You are viewing this thread in readonly mode.

[Final] Past Exams - 2022 #1043

Jero Wang STAFF
12 months ago in **Exam – Final**

1,256
VIEWS

You can find the past exams here: https://cs61c.org/fa23/resources/exams/. Please check the linked past Piazza/Ed Q&A PDFs first before asking here. Many of the questions are already answered in those! Video walkthroughs (if available), are also linked on that page!

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: **SP22-Final-Q1**, or **SU22-MT-Q3**

Anonymous Grasshopper 12mth #1043bed ✓ Resolved

SU22-FINAL-q7.4

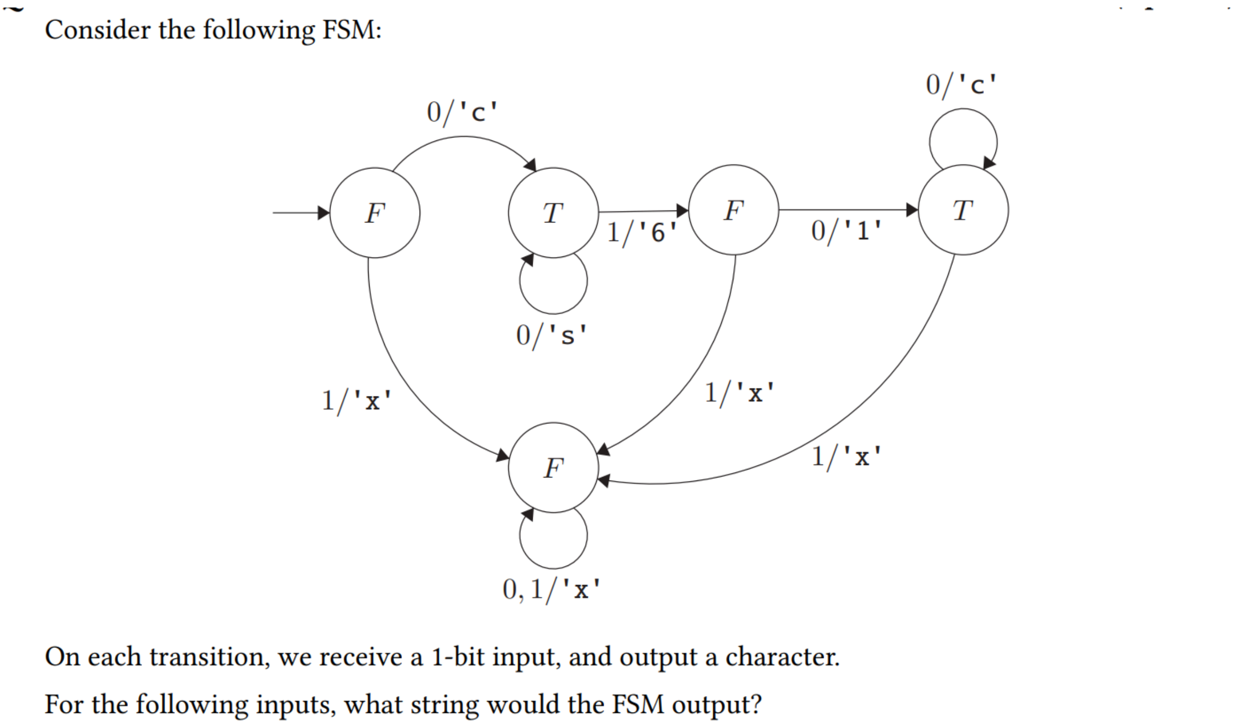
I think there's a typo: solution says 12 bits = 4 hex bytes (should be 3)

Jero Wang STAFF 12mth #1043cad

Yep, thanks for the catch, noted on our end!

Jiries Kaileh 12mth #1043bbf ✓ Resolved

su22-final-q6


Consider the following FSM:

On each transition, we receive a 1-bit input, and output a character.

For the following inputs, what string would the FSM output?

kinda paranoid, but if we put quotation marks around the answers (like "css61c" instead of css61c) would we get docked? technically the quotation marks are not output, but the question itself asks what **string** was output. I assume either answer was fine tho.

♡ ···

Jero Wang STAFF 12mth #1043bdc

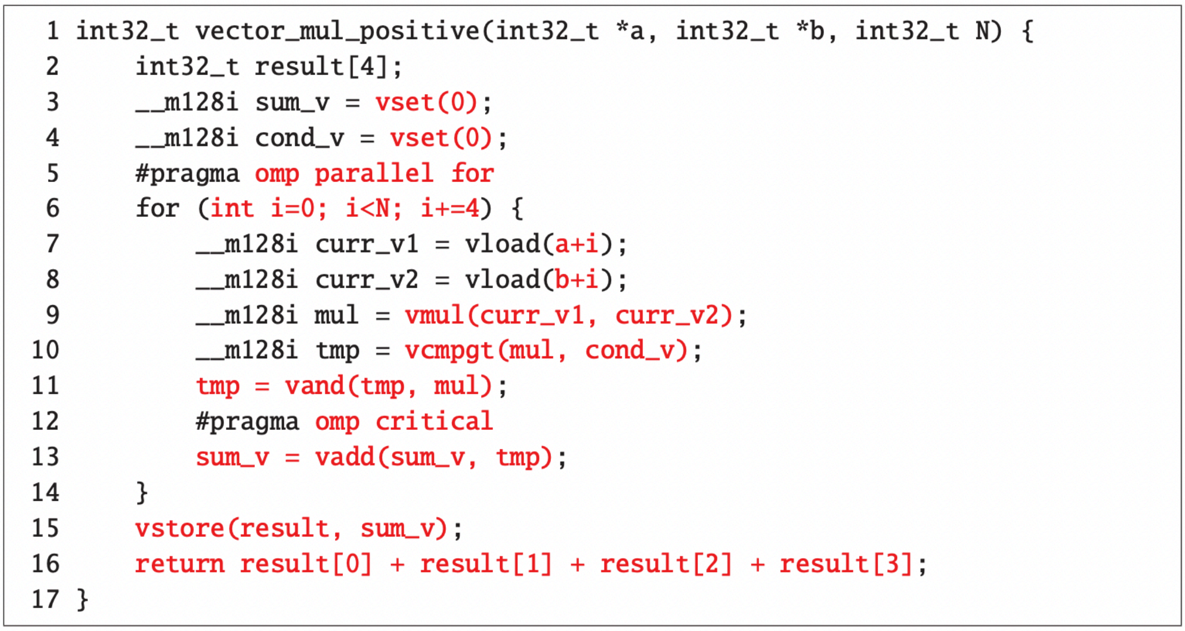
I think either would be fine here, since " is not a valid output of the FSM.

♡ 2 ···

Anonymous Manatee 12mth #1043bad ✓ Resolved

SU22-Final-Q8

For line 15, why are we able to use the result variable as the first parameter of vstore is it's indicated in the spec that vstore's first parameter must be of type __m128i*?


```
 1 int32_t vector_mul_positive(int32_t *a, int32_t *b, int32_t N) {
 2     int32_t result[4];
 3     __m128i sum_v = vset(0);
 4     __m128i cond_v = vset(0);
 5     #pragma omp parallel for
 6     for (int i=0; i<N; i+=4) {
 7         __m128i curr_v1 = vload(a+i);
 8         __m128i curr_v2 = vload(b+i);
 9         __m128i mul = vmul(curr_v1, curr_v2);
10         __m128i tmp = vcmpgt(mul, cond_v);
11         tmp = vand(tmp, mul);
12         #pragma omp critical
13         sum_v = vadd(sum_v, tmp);
14     }
15     vstore(result, sum_v);
16     return result[0] + result[1] + result[2] + result[3];
17 }
```

♡ 1 ···

J

Jero Wang STAFF 12mth #1043bfb

Pointers can be cast implicitly in C.

♡ ···

Anonymous Mantis 12mth #1043bac ✓ Resolved

SU22-Final_Q5.5

Solution:

Earliest: 6 ns. The values of A-H are computed at 1 ns. Then after the 5 ns delay of Box 2, the output changes.
Latest: 8 ns. The inputs change at 0 ns. Then after the 3 ns delay of Box 1, the bottom input to Box II changes. Then after the 5 ns delay of Box 2, the output changes.

I'm very confused about the explanation, could someone elaborate?

♡ 1 ···

J

Jero Wang STAFF 12mth #1043bfd

This question is really asking what the shortest and longest CL paths are - in this case, the shortest path comes from the A-H labels and goes to the output, whereas the longest goes from the input pins to the output. The shortest CL path is the first timestep where the output can change.

♡ ···

Anonymous Mantis 12mth #1043aff ✓ Resolved

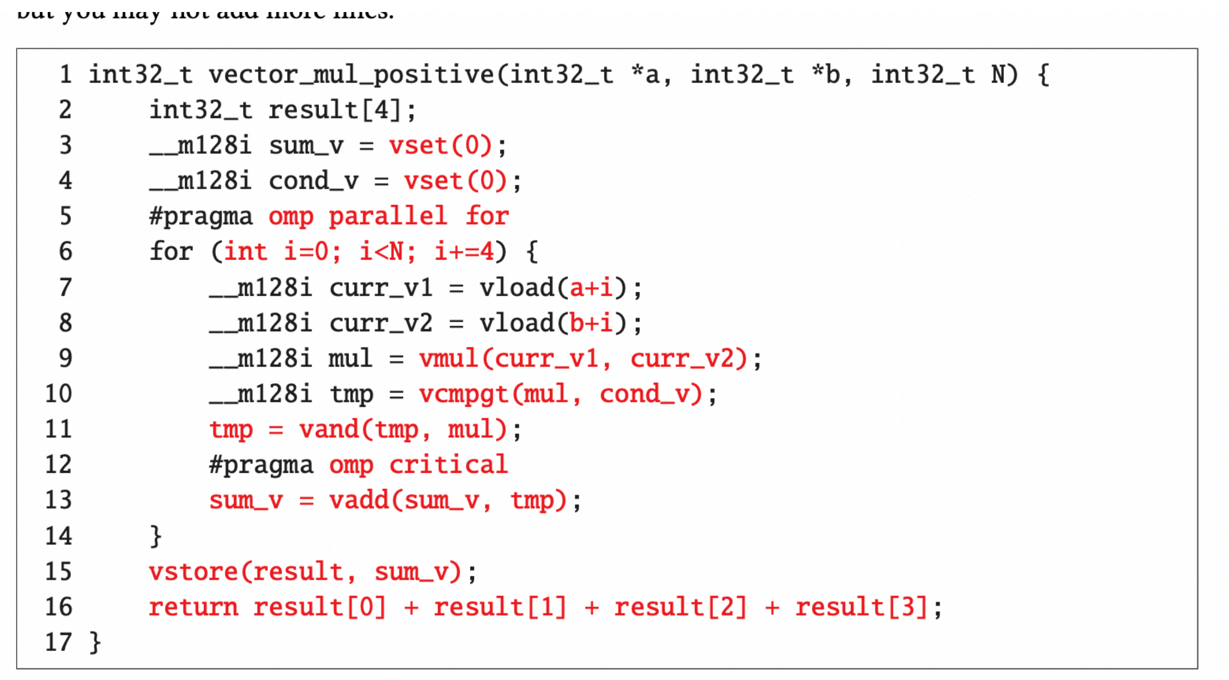
SU22-Final-Q3.4

Could someone explain why the instruction itself is held in ra? I understand that were have to do 3(ra) in order to get to the funct 7 of the instruction, however, I don't get why the full instruction is held in ra on the first place?

♡ ···

Anonymous Marten 12mth #1043afa ✓ Resolved

result should be casted to __m128i pointer?



but you may not add more lines.

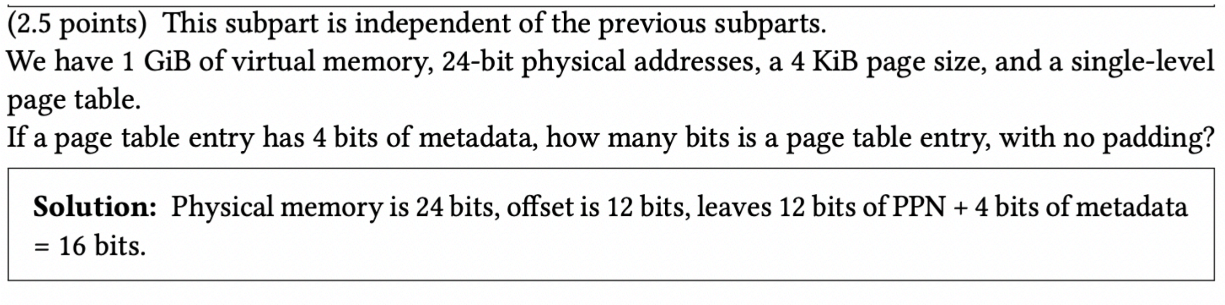
```
int32_t vector_mul_positive(int32_t *a, int32_t *b, int32_t N) {
    int32_t result[4];
    __m128i sum_v = vset(0);
    __m128i cond_v = vset(0);
    #pragma omp parallel for
    for (int i=0; i<N; i+=4) {
        __m128i curr_v1 = vload(a+i);
        __m128i curr_v2 = vload(b+i);
        __m128i mul = vmul(curr_v1, curr_v2);
        __m128i tmp = vcmpgt(mul, cond_v);
        tmp = vand(tmp, mul);
        #pragma omp critical
        sum_v = vadd(sum_v, tmp);
    }
    vstore(result, sum_v);
    return result[0] + result[1] + result[2] + result[3];
}
```

Jero Wang STAFF 12mth #1043bdd

Casts are implicit and not required for SIMD instructions.

Anonymous Marten 12mth #1043aef ✓ Resolved

what abt dirty, valid, and protected bits.



(2.5 points) This subpart is independent of the previous subparts.
We have 1 GiB of virtual memory, 24-bit physical addresses, a 4 KiB page size, and a single-level page table.
If a page table entry has 4 bits of metadata, how many bits is a page table entry, with no padding?

Solution: Physical memory is 24 bits, offset is 12 bits, leaves 12 bits of PPN + 4 bits of metadata = 16 bits.

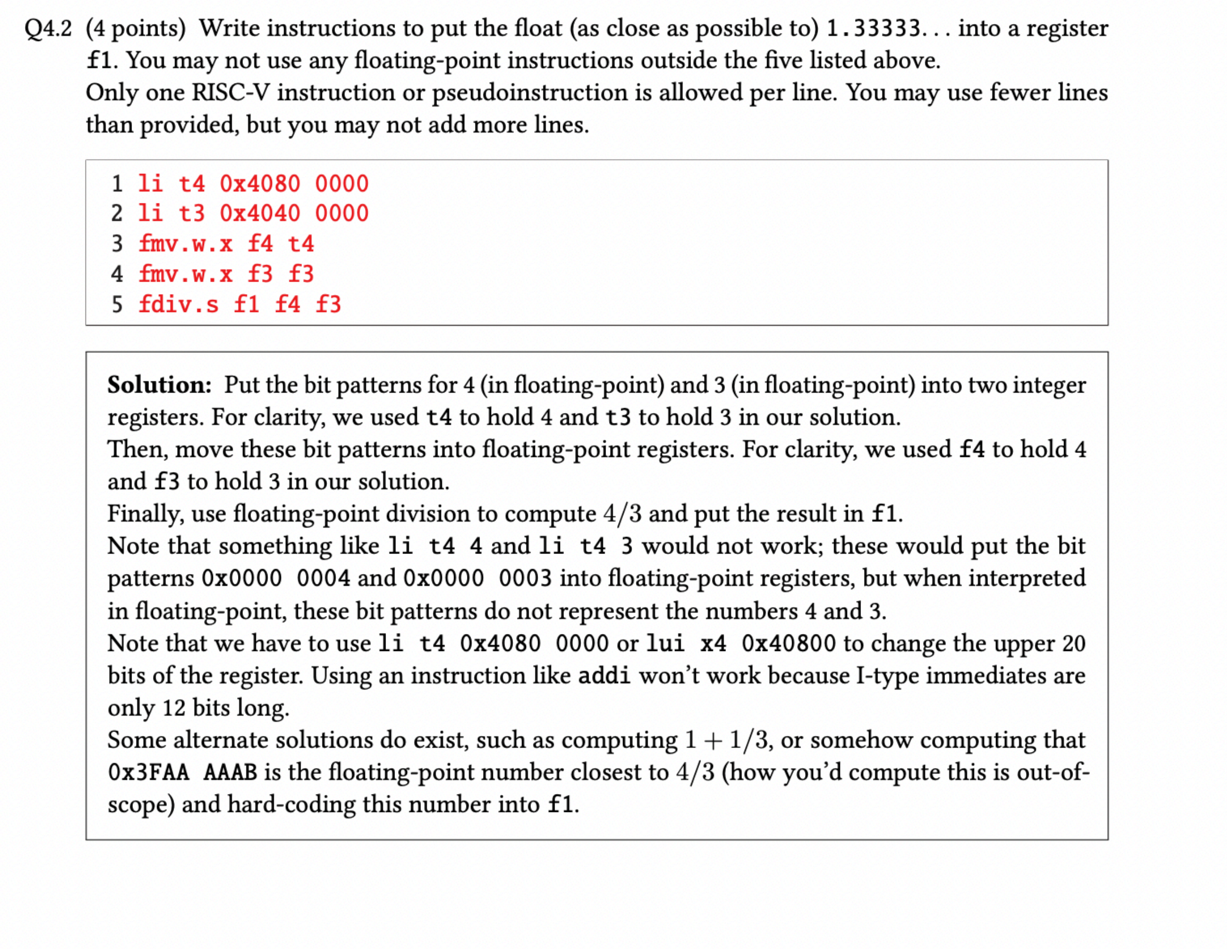
Andy Chen STAFF 12mth #1043afc

Dirty valid and protected bits are considered metadata bits so they would be included in the 4 bits of metadata.

Anonymous Manatee 12mth #1043aee ✓ Resolved

SU22-Final-4.2

Could we have just used the integers 4 and 3 directly instead of its hex translation to input to t4 and t3 respectively?



Q4.2 (4 points) Write instructions to put the float (as close as possible to) `1.33333...` into a register `f1`. You may not use any floating-point instructions outside the five listed above.
Only one RISC-V instruction or pseudoinstruction is allowed per line. You may use fewer lines than provided, but you may not add more lines.

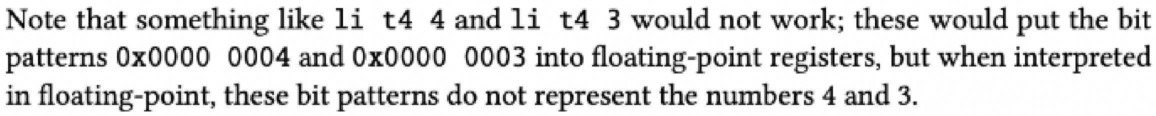
```
li t4 0x4080 0000
li t3 0x4040 0000
fmv.w.x f4 t4
fmv.w.x f3 f3
fdiv.s f1 f4 f3
```

Solution: Put the bit patterns for 4 (in floating-point) and 3 (in floating-point) into two integer registers. For clarity, we used `t4` to hold 4 and `t3` to hold 3 in our solution.
Then, move these bit patterns into floating-point registers. For clarity, we used `f4` to hold 4 and `f3` to hold 3 in our solution.
Finally, use floating-point division to compute $4/3$ and put the result in `f1`.
Note that something like `li t4 4` and `li t4 3` would not work; these would put the bit patterns `0x0000 0004` and `0x0000 0003` into floating-point registers, but when interpreted in floating-point, these bit patterns do not represent the numbers 4 and 3.
Note that we have to use `li t4 0x4080 0000` or `lui x4 0x40800` to change the upper 20 bits of the register. Using an instruction like `addi` won't work because I-type immediates are only 12 bits long.
Some alternate solutions do exist, such as computing $1 + 1/3$, or somehow computing that `0x3FAA AAAB` is the floating-point number closest to $4/3$ (how you'd compute this is out-of-scope) and hard-coding this number into `f1`.

♡ 1 ···

A Andy Chen STAFF 12mth #1043afb

We wouldn't be able to, see this section of the explanation:



Note that something like `li t4 4` and `li t4 3` would not work; these would put the bit patterns `0x0000 0004` and `0x0000 0003` into floating-point registers, but when interpreted in floating-point, these bit patterns do not represent the numbers 4 and 3.

♡ ···

Anonymous Mongoose 12mth #1043aed ✓ Resolved

SU22-FINAL-Q2

Why are we allowed to directly strcpy over contents with strcpy(new_file->data.contents, contents);? I thought we would have to malloc space for file_item_data first before we can access new_file->data.contents

♡ ···

Anonymous Turtle 12mth #1043baa

I have the same question !

♡ ···

N Noah Yin STAFF 12mth #1043bfc

Because we are using Calloc, it automatically allocates enough space for the whole file_item struct and sets it to zero, including the file_item_data union data. So, we are allowed to directly strcpy over contents.

♡ ···

Anonymous Dove 12mth #1043aeb ✓ Resolved

SP22-Final-Q2.2

Can you show step-by-step how (!B)(!(CD)) + CD simplifies to !B + CD using the absorption law? tysm :)

♡ 1 ···

Sonika Vuyyuru STAFF 12mth #1043baf

Starting with the expression given in the problem: !((A + !A)B + (CD)) + CD

Substituting in (A + !A) = 1: !(B + (CD)) + CD

DeMorgan's law [!(A + B) = (!A * !B)] gets us: (!B)(!(CD)) + CD

Finally, law of absorption says [A*(A+B) = A or A+(A*B) = A] , which gets us: !B + CD.

The idea with law of absorption in the last simplification is that it must be true that EITHER !(CD) or (CD), but both cannot be true at the same time. We can use this idea to simplify our expression. If !(CD) is true, then the expression simplifies to !B. If (CD) is true, then the expression simplifies to 1. Essentially, the (!(CD)) in the first term did not change the truth value of !B, which is the essence of the absorption law.

♡ 1 ···

Anonymous Dove 12mth #1043bbc

thank you SO much! This was SUPER useful!!!

♡ ···

Anonymous Boar 12mth #1043add ✓ Resolved

Hello, over the past exams I found that the solutions have been counting valid bits inside of metadata for page table entry, but also recall that they were presented separately on other course materials. When computing the size of a page table entry, should we then follow the format of:

num_valid_bits + num_metadata_bits + PPN_bits

or

num_metadata_bits + PPN_bits

Thanks

♡ 3 ···

Jero Wang STAFF 12mth #1043bce

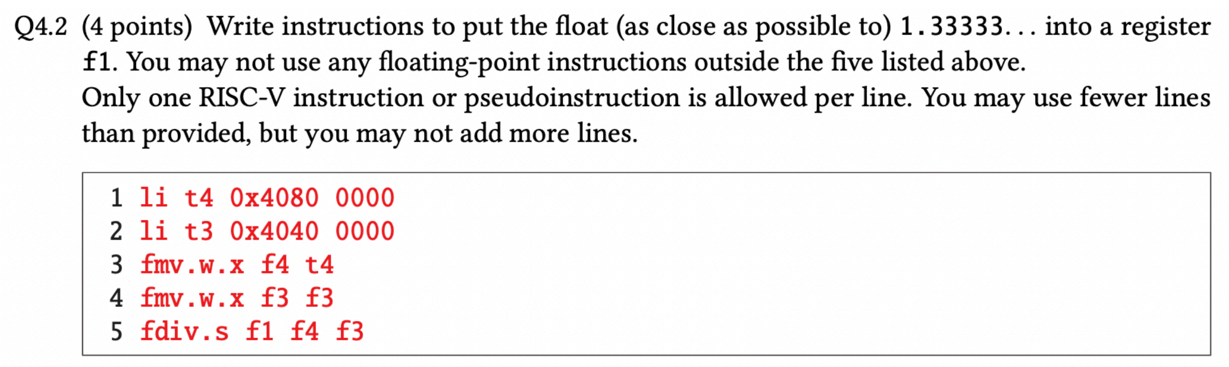
The question will specify if the metadata bits include the valid bit or not.

♡ ···

Anonymous Sparrow 12mth #1043adb ✓ Resolved

SU22-FINAL-Q4.2

I Believe there is an error in the solution to question 4.2. Shouldn't line 4 be: **fmv.w.x f3 t3** since we're converting the f3 integer value to float and storing it in t3?



Q4.2 (4 points) Write instructions to put the float (as close as possible to) 1.33333... into a register f1. You may not use any floating-point instructions outside the five listed above.
Only one RISC-V instruction or pseudoinstruction is allowed per line. You may use fewer lines than provided, but you may not add more lines.

```
li t4 0x4080 0000
li t3 0x4040 0000
fmv.w.x f4 t4
fmv.w.x f3 f3
fdiv.s f1 f4 f3
```

Andy Chen STAFF 12mth #1043afd

Yeah that looks like a typo, good catch!

Anonymous Ant 12mth #1043acf ✓ Resolved

SP22-FINAL-Q4.4

I understand why IEEE-754 standard single precision numbers have to be an odd integer multiplied or divided by a power of 2. However, how exactly are we supposed to figure out whether or not a number meets that criteria? For example, how can we conclude that 0.8 does not fulfill that criteria?

1

Andrew Liu STAFF 12mth #1043caf

You can conclude this by looking at the fraction representation of 0.8. 0.8 is equal to 4/5, or equivalently 4 * 5^-1, and 5^-1 is not a power of 2.

Anonymous Newt 12mth #1043acc ✓ Resolved

SU22-Final-Q2

Is this code acceptable to set name?

```
file->name = calloc(strlen(name) + 1, sizeof(char));
strcpy(file->name, name);
```

Anonymous Eagle 12mth #1043aec

same question

Noah Yin STAFF 12mth #1043cab

Yes, that should work as you allocate enough space for name and the null terminator to be copied over using strcpy.

Anonymous Newt 12mth #1043aca ✓ Resolved

SU22-Final-Q2

I had a question regarding the union and the children array. Given the way the children array is initialized, will an array of 16 elements of file_item pointers be created within the union, but the variable `children` will be equal to a pointer to the first element of the array, and the memory block for `children` that points to the array will not be part of the union, right?

♡ 1

Jero Wang STAFF 12mth #1043bfe

`children` is an array of 16 `file_item *`s, so there are 16 `file_item *`s stored within the union, and `children` is technically a `file_item **`. The memory that each element in the `children` array is not part of the union.

Anonymous Hornet 12mth #1043abb ✓ Resolved

Su22-Final-Q3.3

Would li t0 68 for the first line and jalr x0 t0 -32 for the second line be a valid alternate solution for this question? Thanks!

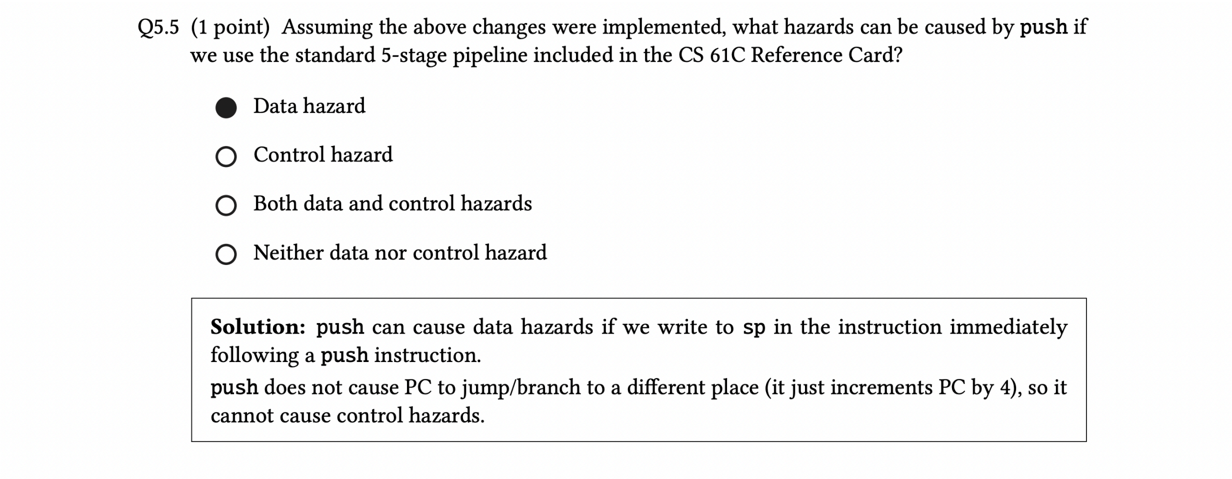
Andrew Liu STAFF 12mth #1043cbe

No, since this jumps the the absolute address `0x24` which might not be where `loop` is.

Anonymous Parrot 12mth #1043aba ✓ Resolved

SP22-Final-Q5.5

How does push cause a data hazard here? If we have data forwarding, then the ALU output of the push instruction will already have calculated (sp - 4). We just forward this ALU output as the input to the next instruction's EX stage and this is solved?



Q5.5 (1 point) Assuming the above changes were implemented, what hazards can be caused by **push** if we use the standard 5-stage pipeline included in the CS 61C Reference Card?

● Data hazard

○ Control hazard

○ Both data and control hazards

○ Neither data nor control hazard

Solution: **push** can cause data hazards if we write to **sp** in the instruction immediately following a **push** instruction.
push does not cause PC to jump/branch to a different place (it just increments PC by 4), so it cannot cause control hazards.

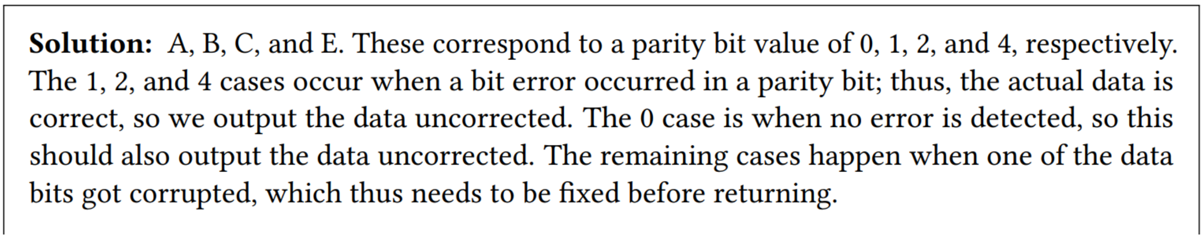
Jero Wang STAFF 12mth #1043bde

The pipeline on the refcard does not have forwarding, so it would be a data hazard.

Anonymous Ram 12mth #1043aab ✓ Resolved

SU22-Final-Q5.4

I'm having a bit of trouble wrapping my head around the explanation for this problem. How exactly do the 001, 010, and 100 bits each correspond to a bit error in the parity bit? Also, how exactly does the error occurring in the parity bit guarantee that the actual data is correct?



Solution: A, B, C, and E. These correspond to a parity bit value of 0, 1, 2, and 4, respectively. The 1, 2, and 4 cases occur when a bit error occurred in a parity bit; thus, the actual data is correct, so we output the data uncorrected. The 0 case is when no error is detected, so this should also output the data uncorrected. The remaining cases happen when one of the data bits got corrupted, which thus needs to be fixed before returning.

1

Anonymous Toad 12mth #1043abe ENDORSED

After we take xor of all values of index_i * bit_i for i from 1... 7, we obtain a single integer, which is the index of the bit flip error. The result can be from 0...7, corresponding to each of the boxes from A...G. If result is 0, that means no error occured --> box A. If the result is 1,2, or 4, these are the locations of the parity bits, so that means the bit flip error occured in these indicies. All of these cases don't affect our final result, we still output the original D1 - D4, thus they have the same value

Anonymous Ram 12mth #1043ff ✓ Resolved

SU22-Final-Q3.4

How did they get the 3 in 3(ra)? I also don't entirely get how that corresponds to the 64th bit.

I'm also a little confused conceptually how modifying ra after line 7 changes our add into a sub instruction that we actually execute. Does calling ret somehow result in the newly modified instruction to execute?

Anonymous Toad 12mth #1043abd

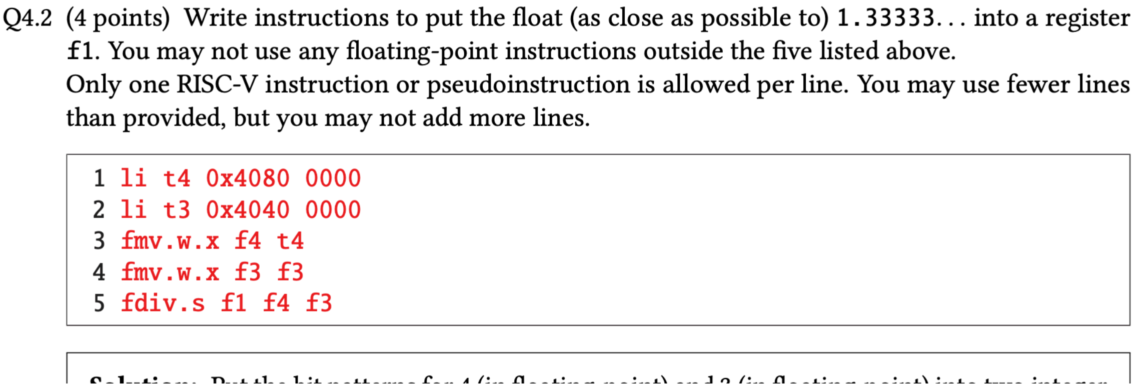
Same question, and I'm wondering why it's 3(ra) instead of 1(ra) for a little endian system

Anonymous Human 12mth #1043ace ENDORSED

The 3 comes from taking the bits [31:24] from the instruction. This makes sense for little endian, because if we were to take 0(ra), this would be bits [7:0]. Then, we know that funct7 takes the bits [31:25] so changing from 000 0000 to 010 0000 would be the same as changing from 0000 000? to 0100 000? concerning the bits [31:24]. Then, wanting to flip the 7th bit is the same as xori'ing with 0b0100 0000 = 64.

Anonymous Jackal 12mth #1043fd ✓ Resolved

Hi, my question is that why can't we just say li t4 4? doesn't fmv.w.x converting integer to float, so when we pass t4 = 4 it convert it to 4.0 and puts it in f4, so why are we saying li t4 0x4080 0000?



Q4.2 (4 points) Write instructions to put the float (as close as possible to) 1.33333... into a register f1. You may not use any floating-point instructions outside the five listed above.
Only one RISC-V instruction or pseudoinstruction is allowed per line. You may use fewer lines than provided, but you may not add more lines.

```
li t4 0x4080 0000
li t3 0x4040 0000
fmv.w.x f4 t4
fmv.w.x f3 f3
fdiv.s f1 f4 f3
```

Solution: Put the bit patterns for 4 (in floating point) and 3 (in floating point) into two integer

Andrew Liu STAFF 12mth #1043cba

> fmv.w.x takes the bitstring in an integer register rs1, and transfers the bitstring over to a floating-point register rd.

> For example, the value 4 is represented in single-precision floating point numbers as 0x4080 0000. If t0 contained the hexadecimal value 0x4080 0000 and we ran fmv.w.x f0 t0, then f0 would be set to the floating point value 4.0.

Does this answer your question?

Anonymous Hornet 12mth #1043fc ✓ Resolved

Su22-FinalQ7.4

Why does the page table remember VPN's across different processes but the TLB is cleared out/flushed every time we change processes? Also, is knowing whether we have a page fault, TLB hit, or TLB miss but page table hit when we're dealing with multiple processes in scope?

1

Jero Wang STAFF 12mth #1043bff

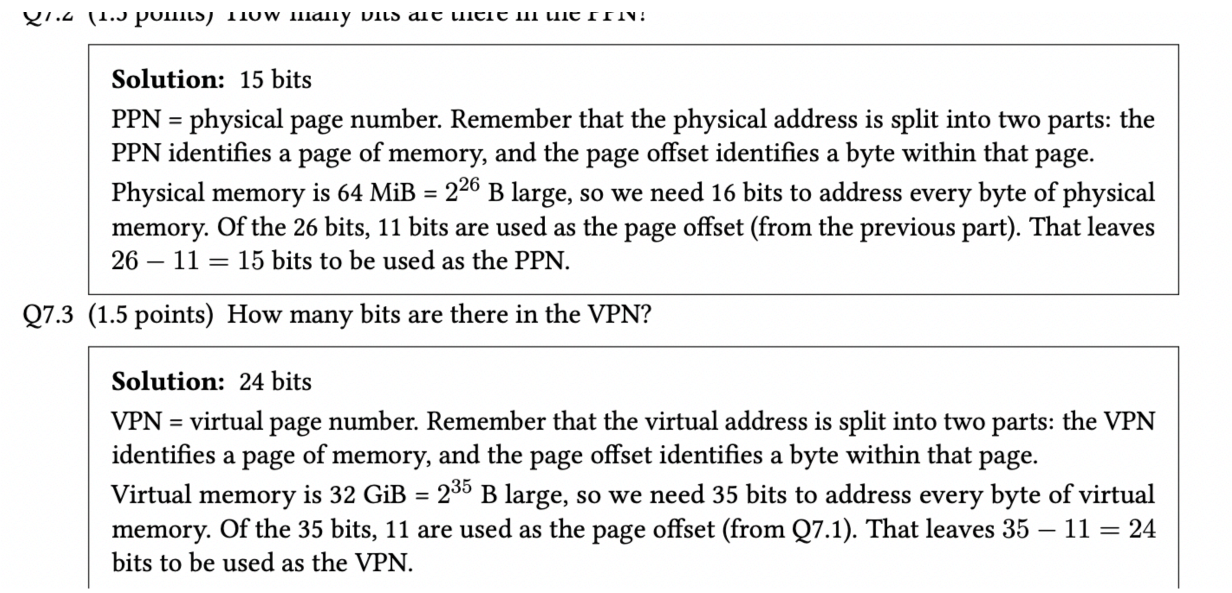
The page table must persist after switching to a different process then back because this mapping isn't changed, whereas TLB is flushed because switching to a different process means switching to a different address space, and your old entries are invalid. Technically, you can also restore the TLB to a previous state in an implementation of a TLB.

Yes, it's in scope.

Anonymous Manatee 12mth #1043fb ✓ Resolved

SP22-Final-Q7.2-7.3

for 7.2, how do we know 16 bits are needed to address every byte of physical memory, and for 7.3, how do we know 35 bits are needed to address every byte of virtual memory?



Q7.2 (1.5 points) How many bits are there in the PPN?

Solution: 15 bits

PPN = physical page number. Remember that the physical address is split into two parts: the PPN identifies a page of memory, and the page offset identifies a byte within that page.

Physical memory is 64 MiB = 2^{26} B large, so we need 16 bits to address every byte of physical memory. Of the 26 bits, 11 bits are used as the page offset (from the previous part). That leaves $26 - 11 = 15$ bits to be used as the PPN.

Q7.3 (1.5 points) How many bits are there in the VPN?

Solution: 24 bits

VPN = virtual page number. Remember that the virtual address is split into two parts: the VPN identifies a page of memory, and the page offset identifies a byte within that page.

Virtual memory is 32 GiB = 2^{35} B large, so we need 35 bits to address every byte of virtual memory. Of the 35 bits, 11 are used as the page offset (from Q7.1). That leaves $35 - 11 = 24$ bits to be used as the VPN.

1

Anonymous Jay 12mth #1043aac

7.2 looks like a typo to me, it should probably say 26 bits. The number of bits needed to address every byte of physical/virtual memory is the exponent of the size of the physical/virtual memory.

♡ ···

Jero Wang STAFF 12mth #1043bda

Yeah, 7.2 should be 26, sorry. For 35 bits, we get it from 32GiB of virtual memory, since 32GiB = 2^5GiB and 1GiB = 2^10B.

♡ ···

Anonymous Okapi 12mth #1043ef ✓ Resolved

SP22-Final-Q6

For the caching question why wouldn't calling prime[0] = 1 initiate the cache? In the solution it says calling prime[4] = 1 is the first miss but wouldn't calling prime[0] at the start do that instead?

♡ 6 ···

Anonymous Herring 12mth #1043abc

^same question

♡ 1 ···

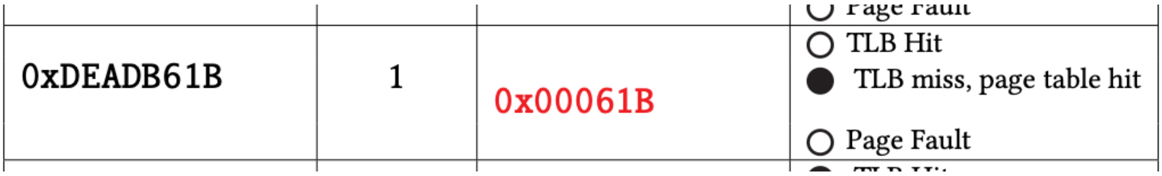
Jero Wang STAFF 12mth #1043bcf

Yes, you're right, I've noted it in our list of typos. This shouldn't change the answer though.

♡ 1 ···

Anonymous Wren 12mth #1043ee ✓ Resolved

for SU22-Final-Q7.4



0xDEADB61B	1	0x00061B	○ TLB Hit ● TLB miss, page table hit ○ Page Fault

why is row 5 a table miss, page table hit?

♡ ···

Anonymous Caterpillar 12mth #1043ade

At that point we do have an existing pid of 1 and VPN DEADB mapping to PPN 000 coming from the very first row so it exists in our page table however in our TLB we are limited to 2 entries which are the procid of 2 and 0000 and procid of 2 and DEADB at that point neither of which is procid of 1 and DEADB so it's a TLB miss.

♡ ···

Anonymous Wren 12mth #1043bab

when the first row is executed, why does is it not saved into the tlb?

♡ ···

Anonymous Caterpillar 12mth #1043bba

Replying to Anonymous Wren

It does but keep in mind we are working with a "2-entry, fully associative TLB" so 2 entries max at a time

♡ ···

Anonymous Wren 12mth #1043bbd

Replying to Anonymous Caterpillar

but then for row 6, why is it a tlb miss and pte hit again? if tlb can only take two entries at a time, wouldn't tlb contain the DEADB VPN now that row 5 was executed?

Anonymous Caterpillar 12mth #1043bbe

Replying to Anonymous Wren

Solutions say it's a TLB hit for row 6 not a tlb miss and pte hit so you are right in saying "if tlb can only take two entries at a time, wouldn't tlb contain the DEADB VPN now that row 5 was executed?"

Anonymous Wren 12mth #1043bca

Replying to Anonymous Caterpillar

got it! Thank you!

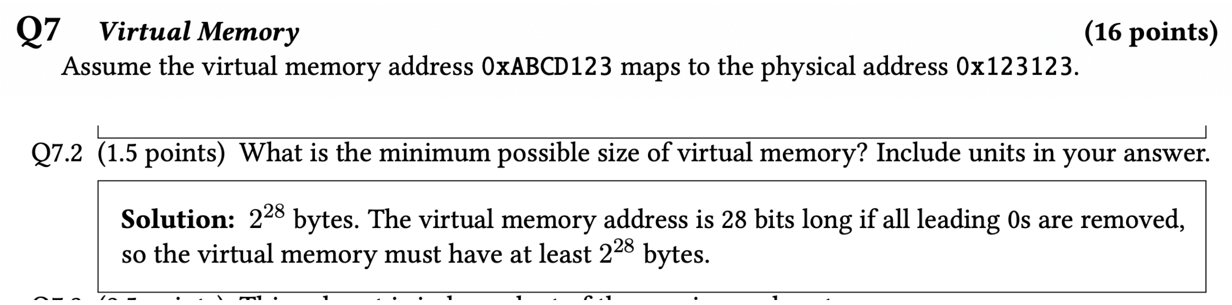
goodluck tomorrow!

Anonymous Meerkat 12mth #1043ec ✓ Resolved

SU22-Final-Q7

For the following question, the virtual memory address has 7 bytes, which is equivalent to 28 bits. 28 bits allows for 2^28 bytes of memory to be accessed.

I don't understand the statement that says: that the address is 28 bits if all leading 0s are removed. Isn't the virtual memory address 28 bits always?



Q7 ***Virtual Memory*** **(16 points)**

Assume the virtual memory address `0xABCD123` maps to the physical address `0x123123`.

Q7.2 (1.5 points) What is the minimum possible size of virtual memory? Include units in your answer.

Solution: 2^{28} bytes. The virtual memory address is 28 bits long if all leading 0s are removed, so the virtual memory must have at least 2^{28} bytes.

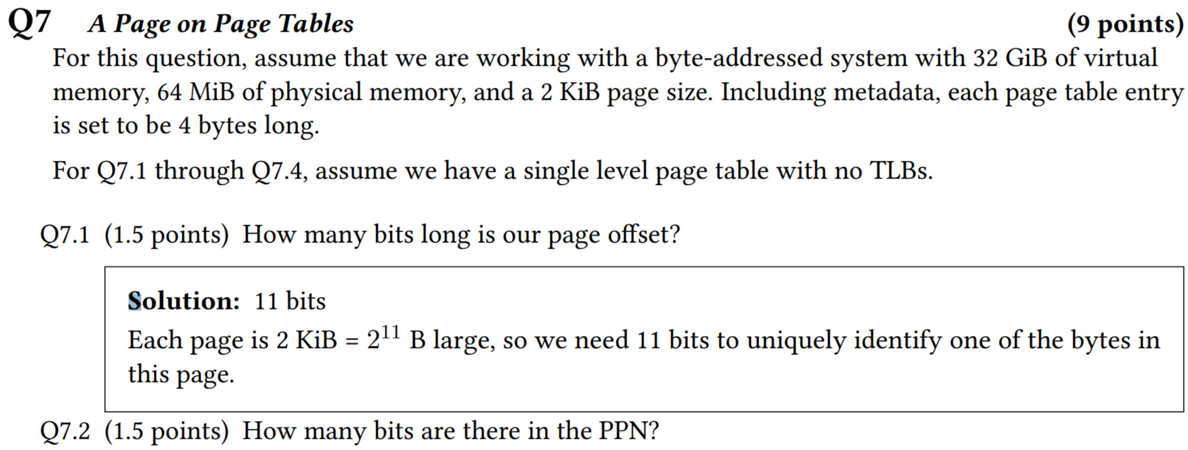
1

Jero Wang STAFF 12mth #1043cac

Where does it say that the virtual address space have 7 bytes? Assuming you mean nibbles/hex characters, it doesn't necessarily guarantee that the address is 28 bits if it has 7 nibbles. For example, a 26 bit number can be represented as `0x3456789` .

Anonymous Ram 12mth #1043eb ✓ Resolved

SP22-Final-Q7.1



Q7 *A Page on Page Tables* **(9 points)**

For this question, assume that we are working with a byte-addressed system with 32 GiB of virtual memory, 64 MiB of physical memory, and a 2 KiB page size. Including metadata, each page table entry is set to be 4 bytes long.

For Q7.1 through Q7.4, assume we have a single level page table with no TLBs.

Q7.1 (1.5 points) How many bits long is our page offset?

Solution: 11 bits

Each page is 2 KiB = 2^{11} B large, so we need 11 bits to uniquely identify one of the bytes in this page.

Q7.2 (1.5 points) How many bits are there in the PPN?

Just to clarify, does the page offset always correspond to page size? And if we calculate virtual memory / page size we are finding the number of different VPNs we can have?

Jero Wang STAFF 12mth #1043bdf

Yes and yes. Page offset needs to represent values from 0 to page size - 1.

Anonymous Sardine 12mth #1043ea ✓ Resolved

SP22-Final-Q6.1

Why isn't it n + 1 instead of n? I thought we need the last element of the array to signify where it ends

Anonymous Wombat 12mth #1043fa

i believe a null terminator is only required for strings

Jero Wang STAFF 12mth #1043bea

^ we're using chars here as 1 byte integers, not to store an actual string.

Anonymous Bison 12mth #1043df ✓ Resolved

Sp22-Final-Q5.5

I understand how push can cause data hazards if we are writing to sp in the instruction right after.

Would it be valid to say that it also causes control hazards, because if we store ra on the stack, and we have a ret a couple instructions after push, we would load in the wrong value of ra and jump incorrectly?

Jero Wang STAFF 12mth #1043beb

Your code should account for this, so it's not a hazard. Control hazards are when you jump/branch to the incorrect instruction.

Anonymous Goat 12mth #1043de ✓ Resolved

Su22-Final-Q5.1

How do you know to take the last 7 bits (1001110) when the raw data is 1001110 1000011?

♡ ···

Andrew Liu STAFF 12mth #1043cbb

We do this because our standard has been defined as two concatenated Hamming codes of bit length 7.

♡ ···

Anonymous Duck 12mth #1043dd ✓ Resolved

SP22-Final-Q4.4

I am a little confused about why it is true that any floating point number can be represented as an odd number multiplied/divided by a power of 2. Why is this true?

♡ 2 ···

J Justin Yokota STAFF 12mth #1043ed

Try proving this for integers first; for example, how would you write 192 as an odd number multiplied by a power of two?

♡ ···

Anonymous Frog 12mth #1043dc ✓ Resolved

SP22-Final-Q7.1
Is this a typo. You share the last 3 hex bits which would be 12 bits not 13. So it seems to me like the answer would be 2^12 bytes which would make sense since the other question uses this assumption to gett to 2^28. (2^16 * 2^12 = 2^28)

♡ 3 ···

Anonymous Hamster 12mth #1043aad

I have the same question as well. also this is SU22-Final-Q7.1

♡ ···

Anonymous Toad 12mth #1043abf

Same question

♡ ···

Anonymous Koala 12mth #1043acb

Replying to Anonymous Toad

Same

♡ ···

Anonymous Grasshopper 12mth #1043bec ENDORSED

'D' and '3' share their last bit, so that's actually the first bit of the offset (along with the identical 0x123 after, thus adding to 13 total bits).

♡ ···

J Jero Wang STAFF 12mth #1043bfa

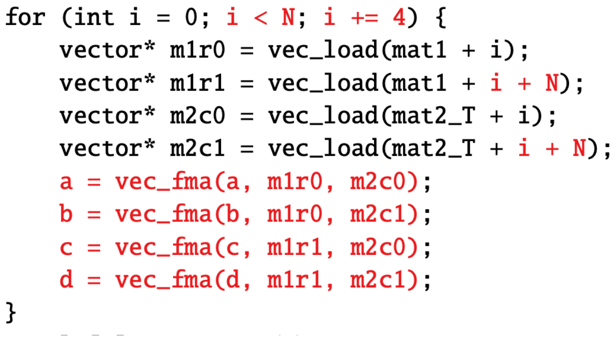
As Anon Grasshopper said, they share the LSB of the fourth nibble. Not sure what you mean by the other question, but Q7.2 does not rely on this calculation and Q7.3 is independent of the previous subparts, so it doesn't say anything about this answer.

♡ ···

Anonymous Goat 12mth #1043db ✓ Resolved

SP22-Final-Q8.1

I'm having trouble understanding why for vector* m1r1 & vector* m2c1 we need to add i + N. Would appreciate further explanation & examples to understand this better. Thanks.



```
for (int i = 0; i < N; i += 4) {
    vector* m1r0 = vec_load(mat1 + i);
    vector* m1r1 = vec_load(mat1 + i + N);
    vector* m2c0 = vec_load(mat2_T + i);
    vector* m2c1 = vec_load(mat2_T + i + N);
    a = vec_fma(a, m1r0, m2c0);
    b = vec_fma(b, m1r0, m2c1);
    c = vec_fma(c, m1r1, m2c0);
    d = vec_fma(d, m1r1, m2c1);
}
```

♡ 1 ···

Jero Wang STAFF 12mth #1043bdb

`m1r1` is row 1 (the second row, since rows are 0-indexed). To reach the second row, we need to add the width of each row to the pointer to the first row. Similarly, since we have a transposed matrix for m2, the second column in the original matrix becomes the second row in the transposed matrix, and the same logic applies.

♡ ···

Taeyoung Kim 12mth #1043cf ✓ Resolved

SU22-Final-Q7.3, should we not assume a valid bit is attached at the most significant bit?

♡ ···

Andrew Liu STAFF 12mth #1043cbc

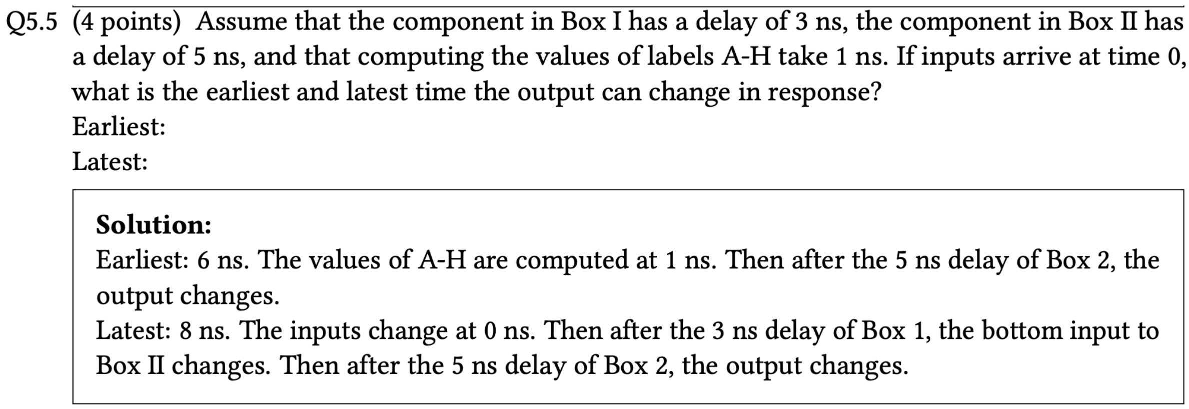
Valid bit constitutes metadata, which is accounted for in the problem.

♡ ···

Anonymous Dragonfly 12mth #1043ce ✓ Resolved

SU22-Final-Q5.5

I'm sorry, I can't understand the answer for Q5.5, can someone elaborate a bit more? Can't the earliest be 5ns? Can't mux(5ns) and the input(1ns) be operated concurrently?



Q5.5 (4 points) Assume that the component in Box I has a delay of 3 ns, the component in Box II has a delay of 5 ns, and that computing the values of labels A-H take 1 ns. If inputs arrive at time 0, what is the earliest and latest time the output can change in response?

Earliest:

Latest:

Solution:

Earliest: 6 ns. The values of A-H are computed at 1 ns. Then after the 5 ns delay of Box 2, the output changes.

Latest: 8 ns. The inputs change at 0 ns. Then after the 3 ns delay of Box 1, the bottom input to Box II changes. Then after the 5 ns delay of Box 2, the output changes.

♡ 4 ···

Anonymous Tarsier 12mth #1043aaf

A TA can correct me if I'm wrong, but I think the mux can only change after the inputs are calculated, not while they are being calculated.

Anonymous Ostrich 12mth #1043adf

Sorry to bother...why is there a matter of early and late? I think all inputs need to go through Box 1, is that correct?

Jero Wang STAFF 12mth #1043bef

A-H are also "inputs' to this CL circuit, and their critical path is shorter since they only go through box II.

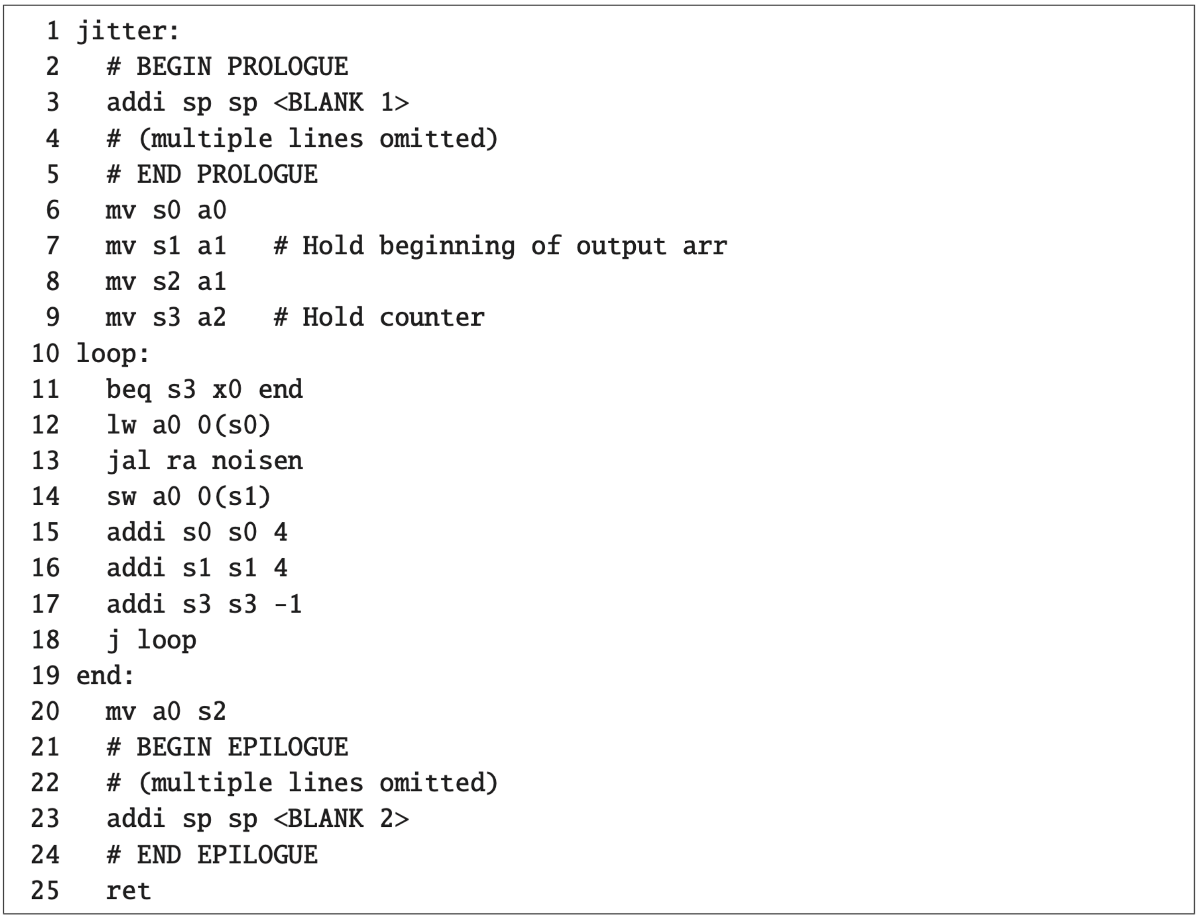
Jero Wang STAFF 12mth #1043bee

This question could have been worded better, but in general, during the CL delay of a circuit component, the output is undefined, which means that the earliest defined output of this circuit would be at 6ns. But yes, it may change sometime between 5ns and 6ns, but the value is just undefined.

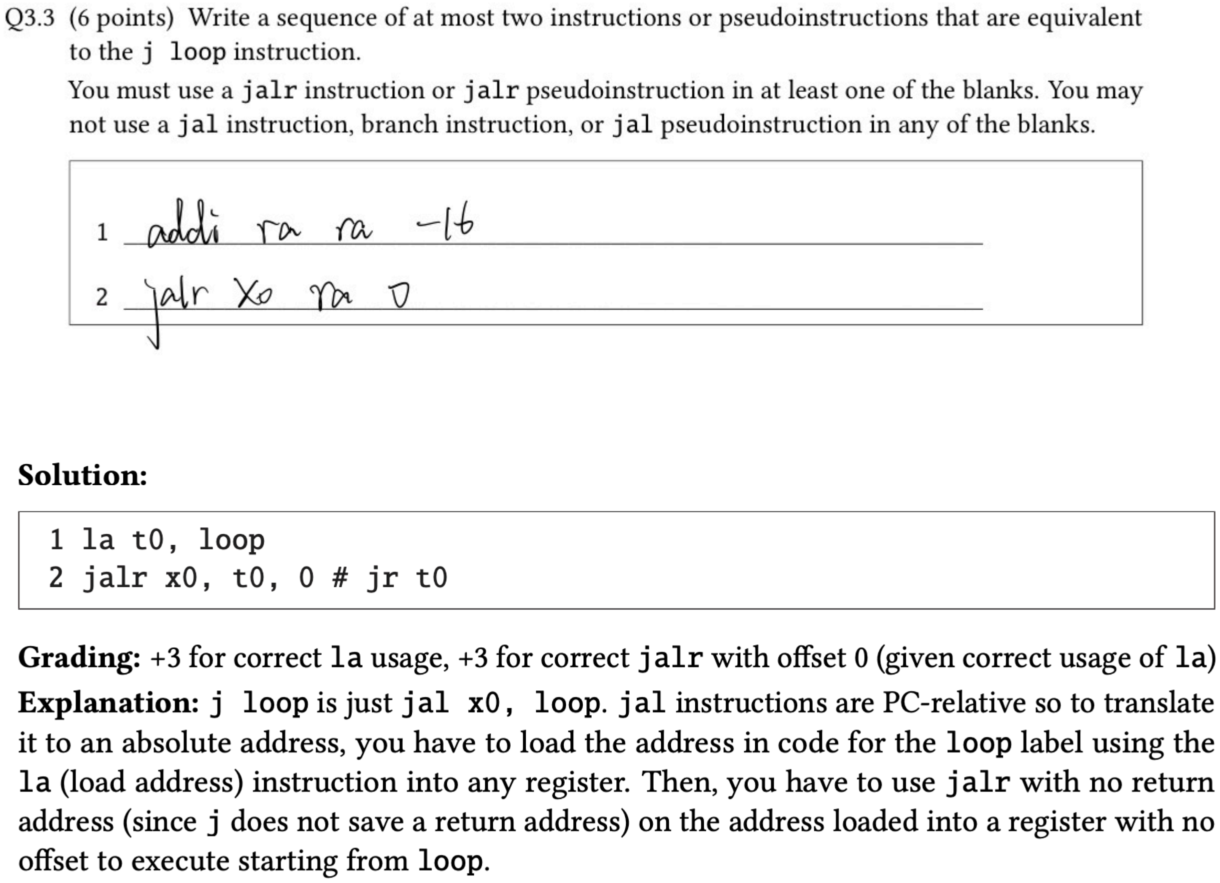
Anonymous Dragonfly 12mth #1043cd ✓ Resolved

SU22-Final-Q3

Hi, is my answer valid?



```
jitter:
  # BEGIN PROLOGUE
  addi sp sp <BLANK 1>
  # (multiple lines omitted)
  # END PROLOGUE
  mv s0 a0
  mv s1 a1    # Hold beginning of output arr
  mv s2 a1
  mv s3 a2    # Hold counter
loop:
  beq s3 x0 end
  lw a0 0(s0)
  jal ra noisen
  sw a0 0(s1)
  addi s0 s0 4
  addi s1 s1 4
  addi s3 s3 -1
  j loop
end:
  mv a0 s2
  # BEGIN EPILOGUE
  # (multiple lines omitted)
  addi sp sp <BLANK 2>
  # END EPILOGUE
  ret
```



Q3.3 (6 points) Write a sequence of at most two instructions or pseudoinstructions that are equivalent to the `j loop` instruction.

You must use a `jalr` instruction or `jalr` pseudoinstruction in at least one of the blanks. You may not use a `jal` instruction, branch instruction, or `jal` pseudoinstruction in any of the blanks.

```
addi ra ra -16
jalr X0 ra 0
```

Solution:

```
la t0, loop
jalr x0, t0, 0 # jr t0
```

Grading: +3 for correct `la` usage, +3 for correct `jalr` with offset 0 (given correct usage of `la`)

Explanation: `j loop` is just `jal x0, loop`. `jal` instructions are PC-relative so to translate it to an absolute address, you have to load the address in code for the `loop` label using the `la` (load address) instruction into any register. Then, you have to use `jalr` with no return address (since `j` does not save a return address) on the address loaded into a register with no offset to execute starting from `loop`.

Justin Yokota STAFF 12mth #1043da

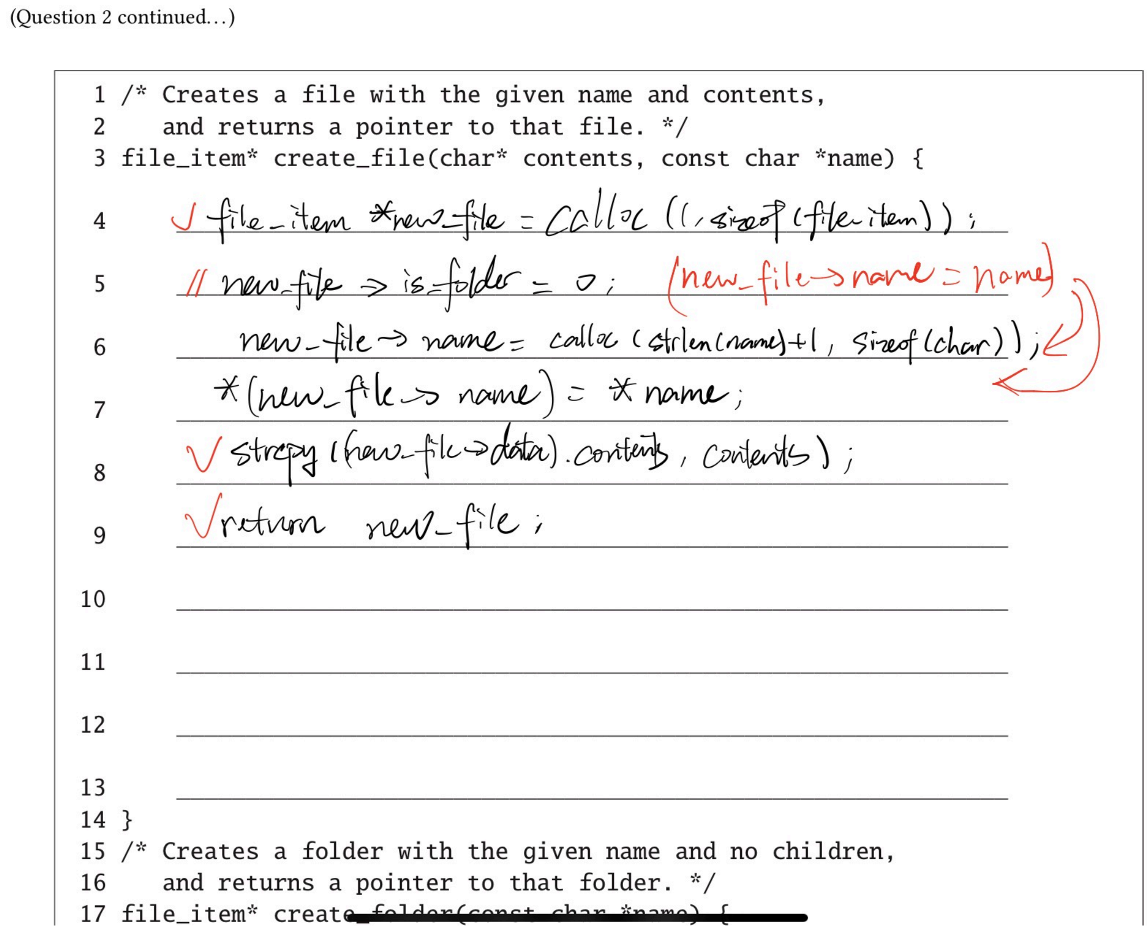
Not quite; you'd end up jumping to the "mv s3 a2" line instead.

Anonymous Human 12mth #1043acd

does that mean that changing the answer to addi ra ra -12 would work?

Anonymous Dragonfly 12mth #1043cc ✓ Resolved

SU22-Final-Q2



(Question 2 continued…)

```
/* Creates a file with the given name and contents,
   and returns a pointer to that file. */
file_item* create_file(char* contents, const char *name) {
    ✓ file_item *new_file = calloc(1, sizeof(file_item));
    // new_file -> is_folder = 0;    (new_file->name = name)
        new_file -> name = calloc(strlen(name)+1, sizeof(char));
      *(new_file -> name) = *name;
    ✓ strcpy((new_file->data).contents, contents);
    ✓ return new_file;
}
/* Creates a folder with the given name and no children,
   and returns a pointer to that folder. */
file_item* create~~_folder(const char *name) {~~
```

Answer says:
file_item* new_file = (file_item*) calloc(1, sizeof(file_item));
new_file->name = name;
strcpy(new_file->data.contents, contents);
return new_file;

Hi, is my answer valid?

Jiries Kaileh 12mth #1043bae

I looked at the walkthrough video and apparently you do not have to allocate for memory for

new_file->name because the input "name" is a constant char*, so we can assume it will never be freed. I think that if you allocated memory for new_file->name and used strcpy to copy the string over, I think the solution should still be valid, but I would need a TA to confirm this.

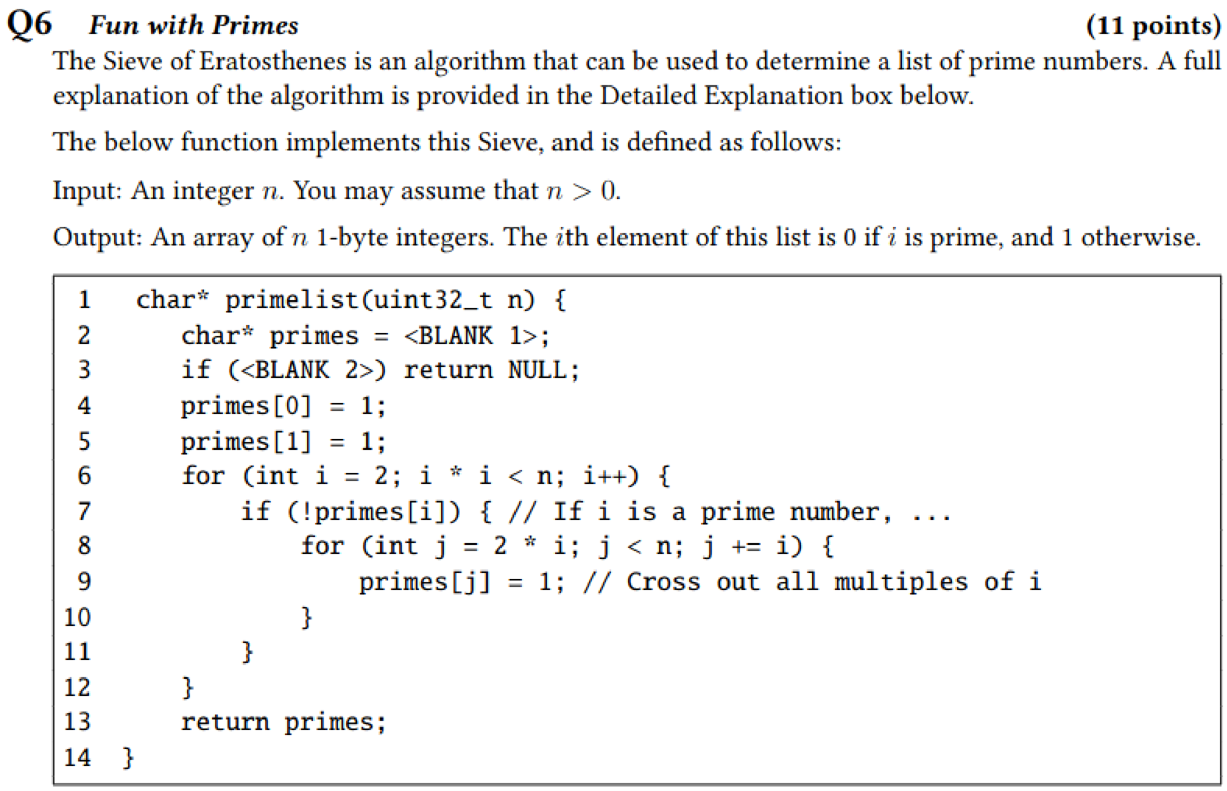
Andrew Liu STAFF 12mth #1043cbf

This solution appears to only copy over 1 character of `name` to `new_file->name`. If you used `strcpy`, I believe your solution would've been equivalent. Remember that taking `*` on a `char*` type gives you a single `char`!

Anonymous Goshawk 12mth #1043cb ✓ Resolved

SP22-Final-Q6

How do we know that `n > 1`? If `n == 1` then wouldn't the line `primes[1] = 1;` cause an error?



Q6 ***Fun with Primes*** **(11 points)**

The Sieve of Eratosthenes is an algorithm that can be used to determine a list of prime numbers. A full explanation of the algorithm is provided in the Detailed Explanation box below.

The below function implements this Sieve, and is defined as follows:

Input: An integer n. You may assume that $n > 0$.

Output: An array of n 1-byte integers. The ith element of this list is 0 if i is prime, and 1 otherwise.

```
char* primelist(uint32_t n) {
    char* primes = <BLANK 1>;
    if (<BLANK 2>) return NULL;
    primes[0] = 1;
    primes[1] = 1;
    for (int i = 2; i * i < n; i++) {
        if (!primes[i]) { // If i is a prime number, ...
            for (int j = 2 * i; j < n; j += i) {
                primes[j] = 1; // Cross out all multiples of i
            }
        }
    }
    return primes;
}
```

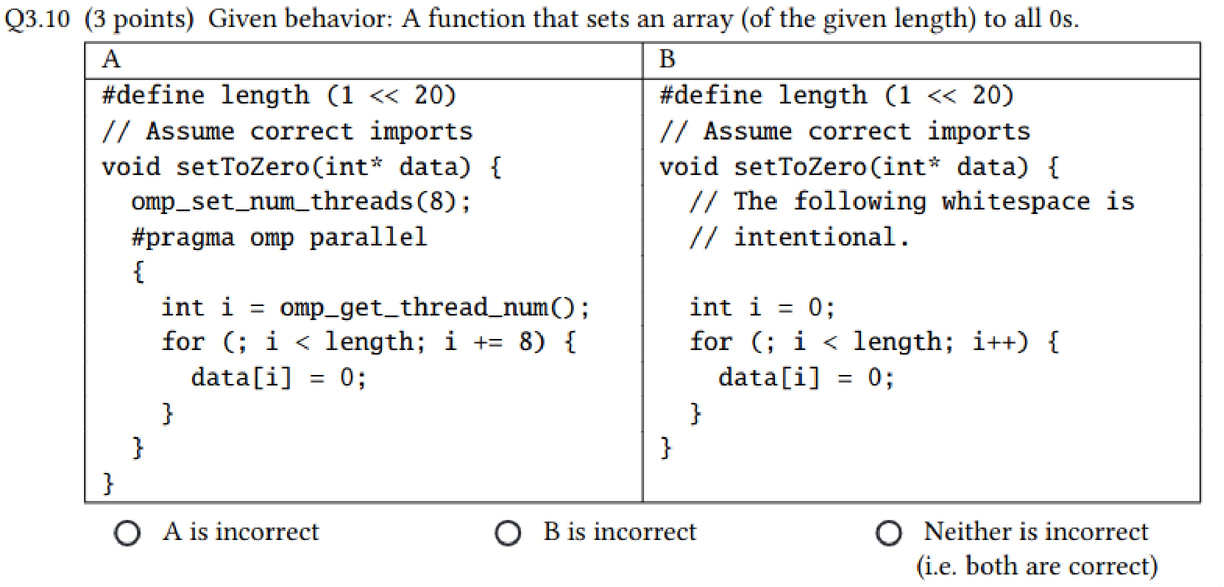
Andrew Liu STAFF 12mth #1043cbd

I think this is a typo in the question, and you should assume n > 1. Nice catch!

Anonymous Goshawk 12mth #1043bf ✓ Resolved

SP22-Final-Q3

The correct answers all mention thrashing. How do we know the structure of the cache and that the data accesses overwrite each other in the cache? If it was a fully associative cache, wouldn't there be no thrashing since all of the values would be in the cache?



Q3.10 (3 points) Given behavior: A function that sets an array (of the given length) to all 0s.

A	B

`#define length (1 << 20)`
`// Assume correct imports`
`void setToZero(int* data) {`
`  omp_set_num_threads(8);`
`  #pragma omp parallel`
`  {`
`    int i = omp_get_thread_num();`
`    for (; i < length; i += 8) {`
`      data[i] = 0;`
`    }`
`  }`
`}`
	`#define length (1 << 20)`
	`// Assume correct imports`
	`void setToZero(int* data) {`
	`  // The following whitespace is`
	`  // intentional.`
	
	`  int i = 0;`
	`  for (; i < length; i++) {`
	`    data[i] = 0;`
	`  }`
	`}`

○ A is incorrect ○ B is incorrect ○ Neither is incorrect (i.e. both are correct)

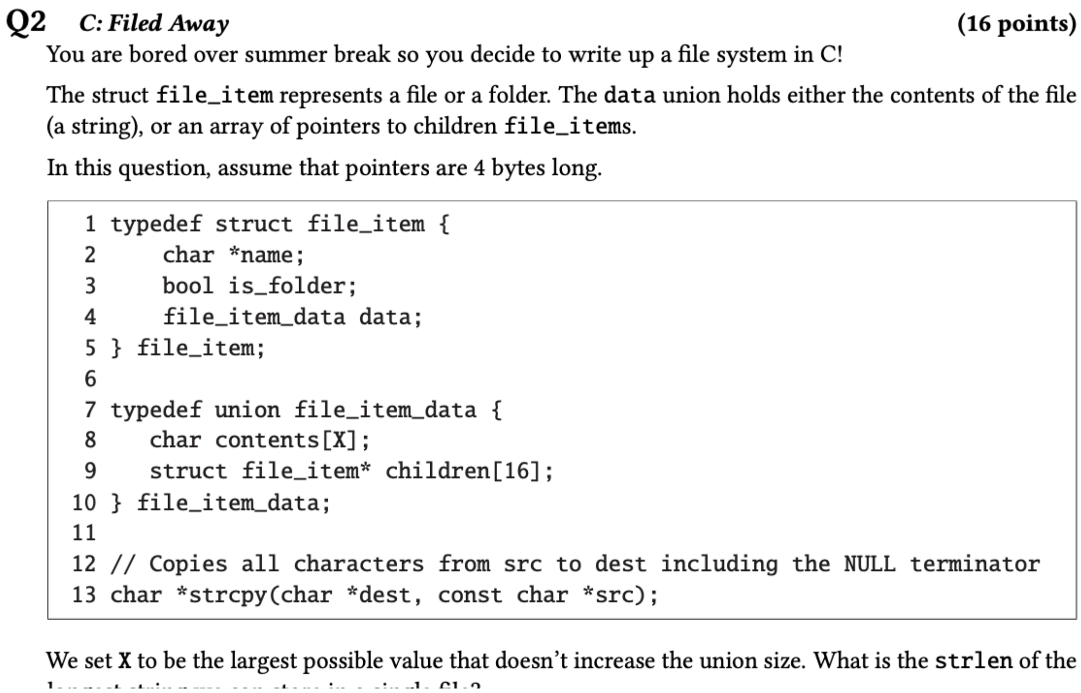
Justin Yokota STAFF 12mth #1043ca

Different threads have different caches, and to write to a location in a cache block, we must invalidate the same block in all other thread's caches (to avoid merge conflicts), regardless of the associativity. The only way we could avoid this is if our blocks were 4 bytes long, which is infeasibly small.

Anonymous Pony 12mth #1043bd ✓ Resolved

[SU22 Q2]

If we were to malloc file_item, would we also have to malloc file_item_data ? From my understanding, since contents is initialized on the stack, we wouldn't need to malloc for it . Similarly, file_item* children[16] is also on the stack. But each element children in the array needs to be malloced as they are pointers. How does this work



Q2 *C: Filed Away* **(16 points)**

You are bored over summer break so you decide to write up a file system in C!

The struct `file_item` represents a file or a folder. The `data` union holds either the contents of the file (a string), or an array of pointers to children `file_items`.

In this question, assume that pointers are 4 bytes long.

```
typedef struct file_item {
    char *name;
    bool is_folder;
    file_item_data data;
} file_item;

typedef union file_item_data {
   char contents[X];
   struct file_item* children[16];
} file_item_data;

// Copies all characters from src to dest including the NULL terminator
char *strcpy(char *dest, const char *src);
```

We set **X** to be the largest possible value that doesn't increase the union size. What is the `strlen` of the

Noah Yin STAFF 12mth #1043cae

When you allocate memory for a struct using malloc, it will do so for all members of the struct, which includes file_item_data, so you would not need to separately malloc memory for it.

For children in file_item_data, enough space would be allocated to hold 16 file_item pointers, but not the actual file_item structs themselves.

Anonymous Mink 12mth #1043bc ✓ Resolved

For **su22-final** Q 3.3,

would

auipc t0 0

jalr x0 t0 label

work?

1

Anonymous Snail 12mth #1043bcc

I have the same question, would really appreciate an answer!

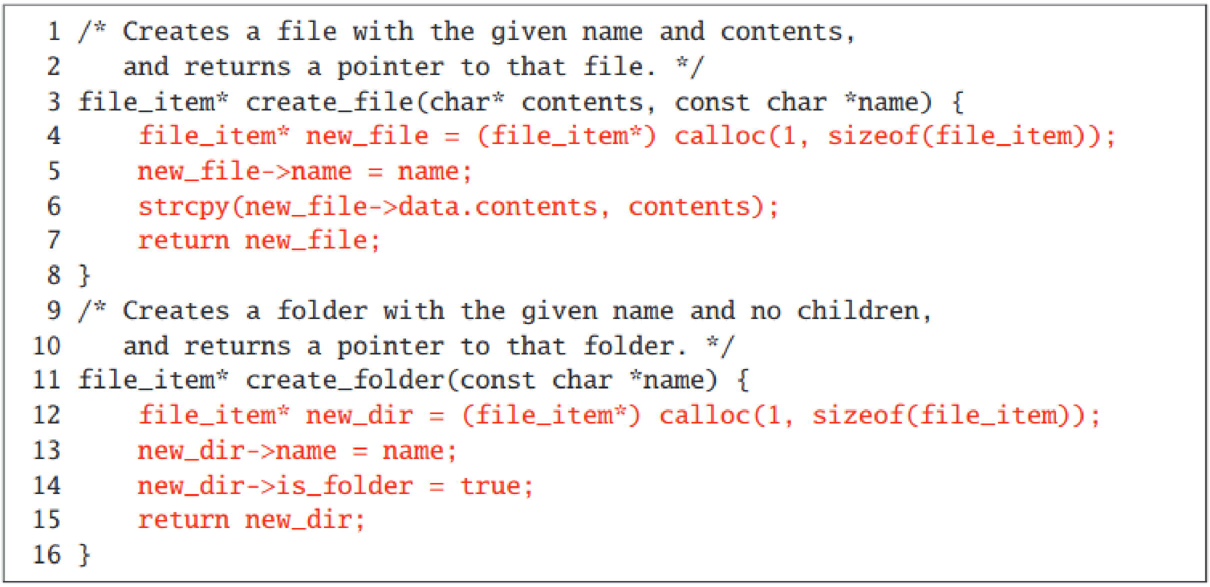
Jero Wang STAFF 12mth #1043caa

#1226a

Anonymous Goshawk 12mth #1043ba ✓ Resolved

SU22-Final-Q2

How come we don't have to malloc for `new_file->name` ? If `name` is allocated on the heap and is deallocated wouldn't the code break?



```
/* Creates a file with the given name and contents,
   and returns a pointer to that file. */
file_item* create_file(char* contents, const char *name) {
    file_item* new_file = (file_item*) calloc(1, sizeof(file_item));
    new_file->name = name;
    strcpy(new_file->data.contents, contents);
    return new_file;
}
/* Creates a folder with the given name and no children,
   and returns a pointer to that folder. */
file_item* create_folder(const char *name) {
    file_item* new_dir = (file_item*) calloc(1, sizeof(file_item));
    new_dir->name = name;
    new_dir->is_folder = true;
    return new_dir;
}
```

Anonymous Goshawk 12mth #1043bb

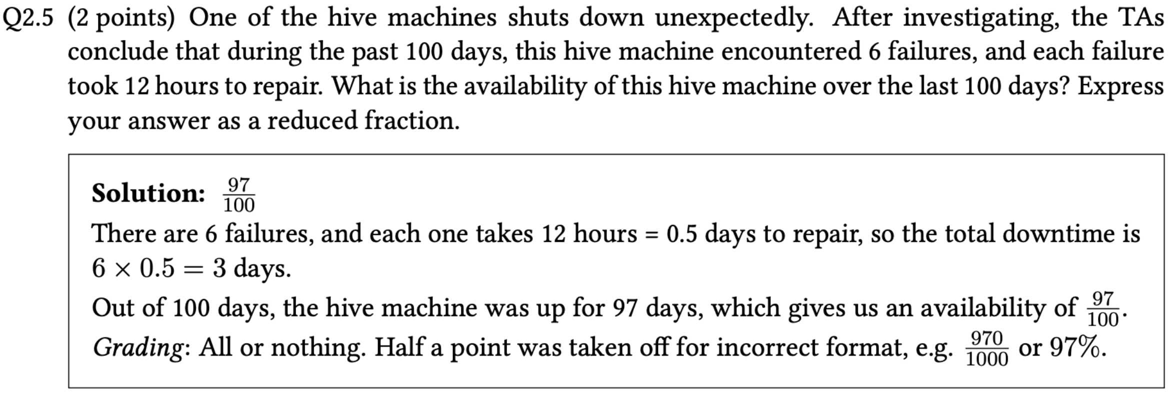
Nvm I just realized the input is `const char *name`

Anonymous Human 12mth #1043be

is *name static? if not, where are consts stored?

Anonymous Tapir 12mth #1043ae ✓ Resolved

SP22-final-Q2.5



Q2.5 (2 points) One of the hive machines shuts down unexpectedly. After investigating, the TAs conclude that during the past 100 days, this hive machine encountered 6 failures, and each failure took 12 hours to repair. What is the availability of this hive machine over the last 100 days? Express your answer as a reduced fraction.

Solution: $\frac{97}{100}$

There are 6 failures, and each one takes 12 hours = 0.5 days to repair, so the total downtime is $6 \times 0.5 = 3$ days.

Out of 100 days, the hive machine was up for 97 days, which gives us an availability of $\frac{97}{100}$.

Grading: All or nothing. Half a point was taken off for incorrect format, e.g. $\frac{970}{1000}$ or 97%.

For this question I used the MTTF/MTTF+MTTR formula and got 100/103, is that going to get full mark if we use that approach on exam?

♡ ···

N Nikhil Kandkur STAFF 12mth #1043af

In this question, we assume MTTF + MTTR = 100 since MTTF + MTTR is the total time the machine was active, which was 100 days. You are assuming that the machine was active and functioning correctly for 100 days, which is not right.

♡ 1 ···

Anonymous Viper 12mth #1043aa ✓ Resolved

SU22-Final_Q5.5

I watched the video explanation but I am still a bit confused on why it would be 6 and 8 for the earliest and latest time the output can change? Why for the earliest time would you not consider the 3ms delay from box I?

♡ ···

N Nikhil Kandkur STAFF 12mth #1043ad

You would not consider the 3ms delay from Box 1 for the earliest since we only want to know when the value would change: in order for the output to change, at least 1 input into the mux should be calculated and ready (we don't necessarily need to wait for all inputs), and then the mux should process these inputs in its 5 ms delay, which gives us 1 + 5 = 6.

♡ ···

Anonymous Elephant 12mth #1043aae

For the latest time, why don't we add 1ns for computing the values of labels A~H?

♡ 1 ···

Anonymous Manatee 12mth #1043afe

Replying to Anonymous Elephant

^

♡ ···

Anonymous Lion 12mth #1043f ✓ Resolved

SU22-Final-Q3.3

Would jalr x0 sp -28 also be a valid solution, as I believe this would set the stack pointer to 7 lines prior?

♡ ···

N Nikhil Kandkur STAFF 12mth #1043ac

`jalr x0 sp -28` would not set the stack pointer to 7 lines prior: it would rather set PC to `RF[sp] - 28`, and would not store the return address anywhere.

♡ ···

Jiries Kaileh 12mth #1043bbb

To add on to this

would the following be a valid solution?:

```
auipc t0 0
jalr x0 t0 -28
```

This sets t0 = PC + 0

then the jalr sets PC = t0 - 28 = PC - 28 and x0 remains unchanged because of how the regfile is constructed.

Anonymous Lion 12mth #1043e ✓ Resolved

SU22-Final-Q2

Is the only reason we use calloc for the first function create_file to save ourselves from having to set is_folder to 0? If so, can we use malloc for the second function instead of calloc? Or is there another reason we need to use calloc in this case?

Nikhil Kandkur STAFF 12mth #1043ab

For the first function, yes that is a main reason.

For the second function however, we use calloc since we want to set all the children pointers in the `file_item_data` struct to NULL. With calloc, that is done automatically for us; had we decided to use malloc, then we would have to iterate through all the children and set all their pointers to NULL.

Anonymous Ram 12mth #1043fe

Would it have been a vaild answer to use malloc for the first function if we had set is_folder to True on a separate line?

Nikhil Kandkur STAFF 12mth #1043aaa

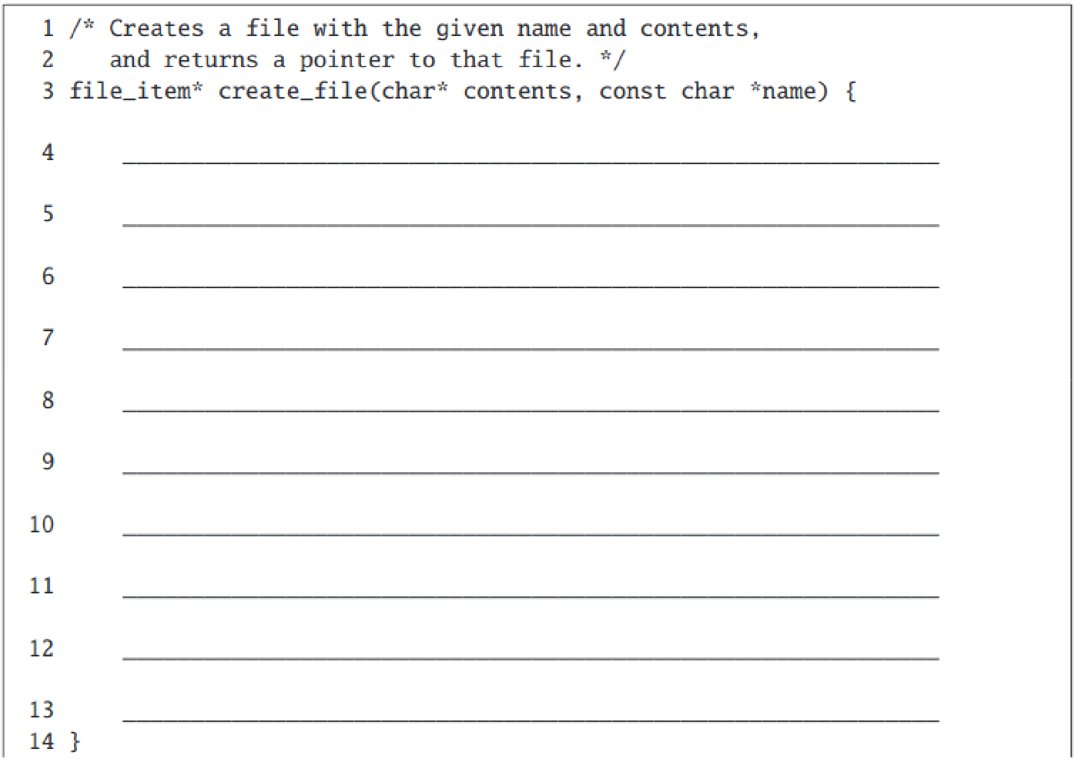
Replying to Anonymous Ram

Yup! But I don't think you had enough lines to do that in the first function

S.A. Sato 12mth #1043bcb

Replying to Nikhil Kandkur

Isn't this more than enough lines to set is_folder to false in the first function? Is there another reason we can't use malloc and add `is_folder = false;` ?



```
/* Creates a file with the given name and contents,
   and returns a pointer to that file. */
file_item* create_file(char* contents, const char *name) {

    ____________________________________________________________

    ____________________________________________________________

    ____________________________________________________________

    ____________________________________________________________

    ____________________________________________________________

    ____________________________________________________________

    ____________________________________________________________

    ____________________________________________________________

    ____________________________________________________________

    ____________________________________________________________
}
```

Nikhil Kandkur STAFF 12mth #1043bcd

Replying to S.A. Sato

Oop never mind, I didn't take a detailed look at the problem, but yeah that should be enough lines.

Anonymous Mongoose 12mth #1043c ✓ Resolved

SP22-Final-Q7.4

When they say "How many PTEs do we have in our table", and the answer is 2^24, does this mean that the page table is 4 * 2^24 bytes large (since size of PTE is 4 bytes)?

Nikhil Kandkur STAFF 12mth #1043d

Yup!

Anonymous Alpaca 12mth #1043a ✓ Resolved

SU22-Final-Q3.4

Is this question in scope for the final? If so, which lecture was modifying unprotected instruction memory covered in?

Nikhil Kandkur STAFF 12mth #1043b

The main concept which this question covers (the use of load/store instructions in memory), was covered in lecture, and therefore in scope for the final.