You can find the past exams here: https://cs61c.org/su22/resources/exams/

When posting questions, please reference the semester, exam, and question in this format so it's easier for students and staff to search for similar questions:

**Semester-Exam-Question Number**

For example: **SP20-Final-Q1**, or **SU20-MT2-Q3**

Summer 2020 midterm walkthrough

Fall 2020 midterm Bit Manipulations Walkthrough

Fall 2020 midterm 1 Slip Walkthrough

Added 2022/08/10:

Extremely rough draft SP20 final solutions

Extremely rough draft FA20 quest solutions

Extremely rough draft FA20 midterm solutions

For question (a), why isn't reg 0 -> AND -> REg 1 a legit CL path? that gives clk + AND = 4 + 9 = 13ns

Follow up:

how is the longest CL 9 here? Shouldn't that be and + not + or?
Peyrin Kao  STAFF  2mth  #1311ea

That's input-->input, not output-->input.
...

Anonymous Aardvark  2mth  #1311dd

I was confused about this explanation of the minimum case. How can we let thread 1 read a and stop, let thread 2 do the work, and thread 1 will just finish? I wonder won't they both need to do all the works?

However, I also think that the minimum value is 5. My reason is that thread1 and thread2 are just going to do the same work at the same time. They can do so because in the n-stage pipeline, both of them can read before they write back. Thus, the work is just like done by one thread. Is my reasoning correct?
...

Peyrin Kao  STAFF  2mth  #1311ed

"Finishing" does mean doing all the thread's work.
...

Anonymous Aardvark  2mth  #1311dc

FA20-Final-Qgeneral

What does "active page table", "inactive page table", and "PTBR" mean? Are they in scope?
...

Peyrin Kao  STAFF  2mth  #1311eb

Active page table = at least one of the entries is valid, invalid page table = all the entries are invalid, PTBR not in scope.
...

Anonymous Aardvark  2mth  #1311db

FA20-Final-Q4 PartB

I was wondering about this problem. What if the opcode is wrong? Why can't we get an exception error in the first Instruction Fetch phase?
...

Peyrin Kao  STAFF  2mth  #1311ec

It sounds like you want the value from the ALU (the next PC being jumped/branched to) rather than the the values read from the regfile.
...
I'm not sure about how #2, #3, #4, and #5 are found out. Why would 0 and 4 have fast small reads but not 2, 3, 5? Why would 1 and 5 have fast small reads but not 2, 3, 4?

Edit: Please Ignore the automatic hyperlinking.

---

For RAID 1, we can write to all disks simultaneously, and so can finish in one step. I would note that I dispute the answer to this question, though, and would have answered 0, 1, 2 (we always write bytes, so we don't need to read, then write to parity like with 3/4/5)

For 2, I think the question is ambiguous between memory overhead and time overhead. I'm not really sure why 2 and 3 are considered low overhead, but 0 is low overhead because there is basically no memory or runtime overhead, because we have no redundancy.

---

What is \xd7 in the solution?

---

It's \times in ASCII (rendering doc was weird).

---

Why do we need the output from regfile for lui instruction? I thought for lui instruction we would simply set bsel to select the output from immgen and set the ALU output to be B. After that we would write the ALU output to rd. I don't see where output from regfile is used in this process.

---

You need to use the regfile to write the result.

---

Why is this sometimes incorrect?
Anonymous Gerbil 2mth #1311ae
I was also confused on this after you exit the for loop will dot_product be the one set equal to zero or a private one? Since we effectively finish our forking after the

#pragma omp parallel for

{code}

I think that the dot_product after the for loop is zero since were not doing TLP anymore. And we never combine each private dot_product. Is this reasoning correct?

...

Anonymous Hornet 2mth #1311af
Yeah, I think the issue is with not combining the private values. Put another way, each thread makes its own copy of dot_product. Like a function receiving a copy of a variable, it doesn't modify the original variable. Not 100% sure on this though.

...

Peyrin Kao STAFF 2mth #1311bd
Every thread has its own copy of dot_product. The only way to get the correct result is if one thread performs all the calculation.

...

Anonymous Salmon 2mth #1311cc
So without the private keyword, this would be correct?

...

Peyrin Kao STAFF 2mth #1311ce
Replying to Anonymous Salmon
I think so.

...

Anonymous Salmon 2mth #1311cf
Replying to Peyrin Kao
What is the difference between private and reduction?

...

Peyrin Kao STAFF 2mth #1311da
Replying to Anonymous Salmon
Private = each thread has its own copy of the variable

Reduction = each thread has its own copy, and all the copies get combined into one final result at the end

...

Anonymous Squid 2mth #1311ab
Resolved
su20-Final-1b: How do you find the number of each level's PT? I know how to find PTE, but now sure how to find PT. Thanks!
The VPN originally consisted of 12 bits, but we've now formed a two level page table scheme. The problem specifies the bits are evenly split between both levels (i.e., the first and second level page tables each have 6 VPN bits). So how many values can be represented with 6 bits? $2^6 = 64$. This is the number of PTEs for each page table.

As there is only one L1 page table, the first answer is 64. If you know that the L1 page table has 64 PTEs, and that each of these PTEs is pointing to a page table of the same size, there must be a total of $64 \times 64 = 4096$ PTEs at the second level.

I wanted to ask what are the differences between a section and a segment? Also for this problem I assume we are using the memory model code -> data -> heap -> stack

I'm also assuming that the addresses are supposed to give away that sean should be in code, but what about jenny where would 0x1620 be?

Sections and segments are the same I believe.

Why is the code for Q2.B considered slower? We aren't given information about the cache beyond the size of its cache lines (blocks). I'm having trouble seeing how we can conclude its slower.

Also, is it true that the reduction clause makes the #pragma omp critical directive redundant (i.e., reduction handles it for us)? Thanks!

It's very likely that there's false sharing, since each iteration of the loop is accessing adjacent elements. reduction handles the critical section for you.
Anonymous Hornet 2mth #1311e

Thank you for the fast reply! Does that mean the directive `#pragma omp parallel for` results in something like the following for each thread (interleaving)?

```c
for (int i = thread_id; i < n; i += thread_count) {
    // some operation with the array at index i
}
```

Peyrin Kao STAFF 2mth #1311f

Replying to Anonymous Hornet

OpenMP makes no promises about interleaving vs. chunking, but either way it's possible that some false sharing occurs.

Anonymous Squid 2mth #1311a Resolved

For Su20-MT2-2cii, why we choose C instead of A? I thought the control signal of RegWEn is determined by RegWEN. Thanks!

Peyrin Kao STAFF 2mth #1311b

We want to write to register x17 whenever a store instruction happens. Usually, RegWEn is 0 during a store instruction, so we have to force RegWEn to be 1 whenever we're writing to memory (i.e. when MemRW is 1).