You are viewing this thread in readonly mode.

[Final] Past Exams - 2022 #1702

Jero Wang ADMIN
2 years ago in Exam – Final

1,451
VIEWS

You can find the past exams here: https://cs61c.org/sp23/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!

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**

Spring 2022 Final Walkthrough

Summer 2022 Final Walkthrough

Anonymous Sheep 2y #1702dfb ✓ Resolved

SU22-Final-Q5.2

The answer given is simply "We need to output the corrected data based on the parity bits; this can only be done using a mux" — however, I don't see how the capability to select one value out of many is able to accomplish the comparisons etc that I assume you would need? 5.1 makes sense because computing parity is fundamentally an XOR operation, but I don't see how a mux aligns with the second stage of the process.

Anonymous Crane 2y #1702dfc

Due to each Box I acting as p1, p2, p3, the left side will output which bit has an error (1-7) and 0 if no error. This is our selector in Box II for which corrected data to output. So if our inputs to Box II correspond with the correct output after the error bit is corrected, we can output the correct data. For instance, A is the original data because there was no error. B is the data with bit 1 flipped. C is the data with bit 2 flipped, etc.

Anonymous Deer 2y #1702dde ✓ Resolved

SU22-Final-Q6.4

The answer for this one is that "An input is accepted if it contains at most one 1 bit, and if neither the leading nor trailing bit is a 1. All other inputs are not accepted." However, what about the input 0110? This fits the accepted criteria, but wouldn't it end in a state labeled F?

Rosalie Fang ADMIN 2y #1702ded

At most 1 bit of 1. Your input has 2 bits of 1.

♡ 1 ···

Anonymous Vulture 2y #1702ddd ✓ Resolved

SP22-Final-Q8.1

Why don't we have to cast the inputs to vload and vstore to a __m128i* type?

♡ ···

E Erik Yang TA 2y #1702ddf

#1702bae

♡ ···

Anonymous Magpie 2y #1702dcd ✓ Resolved

SP22-Final-Q1.1

Where did we learn that b-type and j-type instructions can be 16 bits? On the reference card they are 32 bits long.

♡ ···

E Erik Yang TA 2y #1702ddb

not all riscv instructions are 32 bits long. The one we use in class is 32 bits, which is on the reference card, but there are other variants in RV instructions sets

♡ ···

Anonymous Magpie 2y #1702dcc ✓ Resolved

SP22-Final-Q8.1

Why dont we consider the edge cases, (when i is not divisible by 4, dont we end up using some value that is not in the matrix?) for example if i = N - 1, m2c1 would be vec_load(mat2_T + 2N - 1), which load 4 doubles starting at mat2_T + 2N - 1, but mat2_T is an array of 2N elements.

```
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);
```

♡ ···

E Erik Yang TA 2y #1702ddc

from the problem: you can assume N is a multiple of 4 so we don't need edge case

♡ ···

Anonymous Ram 2y #1702dba ✓ Resolved

SU22-Final-Q1.1

Could you explain the answer?

Thanks!

♡ ···

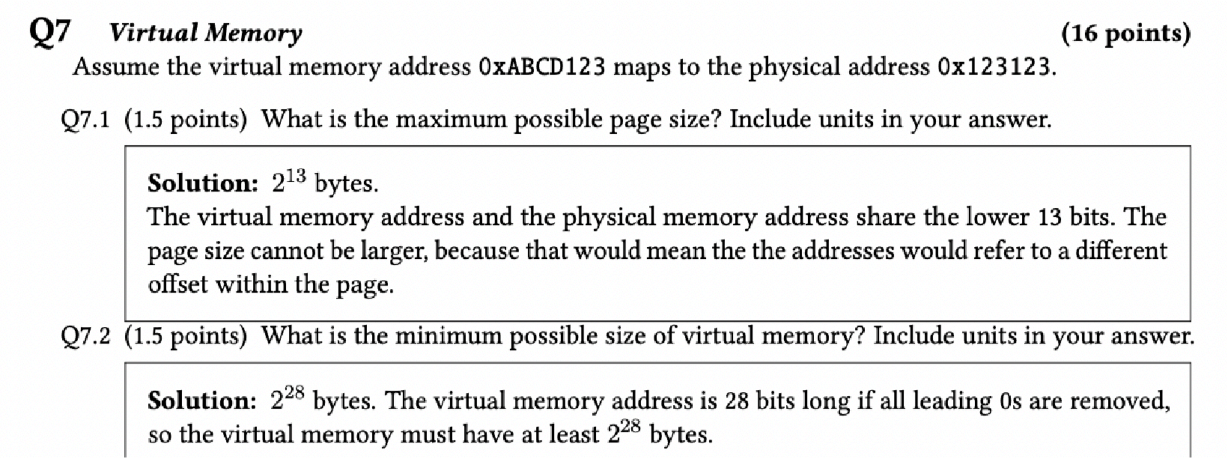
Anonymous Hornet 2y #1702dcb

Using absorption, x+yz = (x + y)(x + z).

Anonymous Chamois 2y #1702dad ✓ Resolved

SU22-Final-Q7.2

im still a bit confused as to how we get 28. where is the padding coming from? and isn't the min possible size of virtual memory = size of physical memory?



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

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

Q7.1 (1.5 points) What is the maximum possible page size? Include units in your answer.

Solution: 2^{13} bytes.
The virtual memory address and the physical memory address share the lower 13 bits. The page size cannot be larger, because that would mean the the addresses would refer to a different offset within the page.

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.

Rosalie Fang ADMIN 2y #1702dec

28 is just from virtual memory address being 0xABCD123, that's 28 bits to represent all the bytes in virtual memory.

Anonymous Kingfisher 2y #1702cff ✓ Resolved

when parallelizing, how can i tell if the for loop conditions should be something like i < (4 * N / 4) vs just i < N. it seems that the first one was used a lot in discussion and worksheets but i < N is used on the tests and I having trouble differentiating when one is appropriate vs the other. thanks!

Nikhil Kandkur TA 2y #1702dab

You would traditionally use `i < (N/4) * 4` when you are working with SIMD, where your vectors can store four elements of the type specified. If you do see `i < N` in something related to parallelism, there is a chance it could be thread-level parallelism instead of DLP, but do see what kind of parallelism you are working with before making any assumptions.

Anonymous Kingfisher 2y #1702dbf

hm that makes sense to me, but fa21-final q9 seems to only be using SIMD instructions where the vectorLoad function loads "4 uint64_t from ptr into a SIMD vector" without any pragma omp stuff going on, and they use i < n. could you please explain why that one is thread-level parallelism? sorry that this is technically in the wrong thread i guess.

Nikhil Kandkur TA 2y #1702dca

Replying to Anonymous Kingfisher

In that question, it is given that n is a multiple for 4, so doing `n/4*4` would be redundant in that case.

♡ 1 ···

Anonymous Kingfisher 2y #1702cfd ✓ Resolved

su22-final-Q7: why do u not include + 1 for valid bit and +1 for dirty bit? and what about permission bits? or is that what "metadata" collectively refers to?

♡ 1 ···

R Rosalie Fang ADMIN 2y #1702dea

"metadata" includes all relevant information about a mapping, so it includes valid + dirty bits. We usually say "1 bit for valid and x bits for **other** metadata" precisely for this reason.

♡ ···

Anonymous Badger 2y #1702cfc ✓ Resolved

SP22-Final-Q7.4

The PTE is calculated using VPN.

But why in **SU22-Final-Q7.3,** the PTE is calculated using PPN?

Which one is correct?

♡ ···

N Nikhil Kandkur TA 2y #1702dac

SP22-Final-Q7.4 asks for how many PTEs are stored in a page table, and that is equal to 2^(# VPN bits). **SU22-Final-Q7.3** asks for how big a single PTE is given the size of the physical space and a page, so both questions cover different areas of content.

♡ 1 ···

Anonymous Badger 2y #1702cdc ✓ Resolved

SP22-Final-Q2.4

Why don't we take into account the time to grade the short-answer?(Why the answer is not 250+50+12)?

♡ ···

Anonymous Kingfisher 2y #1702cde

we can parallelize doing both long-answer and short-answer in one go since we have 7billion threads.

♡ 1 ···

Anonymous Lyrebird 2y #1702cda ✓ Resolved

For su 22, 2 I had some questions about when we directly assign the pointer, and when we use strcpy. In general, if we have an array of pointers, we must use strcpy. But if we have a normal array, we can use an assignment operator. And when do we have to initialize all elements of our array to null?

♡ ···

E Erik Yang TA 2y #1702cdb

we cannot directly assign data->contents and contents var because they are different types. One is a char array while the other is a char pointer. That's why we have to use strcpy. We don't always have to initialize all elements to null. In this case, malloc would have worked as well as a solution

Anonymous Lyrebird 2y #1702cdf

is that the only reason? I thought char pointers are the same as a char array. Also what is the difference between (char* contents, char *name), the * is at different positions

Erik Yang TA 2y #1702cea

Replying to Anonymous Lyrebird

there is no difference between char * and char*. char[X] and char* are not the same thing.

Anonymous Lyrebird 2y #1702dcf

Replying to Erik Yang

"We don't always have to initialize all elements to null. In this case, malloc would have worked as well as a solution".

So when do we have to initialize all elements to null? I thought when we malloced space for an array, we also have to initialize all those array elements first.

Erik Yang TA 2y #1702dda

Replying to Anonymous Lyrebird

if you never intiailize a struct field, and then use it later on, that's when you could use calloc. I would say if you are given a line constraint in the exam, calloc would be a good start if you want to intiailize a struct

Anonymous Albatross 2y #1702ccc ✓ Resolved

SU22-Final-Q7.2

Why can't the maximum address be `0xABCD123` ? Is it a convention that the size of a virtual/physical memory is also a power of 2?

Erik Yang TA 2y #1702ceb

wdym by maximum address? usually, it'll be a power of 2

Anonymous Albatross 2y #1702cee

Like you can't go any further, such as `0xABCD123` ? The actual "maximum" address here is `0xFFFFFFF` , right?

Erik Yang TA 2y #1702cef

Replying to Anonymous Albatross

yeah the max address would be that address for the page table

♡ 1

Anonymous Albatross 2y #1702cfa

Replying to Erik Yang

Got it. Thank you! BTW I should've said `0xABCD124` .

Anonymous Chamois 2y #1702ccb ✓ Resolved

SU22-Final-Q2

When do we use -> and . assignments?

For this question, we used file->data.contents to access contents. Can we use file->data->contents to access contents?

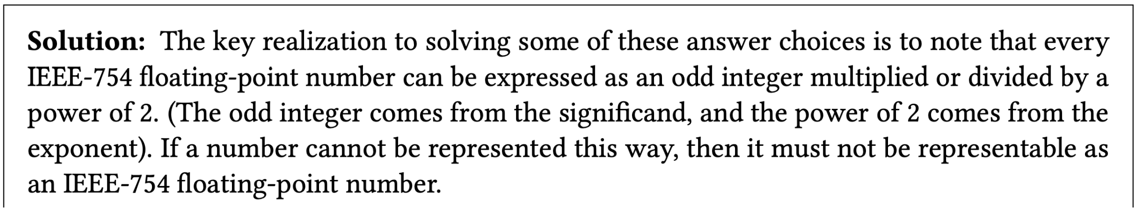
Erik Yang TA 2y #1702cdd

pointer notation is used to access fields of pointers to structs. Dot notation is used to access fields of structs themselves. file->data returns a struct, so that's why we can use dot notation here

Anonymous Elk 2y #1702cca ✓ Resolved

SP22-Final-Q4.4

I don't understand why we necessarily have the odd integer, can't the significand be even?



Solution: The key realization to solving some of these answer choices is to note that every IEEE-754 floating-point number can be expressed as an odd integer multiplied or divided by a power of 2. (The odd integer comes from the significand, and the power of 2 comes from the exponent). If a number cannot be represented this way, then it must not be representable as an IEEE-754 floating-point number.

♡ 4

Rosalie Fang ADMIN 2y #1702deb

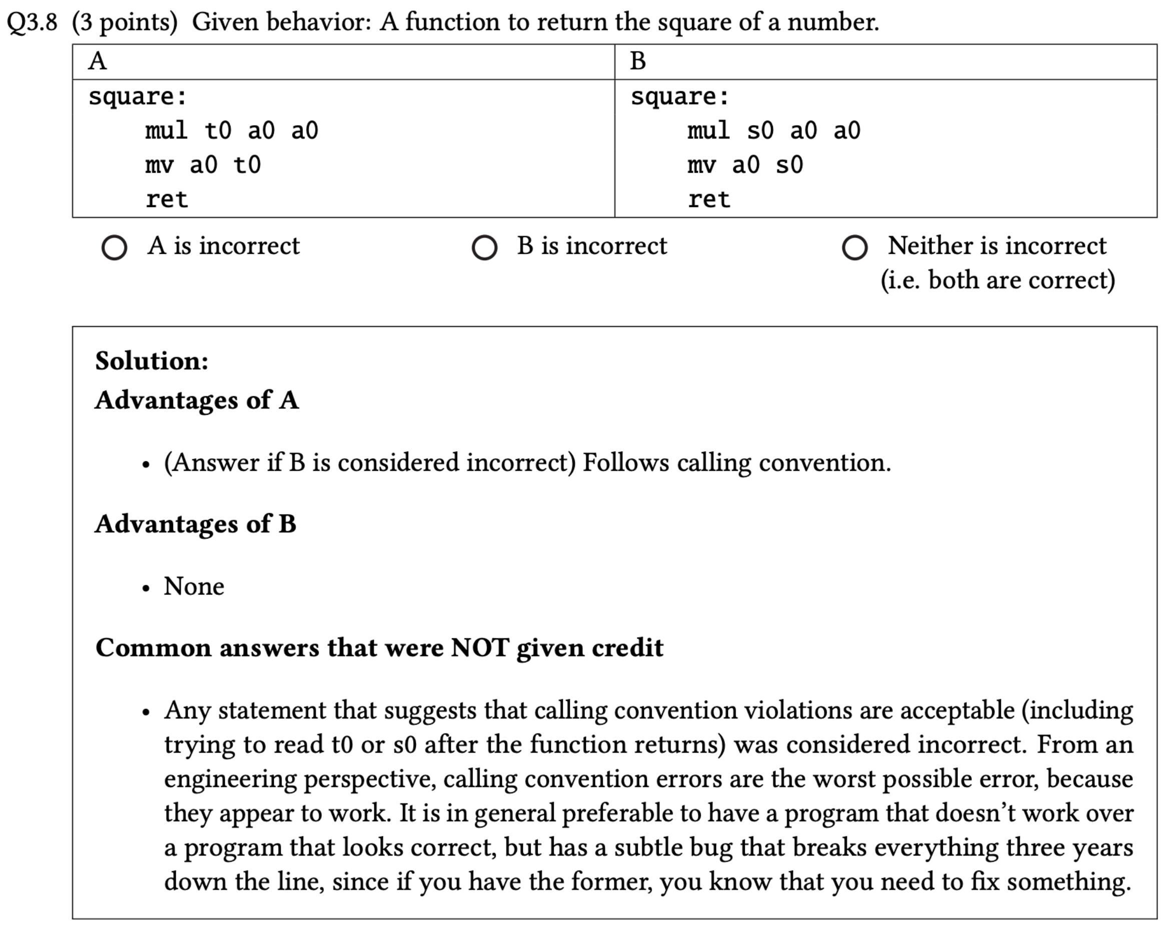
Ah great question so when you think about the IEEE-754 standard FP, it's always

2^exp * 1._ _ _ _ _... _, but when we have an even significand like 00100....... it's the same as if our overall number is 2^exp * 1.001 which is still an odd integer overall.

(Also I remember struggling on this problem when I was testsolving in SP22 the stress... ;-;)

Anonymous Elk 2y #1702cbf ✓ Resolved

SP22-Final-Q1



Q3.8 (3 points) Given behavior: A function to return the square of a number.

A	B

`square:`
`    mul t0 a0 a0`
`    mv a0 t0`
`    ret`
	`square:`
	`    mul s0 a0 a0`
	`    mv a0 s0`
	`    ret`

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

Solution:

Advantages of A

- (Answer if B is considered incorrect) Follows calling convention.

Advantages of B

- None

Common answers that were NOT given credit

- Any statement that suggests that calling convention violations are acceptable (including trying to read t0 or s0 after the function returns) was considered incorrect. From an engineering perspective, calling convention errors are the worst possible error, because they appear to work. It is in general preferable to have a program that doesn't work over a program that looks correct, but has a subtle bug that breaks everything three years down the line, since if you have the former, you know that you need to fix something.

why does B violate CC? Is tit because we didn't restore s0?

♡ 1 ···

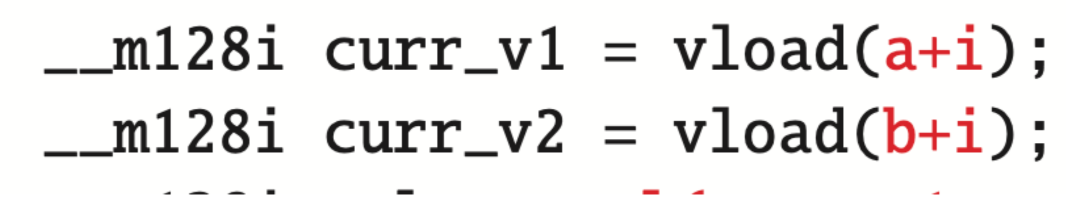
Erik Yang TA 2y #1702cec

since we use s registers, we need to use the stack to preserve their original values

♡ ···

Anonymous Cheetah 2y #1702cbd ✓ Resolved

SU22-Final-Q8



```
__m128i curr_v1 = vload(a+i);
__m128i curr_v2 = vload(b+i);
```

Why don't we cast when loading the values from a and b? I believe in project 4 we need to cast otherwise c outputs errors?

♡ ···

Anonymous Cheetah 2y #1702cbe

sorry just found it was asked before. we do need to cast

♡ ···

Anonymous Mink 2y #1702cbc ✓ Resolved

SP-22-Final-Q5.3 and 5.4 I don't understand why for pop we need a second writeback for regfile, but for push we didn't?

♡ ···

Erik Yang TA 2y #1702ced

push, we don't need a write back, because we are not returning a value into a register. Pop needs two writebacks because we are modifying sp (where it points to because we remove a value) and rd, the value that we pop

Anonymous Ant 2y #1702caf ✓ Resolved

SU22-Final-Q8

For line 11, I used vmul(tmp, mul) so that the positive values would get multiplied by 1 and the negative ones would get multiplied by 0. I don't see this on the alternate solutions. Is this a viable solution, and if not, why?

Rosalie Fang ADMIN 2y #1702cba

Close by not exactly. vcmpgt would actually return either all 0 or all 1's in binary. In this case, the positive values would then get multiplied by all 1's in binary, so -1. (We probably should've specified exactly what vcmpgt returns, in retrospect...)

Anonymous Ant 2y #1702cbb

I see, thanks!

Melanie McKune 2y #1702bfe ✓ Resolved

SP22-Final-Q2.2

How do we simplify from !B*!(CD) +CD to !B +CD by absorption? I tried applying DeMorgan's Law to !(CD) or using the Distributive Law to simplify (!(CD) + CD) = 1, but neither makes it the solution.

Rosalie Fang ADMIN 2y #1702bff

!B * !(CD) + CD

= !(B + CD) + !(!C + !D)

= ! [(B + CD) * (!C + !D)]

= ! [(B * (!C + !D)) +   (CD * (!C + !D))]

= ! [(B * (!C + !D)) +   (C * D * !C) + (C * D * !D)]

= ! (B * (!C + !D))

= !B + !(!C + !D)

= !B + CD

Anonymous Crane 2y #1702bfa ✓ Resolved

For cache questions in general would it be correct to use these equations:

Cache size = block size*num blocks

Cache size = #sets * associativity * block size block size = cache size/ cache lines

I'm trying to come up with different ways to calculate the same thing, I just want to make sure that my understanding of these relationships are correct?

♡ ···

Rosalie Fang ADMIN 2y #1702bfd

sure!

♡ 1 ···

Anonymous Ram 2y #1702bca ✓ Resolved

SU22-Final-Q8

1. For lines 7 and 8, I wrote &(a[i]) and &(b[i]). Is this correct? If this is wrong, what's wrong with my reasoning so I can better understand?

2. I wrote vstore(&result, sum_v); because b is a pointer but result is an array. Is this correct? If this is wrong, what's wrong with my reasoning so I can better understand?

Thanks!

♡ ···

Erik Yang TA 2y #1702bea

1. That's correct as well I think. But a + i is pointer notation, which is similar to what you have.

2. since it is initialized as result[4], **result** is inherently a pointer, so you don't need to do &result.

♡ ···

Anonymous Ram 2y #1702bbe ✓ Resolved

SU22-Final-Q7.3

1. When I did this question, I thought the TLB was shared across processes but not shared across programs. After reading the answers, I guess I'm wrong. To clarify, is the TLB not shared across processes and not shared across programs? Is the TLB shared across threads? Similarly, is the page table not shared across processes and not shared across programs? Is the page table shared across threads?

2. Since we were not given a page table, I didn't know if it was a page table hit or page fault. How were we supposed to know this?

3. To clarify, every time we context switch to a different process, the TLB is cleared but the page table is not cleared?

Thanks!

♡ ···

E

Erik Yang TA 2y #1702bed

1. TLB is flushed in context switches

2. You can assume it's empty at first, so when a process has never accessed a virtual page, it is a page fault.

3. yes

♡ 1 ···

Anonymous Ram 2y #1702dae

1. Is the TLB not shared across processes and not shared across programs?

2. Is the TLB shared across threads?

3. Similarly, is the page table not shared across processes and not shared across programs?

4. Is the page table shared across threads?

Thanks!

♡ ···

E

Erik Yang TA 2y #1702daf

↩ Replying to Anonymous Ram

The TLB is flushed after switching to another process, meaning that Process 1 will have no idea what the TLB of Process 2 is. Threads could share TLB. There's only 1 page table, but different processes could access different regions of that page table.

♡ 1 ···

Anonymous Ram 2y #1702dbb

↩ Replying to Erik Yang

Do different threads access the same region of the page table?

♡ ···

E

Erik Yang TA 2y #1702dbc

↩ Replying to Anonymous Ram

they could, it depends on what memory they are trying to access

♡ ···

Anonymous Ram 2y #1702bbd ✓ Resolved

SU22-Final-Q7.3

I wrote 17 bits because I thought there was 1 valid bit. How do we know whether to include the valid bit or not? Thanks!

♡ ···

E

Erik Yang TA 2y #1702beb

i guess this is something that will hopefully be clarified during the exam; if not, it is worth an ask

♡ 1 ···

Anonymous Ram 2y #1702bbc ✓ Resolved

SU22-Final-Q5.5

Could you explain what is going on? I completely don't understand and my answers are completely wrong :(

E.g. My answer for earliest is 8ns because 3+5ns. Don't we need to consider the delay for box 1?

Thanks!

E

Erik Yang TA 2y #1702bdf

#1702fe

Anonymous Ram 2y #1702bbb ✓ Resolved

SU22-Final-Q5.3

What is the difference between a multiplexor, demultiplexor and priority encoder?

Thanks!

E

Erik Yang TA 2y #1702bde

Mux: it takes in X inputs, and has a selector (in bits) that picks which one of the inputs to output.

Demux: takes in 1 input, and has a selector that picks where this input is going. If you have 4 outputs, and your selector is 3 (0-indexed), then the input will be stored in the fourth output.

Priority encoder: it takes in x inputs, and outputs the highest indexed input that equals 1. Ex: if you have 4 inputs, and index 0, 2, 3 are all 1, then the encoder will return index of 3.

Anonymous Ram 2y #1702baf ✓ Resolved

SU22-Final-Q4.2

1. Conceptual: How do you tell if a number is normal or denormal?

2. Conceptual: What representation are immediates? E.g. Are RISC-V immediates in decimal?

3. For li t4 0x4080 0000, how could we put a hex number as the immediate? I spent a long time thinking about how to convert this hex number into decimal?

Thanks!

E

Erik Yang TA 2y #1702bdd

1. #1702abc

2. immediates can either be decimal or hex

3. the immediate can either be hex or decimal, they're interchangable

Anonymous Shark 2y #1702bae ✓ Resolved

SU22-Final-Q8

In the "vstore(result, sum_v);" shouldn't we have to cast result to type (__mm128i *)?

♡ 1 ···

E Erik Yang TA 2y #1702bdc

yes, this is a typo

♡ ···

Anonymous Ram 2y #1702bad ✓ Resolved

SU22-Final-Q3.3

1. Conceptual: When we have the instruction j loop, does the PC go to line 10 or 11?

2. I wrote this:

auipc t0 0
jalr x0 t0 loop

Is this correct?

Thanks!

♡ ···

E Erik Yang TA 2y #1702bda

since loop is not actually a line of code, I believe PC will go to line 11

♡ ···

Anonymous Shark 2y #1702aff ✓ Resolved

SU22-Final-Q3.4

When we do "lb t0 3(ra)", wouldn't we be loading in the byte at the lowest address, which is the least significant byte in the instruction, since RISC-V is little endian? We should be loading in the most significant byte, since func7 is in the most significant byte of the instruction.

♡ ···

E Erik Yang TA 2y #1702bdb

since it is not specified, we can't assume anything about the question that it is little endian, so that's why the MSB would be the 0th byte

♡ ···

Anonymous Shark 2y #1702bef

How would we know that it is big endian in that case? The question doesn't specify, but the endianness would change the answer.

♡ ···

E Erik Yang TA 2y #1702bfc

Replying to Anonymous Shark

I think it’s fine to assume big endian in most case, but this is a clarification worth asking.

♡ 1 ···

Anonymous Shark 2y #1702afc ✓ Resolved

SU22-Final-Q2

I had a few questions for the create_file function part of this question. First, why do we have to use calloc instead of malloc for the create_file function, and why do we cast the result of calloc to type (file_item*)? Shouldn't the variable declaration already handle this? Also, I don't understand what the solution means when it says "we can directly set the name field because we’re just setting the field’s value to the pointer value passed in from the const char *name argument." Aren't all strings just a pointer to the first character of the string?

Erik Yang TA 2y #1702bce

You could use malloc for this question, that was an option. I don't think you absolutely *need* to cast it, but it is good to include this at least to show understanding.

Since we're given a pointer to name, we can just directly assign that variable. The struct has a field of name, so we're just giving it that attribute.

Anonymous Vulture 2y #1702ccf

But why can't we also do that for contents. Can't the name and contents both be freed at some point?

Nvm, it's because name is type const char* which means it can't be freed

Anonymous Chimpanzee 2y #1702afb ✓ Resolved

SU22-Final-Q3.4

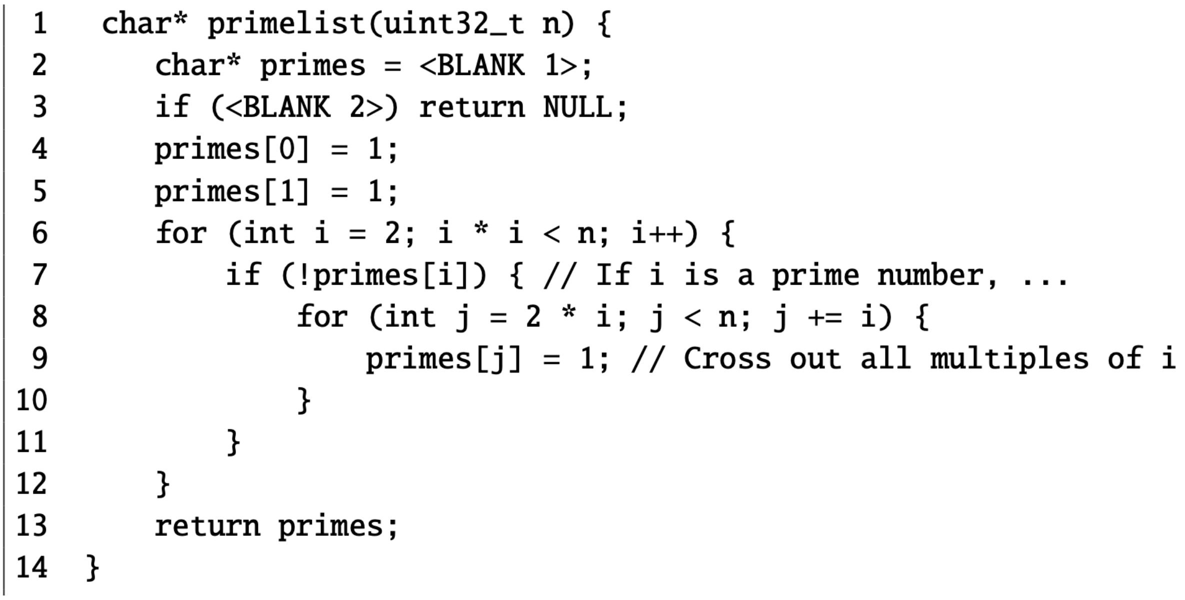
I don't understand how line 8 works, how do we know `3(ra)` is referring to the add instruction above it.

Erik Yang TA 2y #1702bec

ra is the address of the add instruction because **jal ra** sets ra to be the next line after that instruction. And the only byte that differs between sub and add instruction is the 3rd byte from the MSB, so that's why we have lb 3(ra)

Anonymous Penguin 2y #1702afa ✓ Resolved

SP22-Final-Q6.1



```
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;
}
```

For black 1, we are allocate memory for a char pointer, so we need a Null at the end, right? Why the answer is calloc(n, sizeof(char)), NOT calloc(n + 1, sizeof(char))?

For black 2, we just allocate memory to primes, how could primes be NULL?

♡ 1

Erik Yang TA 2y #1702bcd

blank 1: good question, since these are 1 byte integers, we don't need a null terminator. The terminator only applies to strings.

blank 2: not every malloc/calloc intialization will succeed, so this is the error checking casefor that. Remember in project 1(also lab), we also wanted you to see if malloc/calloc failed, then we return null.

Anonymous Ibex 2y #1702caa

But since primes is a char array, wouldn't it be treated like a string in C?

Erik Yang TA 2y #1702cac

Replying to Anonymous Ibex

not necessarily, since one char = 1 byte, so it could also be a 1 byte integer array. In this problem specifically, we're working with integers, so we can't assume that primes is a string array

Anonymous Ibex 2y #1702cad

Replying to Erik Yang

Oh, so even primes is initialized as char*, you can assign ints to it? I thought the compiler would raise an error/warning.

Erik Yang TA 2y #1702cae

Replying to Anonymous Ibex

you can assign an int to a char only because we are working with 1 byte integers

Anonymous Gull 2y #1702add ✓ Resolved

SU22-Final-Q8

On line 15, would we need to cast result to __m128i * within the vstore call?

Also, would this work for line 15: result = (int32_t *) sum_v

Erik Yang TA 2y #1702ade

i think we do need to cast. That would not work because sum_v is a vector and result is an int array so you have to use vstore

Anonymous Cobra 2y #1702abb ✓ Resolved

SU22 final 7.4

How the data in TLB change when switching between different process? Remain the same?

Erik Yang TA 2y #1702acc

in the problem statement, it says that it will context switch between processes, meaning that the TLB entries are flushed

Anonymous Lyrebird 2y #1702afd

Just a follow up question, how are the two process distinguish from each other in TLB? I suppose if not then we would have two of the same VPN corresponding to different PPN for the first two memory access, which breaks the rule right?

Erik Yang TA 2y #1702afe

Replying to Anonymous Lyrebird

context switches mean that when the processes switch, the TLB gets flushed, so that process 2 has no memory of process 1's progress in TLB

♡ 1

Anonymous Vulture 2y #1702dbd

Replying to Erik Yang

Is the TLB the only thing that gets flushed/erased during a context switch, what else?

Erik Yang TA 2y #1702dbe

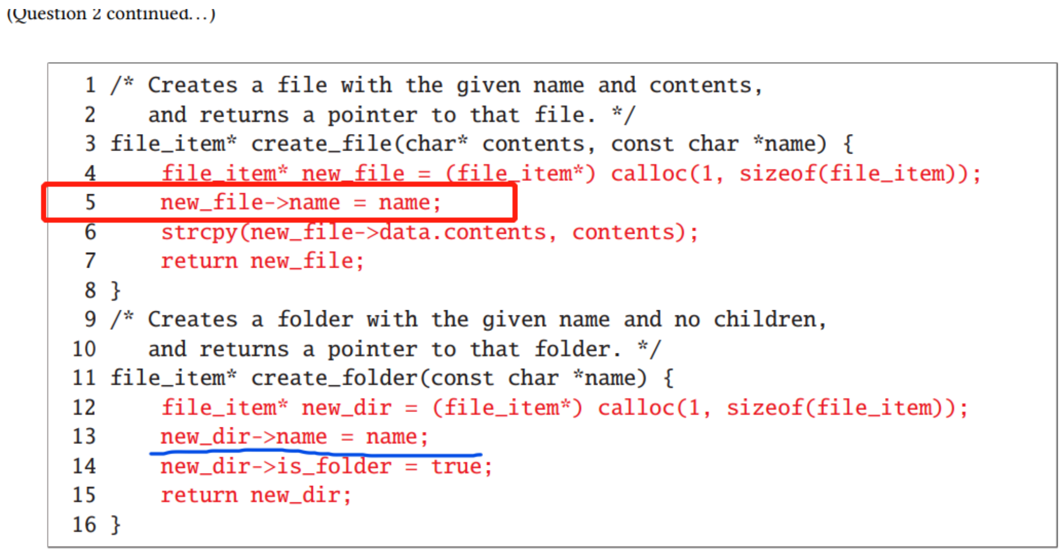
Replying to Anonymous Vulture

I think it's just TLB, other things have their state preserved during switching

Anonymous Hamster 2y #1702aae ✓ Resolved

su22 final Q2

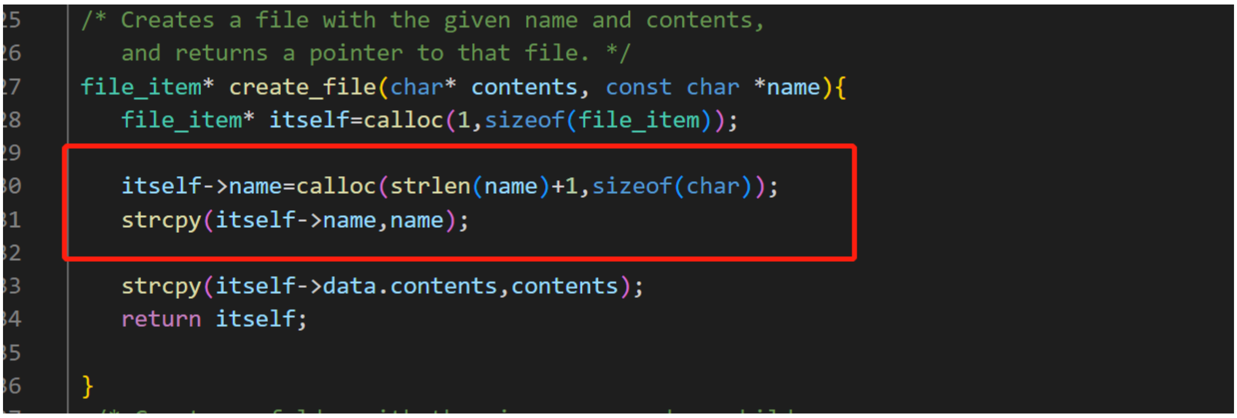
Line 5. Why don't we allocate new space for const char* name? I think our code should be robust and behave consistently even if string name is later freed.



(Question 2 continued...)

```
 1 /* Creates a file with the given name and contents,
 2    and returns a pointer to that file. */
 3 file_item* create_file(char* contents, const char *name) {
 4     file_item* new_file = (file_item*) calloc(1, sizeof(file_item));
 5     new_file->name = name;
 6     strcpy(new_file->data.contents, contents);
 7     return new_file;
 8 }
 9 /* Creates a folder with the given name and no children,
10    and returns a pointer to that folder. */
11 file_item* create_folder(const char *name) {
12     file_item* new_dir = (file_item*) calloc(1, sizeof(file_item));
13     new_dir->name = name;
14     new_dir->is_folder = true;
15     return new_dir;
16 }
```

My answer is:



```
/* 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* itself=calloc(1,sizeof(file_item));

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

   strcpy(itself->data.contents,contents);
   return itself;

}
```

Is my answer fully correct?

Erik Yang TA 2y #1702acb

the name and contents are already given, so you don't need to allocate memory for that

Anonymous Hamster 2y #1702acd

I mean what's the rubric of this question? The solution says there could be alternative answers, including additional memory allocation, which is shown in the Gradescope. But I have no access to that year's GS, so could you let me see the complete rubric (or solution) of this question?

Erik Yang TA 2y #1702ace

Replying to Anonymous Hamster

i don't think your soln in the red box would apply to the rubric. We're just setting the name variable to the name passed in from the parameters

Anonymous Hamster 2y #1702acf

Replying to Erik Yang

But what if the name string is later freed? If we do not make another copy, the file_item may result in pointing to some garbage.

Erik Yang TA 2y #1702ada

Replying to Anonymous Hamster

name doesn't necessarily need to be on the heap, it could be on the stack

Anonymous Hamster 2y #1702adb

Replying to Erik Yang

But if it's on stack it's more likely to be garbage later, when the function, whatever it is, returns.

Erik Yang TA 2y #1702adc

Replying to Anonymous Hamster

since new_file is on the heap, it will keep a pointer to the struct that will contain name

Erik Yang TA 2y #1702bcf

Replying to Erik Yang

i'm sorry I actually think your answer is correct because you are strcpy it into the field. Sorry about that !

Anonymous Hamster 2y #1702aac ✓ Resolved

Su22 Final Q3.3
The question asks us to replace j Loop with other instructions using jalr,

The solution is
la t0 Loop
jr t0
My answer is
addi t0 ra -16
jr t0
I used ra since it pointed to line 14, 4 lines below Loop tag.
Is my answer correct?

Q3 Error-Introducing Code (20 points)

In machine learning, some data scientists add random noise to a training dataset in order to improve their models. Here, we will take a dataset and transform it into usable data in RISC-V!

The function `jitter` is defined as follows:

- Inputs:
  - `a0` holds a pointer to an integer array.
  - `a1` holds a buffer large enough for n integers (which does not overlap with the array in `a0`).
  - `a2` holds n, the length of the arrays.
- Output:
  - `a0` holds a pointer to the buffer originally in `a1`. The buffer is filled with the result of calling `noisen` on each element in the `a0` array.

The function `noisen` is defined as follows:

- Input: `a0` holds an integer.
- Output: `a0` returns the integer with noise added.

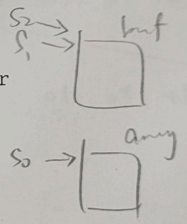
Eric has provided the correct implementation of `jitter` below, following calling convention:

```
jitter:
  # BEGIN PROLOGUE  =20
  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  20
  ret
```

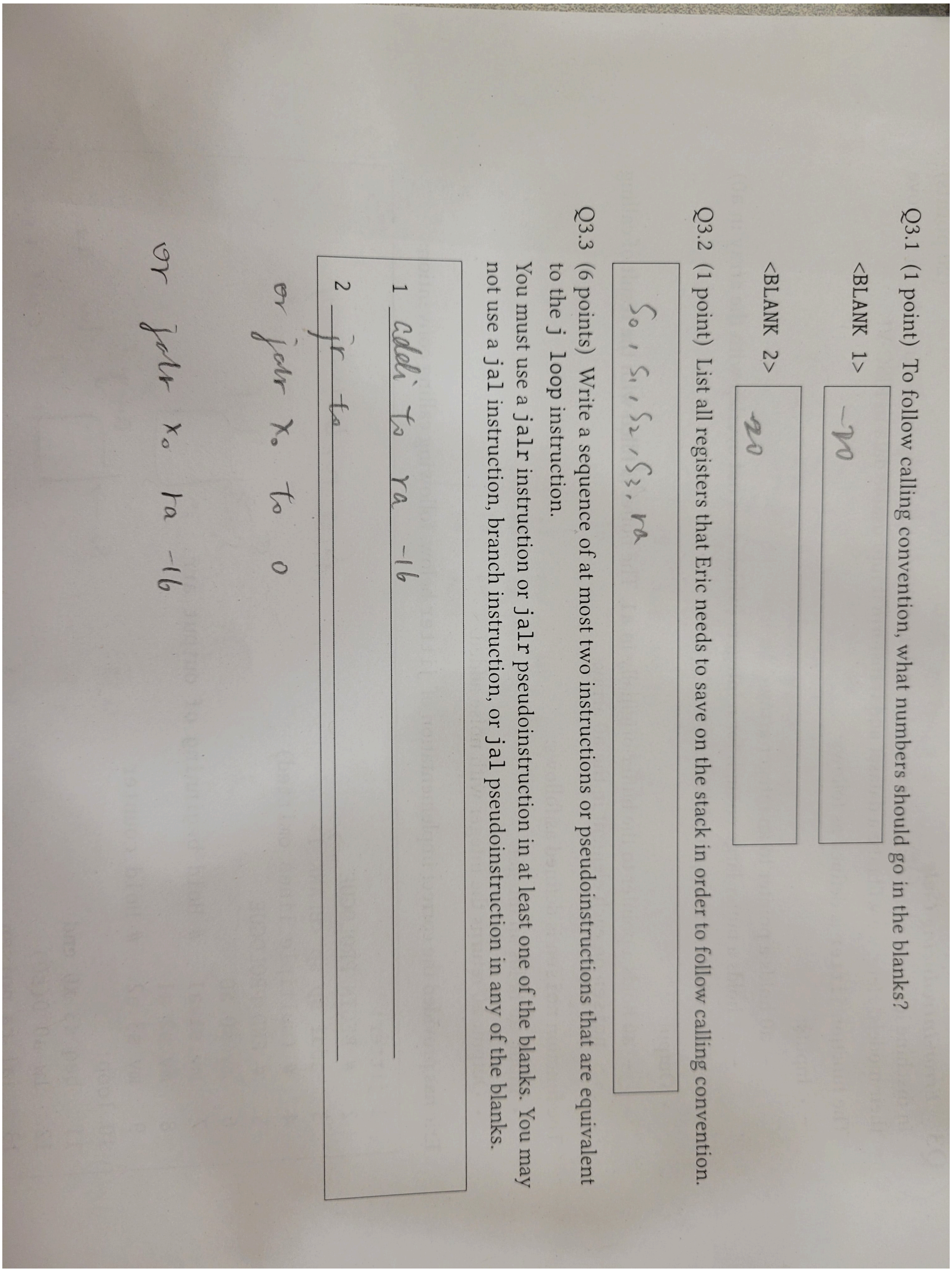
ra-16

ra →

S3 = n, length of array



lw a0 a(s0)
call noisen.
sw a0 0(s1)



Q3.1 (1 point) To follow calling convention, what numbers should go in the blanks?

`<BLANK 1>` -20

`<BLANK 2>` 20

Q3.2 (1 point) List all registers that Eric needs to save on the stack in order to follow calling convention.

s0, s1, s2, s3, ra

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.

1 addi t0 ra -16

2 jr t0

or jalr x0 t0 0

or jalr x0 ra -16

Anonymous Rabbit 2y #1702aaf

Same question. How about

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

Rosalie Fang ADMIN 2y #1702aeb

Looks good

Erik Yang TA 2y #1702bcc

This one looks good

Rosalie Fang ADMIN 2y #1702aea

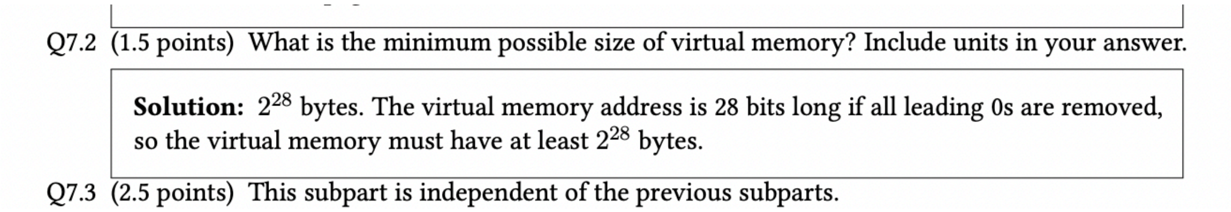
Very close!! Remember that loop doesn't actually take up instruction memory, it's just a label, so I think it'd be -12.

Anonymous Hamster 2y #1702aee

Oops, yes. Thx!

Anonymous Lyrebird 2y #1702aab ✓ Resolved

Could someone explain the methodology of why minimum size is 2^#of bits. What would the max be. In the dicussion the formula was number virtual pages x page size in bytes.



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.

Q7.3 (2.5 points) This subpart is independent of the previous subparts.

Rosalie Fang ADMIN 2y #1702aec

These are the exact same formulas. number of virtual pages is 2^(# bits used for VPN) and page size in bytes is 2^(# bits used for offset), and # VPN bits + # offset bits = # virtual address bits.

Anonymous Marten 2y #1702aaa ✓ Resolved

Just to confirm:

whether fully-associated caches don't have conflict miss?

♡ 1

Anonymous Hamster 2y #1702aad

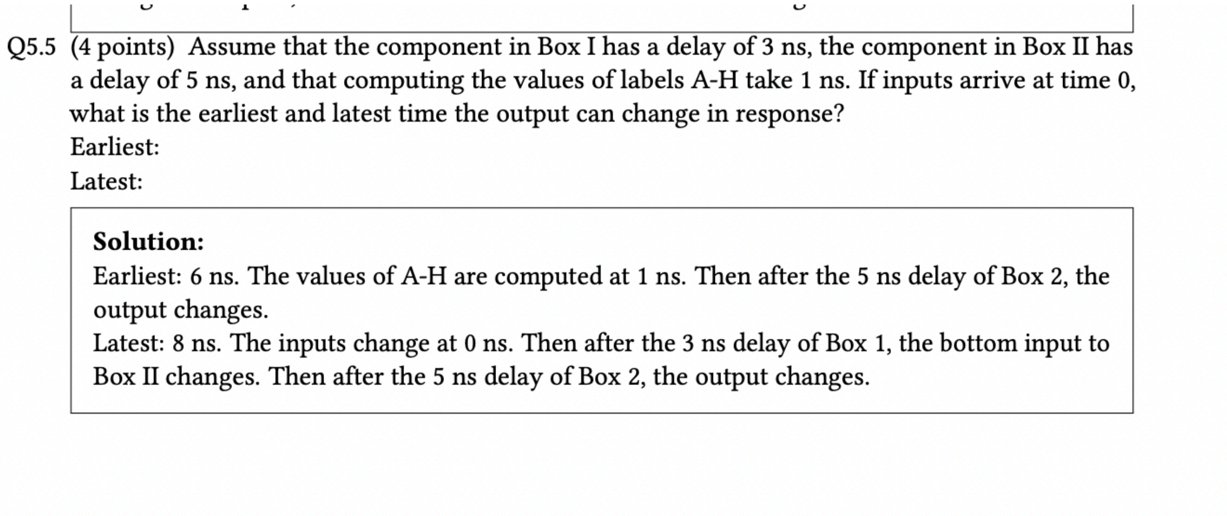
I think yes

Erik Yang TA 2y #1702abd

yes

Anonymous Lyrebird 2y #1702fe ✓ Resolved

SU 22 5.5. Could someone better explain this? For earliest, why are we not considering the delay for box 1. For latest, why are we not considering the 1ns delay for AH



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.

♡ 1 ···

E Erik Yang TA 2y #1702abe

There's two possibilities with which the output can change: if any value of the labels A-H change, or if the output of Box 1 changes. The earliest would be if Box1 remains the same and only the value of the labels change. The latest would be if Box 1 changes.

♡ 1 ···

Anonymous Shark 2y #1702baa

But even if Box 1 changes, wouldn't Box 2 still output a new value at 6ns, since it still received the new A-H inputs at 1ns?

♡ ···

E Erik Yang TA 2y #1702bcb

Replying to Anonymous Shark

If box 1 changes, the possibility of a new set of a-h inputs is still possible but this would happen at the same time as box1 changing, so the latest would be max(box1 changing, inputs changing) plus box2 delay

♡ 1 ···

Anonymous Shark 2y #1702bee

Replying to Erik Yang

Right, but what I meant to say was that if Box 1 and a-h both change, then a-h would change at 1ns and Box 2 would start computing a new value starting from that 1ns, then after another 5ns Box 2 would then output that value, so the output would still change at 6ns regardless of whether or not Box 1 changes.

♡ ···

E Erik Yang TA 2y #1702bfb

Replying to Anonymous Shark

If box 1 changes, it takes 3 ns before box2 can start its delay so you must wait 1 or 3 ns before box2 can start working with its inputs

♡ 2 ···

Anonymous Lyrebird 2y #1702fc ✓ Resolved

In general for floating point questions, when we are asked for the smallest or largest number, does that imply the smallest or largest normal number. Or do we also consider denormal numbers.

Erik Yang TA 2y #1702abc

smallest positive number usually means denorm, largest would be norm. Think of denorm as really small decimal numbers

Anonymous Crane 2y #1702ee ✓ Resolved

SU22-Final-Q5.5

I understand how the latest one gets calculated (sums all the largest delays). But I'm not really sure why we can completely disregard the Box 1's delay

Jero Wang ADMIN 2y #1702fb

For the earliest change, there's a shorter critical path from A-H, through Box II, to the output. The fastest path involving box A would be 8ns.

Anonymous Marten 2y #1702ea ✓ Resolved

I know that 2022 exam problems are the latest and more important.

But why no one is answering the problems before 2022? (T T)

Erik Yang TA 2y #1702eb

sorry i'll start looking at them

Anonymous Goose 2y #1702ce ✓ Resolved

sp22-final-q6.3: why is there one compulsory miss for every 128-byte block? I don't get why primes[4] is a miss (why aren't primes[2], primes[3] a miss?)

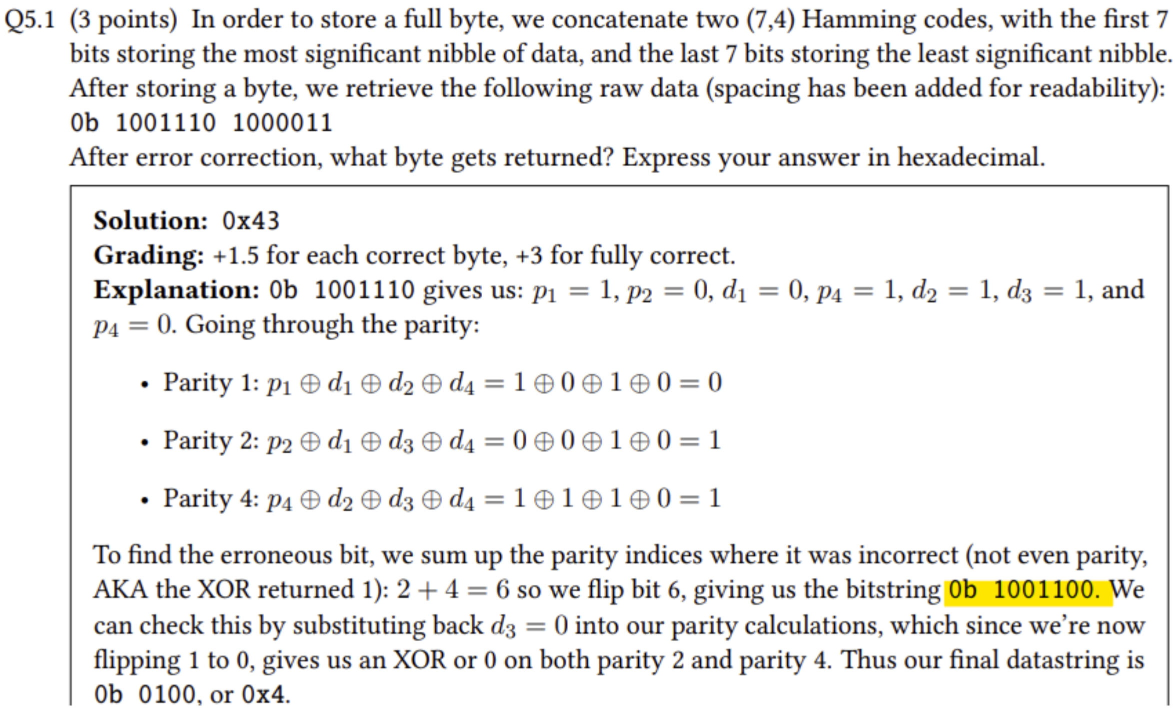
Jero Wang ADMIN 2y #1702df

I think the explanation is slightly incorrect, the compulsory miss happens when we set `primes[0] = 1`, not when we access `primes[4]`.

Anonymous Mandrill 2y #1702cd ✓ Resolved

SU22-Final-Q5

I understand how we got to the highlighted bitstring, but how does this give us 0b0100?



Q5.1 (3 points) In order to store a full byte, we concatenate two (7,4) Hamming codes, with the first 7 bits storing the most significant nibble of data, and the last 7 bits storing the least significant nibble. After storing a byte, we retrieve the following raw data (spacing has been added for readability):
0b 1001110 1000011
After error correction, what byte gets returned? Express your answer in hexadecimal.

Solution: 0x43
Grading: +1.5 for each correct byte, +3 for fully correct.
Explanation: 0b 1001110 gives us: $p_1 = 1$, $p_2 = 0$, $d_1 = 0$, $p_4 = 1$, $d_2 = 1$, $d_3 = 1$, and $p_4 = 0$. Going through the parity:

- Parity 1: $p_1 \oplus d_1 \oplus d_2 \oplus d_4 = 1 \oplus 0 \oplus 1 \oplus 0 = 0$
- Parity 2: $p_2 \oplus d_1 \oplus d_3 \oplus d_4 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$
- Parity 4: $p_4 \oplus d_2 \oplus d_3 \oplus d_4 = 1 \oplus 1 \oplus 1 \oplus 0 = 1$

To find the erroneous bit, we sum up the parity indices where it was incorrect (not even parity, AKA the XOR returned 1): $2 + 4 = 6$ so we flip bit 6, giving us the bitstring 0b 1001100. We can check this by substituting back $d_3 = 0$ into our parity calculations, which since we're now flipping 1 to 0, gives us an XOR or 0 on both parity 2 and parity 4. Thus our final datastring is 0b 0100, or 0x4.

Jero Wang ADMIN 2y #1702de

In 7,4 Hamming code, there are 4 data bits at bits 3, 5, 6, and 7 (1-indexed, from the left), which give us 0b0100.

Anonymous Chamois 2y #1702ff

Will we be given the bit pattern (ie. what index each data or parity bit is) for the type of Hamming Code we receive or should we be writing down the different type of Hamming Codes on our cheatsheet

♡ 1

Erik Yang TA 2y #1702cfb

Replying to Anonymous Chamois

the reference card will have a Hamming ECC chart.

Anonymous Chamois 2y #1702daa

Replying to Erik Yang

When will the reference sheet on the website be updated with the one that will be used during exams?

Anonymous Crane 2y #1702cb ✓ Resolved

SU22-Final-Q7.1

For this question, would the shared bits always be up to the number of bits that are the same? Like for example if my Virtual memory address happened to be 0x12345067 and my Physical memory address happened to be 0x234167 then would the maximum page size be 2^8?

♡ 1

Anonymous Mandrill 2y #1702cf

following!

♡ ···

Jero Wang ADMIN 2y #1702dd

Yes, maximum would be 2^8 here.

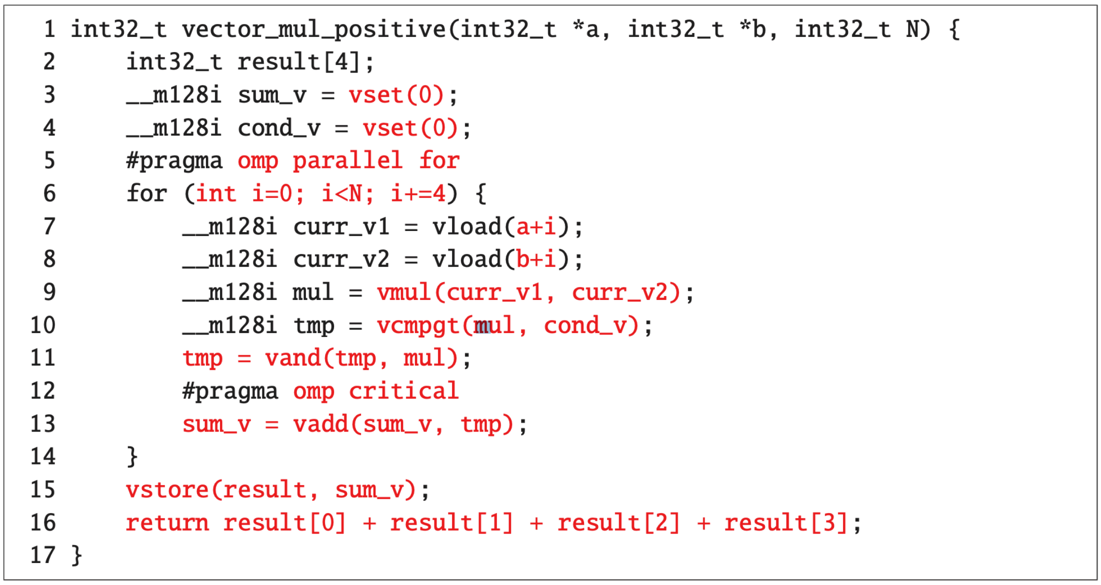
Shared bits here just mean the bits are the same, and tells you what the maximum page size is.

♡ 1 ···

Anonymous Crane 2y #1702bc ✓ Resolved

SU22-Final-Q8

Why do we need line 11? I have the same solution as the one below but I assumed line 10 would be equal to something like tmp = [0, 12, 0, 32] if we were to use a = [-1, 2, 3,4] b=[5,6,-7,8] n= 4.



```
 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 }
```

♡ ···

Erik Yang TA 2y #1702bd

line 10 and line 11 are important because we only want to multiply the positive numbers, so you first have to compare to a vector of all 0's to see if it's positive. And then **and** the result of that to see which numbers to add to sum

♡ ···

Anonymous Crane 2y #1702bf

If after line 10 tmp is [0,12,0,32] then we can just and tmp to sum without anding it right? Because our neg values are already zeroed out? Unless I'm misunderstanding what tmp returns?

♡ ···

Erik Yang TA 2y #1702ca

Replying to Anonymous Crane

Compare returns 1s and 0s so you need to have the and in order to get the right values

♡ 2 ···

Anonymous Lark 2y #1702bb ✓ Resolved

SU22-Final-Q7.1

Why is 13 bits the max page size? How does going higher mess up the offset between VPN and PPN?

Erik Yang TA 2y #1702be

The shared bits between the two are the offset and these bits are determined by the page size.

Anonymous Chamois 2y #1702ccd

Why is it not 12 bits? VA is 0xABCD123, and PA is 0x123123. So the shared bits are 123, which is 12 bits.

Erik Yang TA 2y #1702cce

Replying to Anonymous Chamois

This is in hex so D and 3 in hex match in the first bit. 12+1 bit shared is 13

Anonymous Lark 2y #1702ac ✓ Resolved

SU22-Final-Q8

Why don't we need to cast a and b to __m128i*?

Erik Yang TA 2y #1702ad

i think you're right, we do need to cast

Anonymous Goose 2y #1702aa ✓ Resolved

su22-final-Q7:

7.1: How do we know that virtual memory address and the physical memory address share the lower 13 bits and how does this help to determine the max page size?

7.3: how is the bits for physical memory and offset calculated?

♡ 2

Anonymous Lark 2y #1702ab

For 7.3, a 4 KiB page size = 2^12 bytes which means we need 12 bits for the page offset. Since the total amount of bits for the physical address is 24 bits, we have 24-12 = 12 remaining for the PPN.

Anonymous Mandrill 2y #1702da

following!

Jero Wang ADMIN 2y #1702dc

7.1: If you write out the physical address and the virtual address, you can see that their lowest 13 bits match. Then, imagine if page size was 2^14 bytes, the two addresses would then have different offsets within a page.

Anonymous Lark 2y #1702f ✓ Resolved

SU22-Final-Q5.4

What are the tags A-H supposed to be? I get that Box II is a mux but I don't understand how it outputs the correct nibble at the end.

Erik Yang TA 2y #1702ae

these represent the parity bits 0-7

Anonymous Lark 2y #1702ba

What exactly do you mean by parity bits 0-7?

If the raw data is something like `0b1001110`, the result after merging from the XOR gate should be `110`. If A-H are parity bits, how does the mux output the correct nibble `0b0100` ?

Jero Wang ADMIN 2y #1702db

Replying to Anonymous Lark

I think that should be the parity bits plus the data bits.

Anonymous Lark 2y #1702ec

Replying to Jero Wang

Sorry, I'm still confused about how it works. Using the example above, how does `0b0100` end up getting produced?

Jero Wang ADMIN 2y #1702ef

Replying to Anonymous Lark

#1702de

Anonymous Lark 2y #1702fd

Replying to Jero Wang

Sorry, I don't think I'm making my question clear. I understand how mathematically the nibble is produced, but I don't see how the mux is doing that in the circuit. If A-H is the 7 bits, and the result from the XOR gate is 110, doesn't the 6th bit get produced by the mux?

Rosalie Fang ADMIN 2y #1702aed

Replying to Anonymous Lark

Sorry I think the previous responses were confused. A-H should produce a nibble of data each. A would correspond to p1 being the error-ed bit, B would correspond to p2 being the error-ed bit, C would correspond to d1 (index 3) being the error-ed bit, so on.

So if the result from the XOR gate is 110, then that means the 6th bit is wrong, which is d3, so we want to flip d3 and take the data bit, 0110 -> flip d3 = 0100, which should be the value of F.

Anonymous Lark 2y #1702bab

Replying to Rosalie Fang

Why are there 8 tags for 7 bits tho?

Also what does parity bit 0 in the solution mean? I thought it parity bits were 1, 2, 4, etc.

Anonymous Ram 2y #1702bba

Replying to Rosalie Fang

Are there 8 tags for 7 bits because one tag represents no error? If so, which tag represents no error?

Anonymous Crane 2y #1702cab

Replying to Anonymous Ram

The tags correspond to which bit in the input has an error, or the "parity bit value" in the solution, aka the addition of the indices of parity bits with errors. A = no error, so output received data bits. B = error at bit 1, which is parity bit 1, C = error at bit 2 / p2, E = error at bit 4 / p4, which all correspond to the data being intact and sending the received data bits.

Anonymous Lark 2y #1702e ✓ Resolved

SU22-Final-Q3.4

How does changing the instruction from an add to a sub after storing the sum into s0 change what is in s0? Also, how does this guarantee that we alternate between adding and subtracting randomBool()? It seems like we're always subtracting.

Erik Yang TA 2y #1702af

jal ra randombool sets ra to be equal to the instruction in line 7, so modifying ra modifies the instruction in that line, therefore changing s0.

Imagine it changes the add to sub the intial time. Now, it will say sub s0 s0 a0, so the next time you call this function, it will start off with the sub. This is just the initial state of the code.

Anonymous Rabbit 2y #1702aba

I don't think the solution is convincing. How can we claim that we are fetching IMEM at line 7 and how can we write to IMEM at line10, since we are read/write to DMEM by default. Moreover, the distribution of the generated random number is not uniform as we flip the bit every time. So that we can never return two consecutive `a0-1`.

Anonymous Lark 2y #1702a ✓ Resolved

SU22-Final-Q2

Sorry I kinda forgot a lot of C. Why don't we have to malloc stuff out for things inside file_item? Is it because the size of file_item calculates the size of file_item_data which is 64 bytes? And do we just assume contents will be less than 64 bytes?

Also what is the difference between char* [16] and char [16]? Thank you.

Erik Yang TA 2y #1702b

Yup the malloc just gets the sizeof(struct) which includes all the elements inside the struct. The difference between the two is that one is a char array of pointers while the other is a char array

For this question, we don't want to exceed 64 bytes because if we have 16 pointers, that is 16*4 bytes. That means the length of the string must be 63 bc of the null terminatie

Anonymous Lark 2y #1702c

Gotcha thanks. And just to double check, strcpy takes in 2 char* and contents is a char array. So a char* is essentially a char array which ends until it hits a null terminator?

Also, is it ok that we don't do anything with the children array? Will it just stay null? If we had malloced instead of calloced would we have to manually null the array?

Erik Yang TA 2y #1702d

Replying to Anonymous Lark

Yup. If we don't do anything with children array, it's just a pointer to uninitialized data if we malloc. If we calloc, it will be a pointer to zeroes. So if you want an entirely null array if you malloc, then yes manual

Anonymous Albatross 2y #1702adf

Replying to Erik Yang

As a follow-up question, when do we have to allocate space the fields in the struct? If we were to malloc file_item instead of calloc, would we have to manually malloc the file_item_data? I think I'm just a bit confused because I remember for projects we had to malloc some struct fields individually... Also, if the parameter for the create_file function was a char array instead of a string literal, do we have to use strcpy?

Erik Yang TA 2y #1702aef

Replying to Anonymous Albatross

We malloc some struct fields invidividually if they occupy space in memory, and if they are not given already. For example, in project 1 you have to allocate space for the board, which was not passed in as a parameter, and because we want to access the state of the board after the function, we need to allocate space in memory for it. As a rule of thumb, everything in the stack is wiped after the function ends, so if you need something after the function call, and it hasn't been initialized already, then you probably need to malloc space for it

Anonymous Ram 2y #1702bac

Replying to Erik Yang

As a follow-up question, why do we not malloc space for new_file->data in this question? Since everything in the stack is wiped after the function ends, don't we want new_file->data to persist? Thanks!

Erik Yang TA 2y #1702bbf

Replying to Anonymous Ram

New file->data is part of the malloced space of new file so that will persist

This comment was deleted

Jero Wang ADMIN 2y #1702fa

#1702df

Please check the other posts before posting to avoid duplicates.

This comment was deleted

Anonymous Quelea 2y #1702def

72/2400 = 0.03

Anonymous Herring 2y #1702dfa

thank you sm! i spent so long looking at that...