victim cache **[3 Points]**

Viable

For the direct-mapped cache:
4 KB (page size) * 1 (associativity) = 4 KB >= 4 KB (cache size)

Would accept either: (1) The victim cache is fully-associative, which means 0 index bits, so synonyms are not possible. Or (2) assumed the vitim cache is not virtually indexed and will get the PPN from the TLB before access and therefore cannot have aliases. Or (3) victim cache smaller than page size.

iii) 2-way skew associative 4 KB cache **[3 Points]**

Not Viable

A skew associative cache uses the higher bits of address (past the block offset) as part of its indexing function.

8) Bonus [0 Points]

a) Did you use ChatGPT to write Verilog during the course, and in what ways was it helpful/hurtful? (you will get full points regardless of what you answer, we are just curious)

b) What are your recommendations for future students in this course?