[Final] Past Exams - 2021 #1701

J Jero Wang ADMIN
2 years ago in Exam – Final

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 2021 final walkthrough

Anonymous Dugong 2y #1701dfc ✓ Resolved

SP21-Final-9.3

Why do we need to wire x0 as rs2 rather than just adding it as a second argument when calling the pseudoinstruction? I thought the latter is what we usually do with pseudoinstructions

♡ ···

Anonymous Gazelle 2y #1701eab

I think we're wiring rs2 to be x0 since is_null only takes rd and rs1.

♡ 1 ···

Anonymous Quail 2y #1701def ✓ Resolved

FA21-Final-Q5.2 + 5.3,

can someone walk me through why we need a third rs3 input? and also for 5.3 why would the staff solution change the mul work? wouldn't a0 depend on the previous line's data to be write back?

♡ ···

Anonymous Gazelle 2y #1701dfb

We don't need a third rs3 input.

For 5.3, mul a0 a1 a1 works because the register that gets written back to is a0. In the second line of mul a0 a1 a1, it just overwrites the previous a0 value (with the same value since the operation is identical) and only needs to read a1, which wasn't changed by the first line. So there is no dependency with a mul operation.

♡ 1 ···

Anonymous Quail 2y #1701dfe

sorry i meant third rs3 output

♡ ...

Anonymous Quail 2y #1701dff

i think i get it, do we need a third one because we need to perform arithmetic on using the data of all three registers, not just two?

♡ ...

Anonymous Gazelle 2y #1701eaa

Replying to Anonymous Quail

Yup exactly!

♡ ...

Anonymous Caribou 2y #1701dbb Unresolved

B. Calculate all absolute addresses.

- ☐ Compiler
- ☐ Assembler
- ■ Linker
- ☐ Loader

SP21 Q3V2B, why does linker calculate all absolute addresses? Some dynamically-linked libraries are unknown when we run linker.

♡ ...

Anonymous Ibis 2y #1701dab ✓ Resolved

Sp 21 2

Could someone explain how they got 100. Is there a specific formula for MTTF

ii. Q2B

A. Your company would like to restrict the annualized failure rate to be 1% for the individual machines in a large cluster. What does the Mean Time To Failure (MTTF) have to be to satisfy this annualized failure rate? Assume that the MTTF in this question is unrelated to that of part a. Write down your answer in years.

100

♡ ...

E Erik Yang TA 2y #1701dad

#1701bbc

♡ ...

D Deniz Demirtas 2y #1701cfb ✓ Resolved

FA21-Final-Q1.9

What would be the answer if the question asked the highest possible value for y, and what would be the explanation for it? Thanks!

♡ ···

R Rosalie Fang ADMIN 2y #1701ded

Homework 7.5

♡ ···

Anonymous Peafowl 2y #1701cee ✓ Resolved

FA21-Final-Q1.9

I don't understand the answer. How can Thread 1 wake up and write y=1? If it read y=0 (line 1) and went to sleep, the next time it wakes up it should read line 2 x = 0 (since thread 2 finished executing). Where am I going wrong? Thanks!

♡ ···

R Rosalie Fang ADMIN 2y #1701dec

When thread 1 goes to sleep, all it knows is that it's in the middle of execution. since it was still on line 1, it will not know to check x = 0 before writing y. It will in fact wake up, write y = 1, read x = 0, write x = -1, and then check to see if x > 0 and exit.

♡ ···

Anonymous Peafowl 2y #1701ced ✓ Resolved

FA21-Final-Q4.2

I wrote the smallest combinational logic delay is 0ps because on the left and right hand side of the circuits are 0 tunnel so the register on the right is connected to the register on the left with no delay. Why is this wrong? Thanks!

♡ ···

R Rosalie Fang ADMIN 2y #1701deb

The tunnels on the left are not state elements (they don't depend on time). The shortest combinational logic delay needs to go from register to register.

♡ ···

Anonymous Peafowl 2y #1701cea ✓ Resolved

FA21-Final-Q9

1. For blank 6, is it possible to write total += isequal? Why not?

2. Could you explain why output[0]+output[1]+output[2]+output[3] == 0 is incorrect? The answer says that we could have received outputs that happened to add to 0, even if they aren't all 0 but I don't see how this is possible given that we could have only received outputs that are either 0 or 1.

Thanks!

♡ 2 ···

R Rosalie Fang ADMIN 2y #1701dea

1. total is a vector. You can't do normal arithmetic on it.
2. Please read the example it gave. There could be an overflow error.

♡ ···

Anonymous Panther 2y #1701cdd ✓ Resolved

SP21-Final-Q1 2F

Can someone explain what access patterns are? The solution says that since we only have 2 nested for loops, we have 2 different access patterns which is less than the 4 in our 4-way associative cache, so nothing gets evicted.

Would something get evicted if we had like 5 loops nested within each other?

♡ ···

R Rosalie Fang ADMIN 2y #1701ddf

Uhhh it really depends on the loop. Here I would really recommend to write out exactly what values are accessed, and you should be able to find a pattern see which set each block lands.

♡ ···

Anonymous Panther 2y #1701cdc ✓ Resolved

SP21-Final-Q1 2F

So, in the answer solution it says that no block is evicted. How do we know this and how would I be able to tell if a block should be evicted in problems like this where the code is given?

♡ ···

R Rosalie Fang ADMIN 2y #1701dde

Write out the cache and try to go access by access - try to see where each block lands within the cache and find a pattern.

♡ ···

Anonymous Peafowl 2y #1701cdb ✓ Resolved

SP21-Final-Q7 (a)(i)(B)

Could you explain the answer? I put none of the above because I thought that the ALU could handle the is_null rd rs1 operation and input the data to rd as per normal. If anything, a new ALU operation is needed but that was not given as an option. Thanks!

♡ ···

R Rosalie Fang ADMIN 2y #1701ddd

the branch comparator already does the comparison for is_null, so we don't need a new ALU operation be can safely use BrEq as another input to WBMux.

♡ ···

Anonymous Jay 2y #1701ccc ✓ Resolved

FA-Final-6.1

Would Not W | Not Y be a valid answer that would get me full points?

♡ ···

R Rosalie Fang ADMIN 2y #1701ddc

Close to full points. Similar to the midterm, this can be simplified more to be ~(W&Y) which uses 1 less Boolean logic operator

♡ ···

Anonymous Falcon 2y #1701cca ✓ Resolved

FA22-FINAL-Q2.2

How does the mantissa round down ?

Q2.2 (3.5 points) 1/3 (whose binary representation is 0b0.0101 0101…)

Solution: Answer: $21 * 2^{-6} = 0.328125$. We can move the binary point to get our floating point representation $0b1.010101... * 2^{-2}$. The mantissa rounds down, so our float would be $0b1.0101 * 2^{-2}$, or $0b10101 * 2^{-6}$ or $21 * 2^{-6}$

♡ ···

R Rosalie Fang ADMIN 2y #1701ddb

"Compute the value precisely, then round to the nearest floating point number." Here, the nearest floating point number means rounding down.

♡ ···

Anonymous Jay 2y #1701cbd ✓ Resolved

FA 21 Final Q 4.4

clk-q-delay = 3

largest combinatorial delay = 16

Based on question Q 4.3, the solution for the hold time is 15 so shouldn't it be 15 instead of 6 as the 6 is the setup time constraint?

Is the formula supposed to be clk-q-delay + largest combinatorial delay + setup time instead?

Solution: 25 ps

Shortest clock period = clock-to-Q delay + largest combinatorial delay + hold time = 6+16+3 = 25 ps

♡ ···

R Rosalie Fang ADMIN 2y #1701dda

Sorry this formula is wrong. shortest clock period = clk-to-q + largest combinational delay + setup time

♡ ···

Anonymous Jay 2y #1701cad ✓ Resolved

FA-21-Final Q 8.9

Q8.9 (1 point) Program 1: 0x12345664

Solution: This is a hit on program 1's version of this page, so we reuse the same page number from before. We get the physical address 0x01664.

Is the solution a typo? In program 1, there is no other version of this; thus would this be 0x05664?

♡ ···

Anonymous Stingray 2y #1701cbe

This should be fine. Look at Q8.5 where we had the Virtual Address 0x12345678; we assigned the VPN 0x12345 to the PPN 0x01

♡ ···

Anonymous Jay 2y #1701cbf

That makes sense, blanked on the fact that we use the VPN as a look up, thanks for the clarification.

♡ 1 ···

Anonymous Duck 2y #1701cac ✓ Resolved

sp21-final-Q4.3

Would "jalr x0 ra 4" also be correct?

♡ ···

R Rosalie Fang ADMIN 2y #1701dcf

sure!

♡ ···

Anonymous Tiger 2y #1701cab ✓ Resolved

FA21-Final-Q7.4

For the last (#6) memory access, 0b1111111111, what type of miss is it? The solutions don't just say it is a miss, but don't specify. Should this be compulsory, and would the valid bit turn to 1 after this memory access?

4: Miss; as with 3, this must be compulsory

5: Hit; this one hits the valid block in that index.

6: Miss; As noted above, we don't count this as a hit, since our valid bit is off.

♡ ···

Anonymous Stingray 2y #1701caf

Yes, I believe this would be compulsory since the two blocks have valid bit off so this set of the cache is "empty" which means that it must be a compulsory miss.

♡ 2 ···

Anonymous Falcon 2y #1701caa ✓ Resolved

FA21-FINAL-Q1.10

Can i get a further explanation on how to use the availability equation with this given info? Based on the question ik that availability = 980/1000 because it broke twice with 10 days to complete, MTTR = 24*10 = 240. I am confused on if my MTTR would be 480 since it happens twice. is MTTR the total amount of hours that was used for all repairs or is it the number of hours for one repair? either way I don't get the same solution

Q1.10 (1 point) Justin purchased his HP Pavilion 15t-cs300 laptop 1,000 days ago. During this time, it has broken twice, and had to be repaired. Each repair took 10 days to complete, during which time the laptop was unusable. What is the mean time to failure (MTTF) of Justin's laptop, in days?

[ ] days

Solution: 490 days. 1,000 days minus 20 broken days = 980 days, divided by 2 failures.

♡ ···

Anonymous Mandrill 2y #1701daf

MTTR = 10 days = total amount of days for one repair
MTTF = (1000 - 20) / 2 = 490 days
MTBF = 1000/ 2 = 500 = MTTR + MTTF

♡2 ···

Anonymous Jay 2y #1701bff ✓ Resolved

fa-21 final 7.4

Index	Tag 1	Valid 1	Tag 2	Valid 2
0b00	0b1011	1	0b1101	1
0b01	0b0011	1	0b0010	1
0b10	0b1110	1	0b0111	0
0b11	0b1111	0	0b0001	0

Address	
0b 0011011011	☐ Hit ☐ Compulsory miss ☐ Capacity miss ☐ Conflict miss
0b 0011001101	☐ Hit ☐ Compulsory miss ☐ Capacity miss ☐ Conflict miss
0b 0110100010	☐ Hit ☐ Compulsory miss ☐ Capacity miss ☐ Conflict miss
0b 0010111100	☐ Hit ☐ Compulsory miss ☐ Capacity miss ☐ Conflict miss
0b 1110100010	☐ Hit ☐ Compulsory miss ☐ Capacity miss ☐ Conflict miss
0b 1111111111	☐ Hit ☐ Compulsory miss ☐ Capacity miss ☐ Conflict miss

Solution: The crux of this question is that data is always set to some garbage (there's no such thing as blank in binary, and we often don't zero out data if we can avoid it; as such, the cache generally ends up containing whatever data used to be there the last time a program ran), even if the cache block is empty. As such, the valid bit is needed to tell if a block is actually there. If the valid bit is 0, then the block is considered empty, regardless of the corresponding tag.

1: Hit; we notice that Tag 1 matches for our given index, and the valid bit is on.

2: Miss; Our tag isn't in our index. This can either be compulsory or conflict, since we don't know if this block has been accessed before and ejected. This can't be a capacity miss, because there are still empty slots in our cache.

3: Miss; Our tag isn't in our index. This must be compulsory, because we still have an empty slot in this index. We only ever kick out a block when the cache is full, and only to replace that block with a new one; as such, we can't have kicked out a block while there is still empty space in the cache.

4: Miss; as with 3, this must be compulsory

5: Hit; this one hits the valid block in that index.

6: Miss; As noted above, we don't count this as a hit, since our valid bit is off.

Are the explanations for 4 and 5 switched?

If not, I don't understand how 4 is a miss and 5 is a hit

And for 6 if the valid bit is off, what type of miss would it be? It's kind of hard to categorize it under any of the three.

♡ ···

Anonymous Stingray 2y #1701cba

4 is a miss since looking at the set of blocks with index 11, we do not have the tag 0010, which means that, since both the blocks have valid bit off, the set is empty so it must be a compulsory miss.

5 is a hit because looking at the set of blocks with index 10, we do have the tag 1110 and it has valid bit on.

6 is a compulsory miss like 4 by the same reasoning since all valid bits are 0 for this set.

♡ 1 ···

Anonymous Jay 2y #1701cbc

Thanks, I just realized I was reading it wrong; I thought the first two bits were the index even though I filled out the T/I/O correctly lol

♡ 1 ···

Anonymous Dolphin 2y #1701cec

Hi, I have a follow-up question on #2: 0b 0011001101. the index is 00 (row 1 in the cache), and now the block is full, why it cannot be a capacity miss? (In this case, we need to evict one of the entries in row 1 to load TAG 0011 into row 1).

♡ ···

Anonymous Stingray 2y #1701cfa

Replying to Anonymous Dolphin

It can't be a capacity miss because other blocks in the cache are empty (have valid bit 0) so the cache was never full.

♡ 1 ···

Anonymous Gorilla 2y #1701bfc ✓ Resolved

for fa21-final-7.4, if it's a miss since the valid bit is off, do we mark all 3 kinds of misses? or which specific one is it?

♡ ···

Anonymous Stingray 2y #1701cbb ENDORSED

We do not mark all the miss types. For this problem, we never mark the capacity miss because the cache is always empty since there is at least one block with a valid bit off. Therefore, whenever we have a miss, it is not because the cache is too small.

When the set of blocks all have valid bit 1 (which is the case of row 2), it could either be compulsory or conflict because the block may never have been loaded in or it could have been evicted before and the cache is not full.

When the set of blocks have at least one valid bit 0 (which is the case of rows 3,4,6) , it must be a compulsory miss since it is impossible to have evicted a block if the set of blocks has an empty block (it is not a conflict miss).

Hopefully this helps!

♡ 2 ···

Anonymous Dolphin 2y #1701ceb

Hi, I have a follow-up question. When considering capacity miss, are we looking at the block being full or the entire cache being full?

For example, in the second memory access: 0b 0011001101, the index is 00 (row 1 in the cache), now the block is full, why it cannot be a capacity miss? (In this case, we need to evict one of the entries in row 1 to load TAG 0011 into row 1).

♡ ···

Anonymous Stingray 2y #1701cef

Replying to Anonymous Dolphin

If we look at the definition given in the discussion WKS 11, "capacity: occurs if the block was fetched before, but evicted while the **cache was full**".

Alternatively, this is the definition given towards the end of lecture 32, "if the block was evicted when the **cache was full**, it's a capacity miss".

Hope this helps!

♡ 1 ···

Anonymous Duck 2y #1701bfb ✓ Resolved

sp21-final-Q9.1

For the is_null instruction, wouldn't this be not implementable because we would need to add an additional option for the write back mux, since we need to write back either 0 or 1 based on the BrEq control bit? This would require additional hardware since we are not writing back one of PC+4, MEM, or ALUResult

♡ ···

R Rosalie Fang ADMIN 2y #1701dce

There's one more slot on the WBMux!! So we can just add this as an input and not have to add a mux

♡ ···

Anonymous Gorilla 2y #1701bef ✓ Resolved

fa21-final-q3.4: could someone do a step-by-step on how to get this answer? i translated t1 to be 0b00110 since t1 corresponds to x6, but it seems the answer key (converted to binary) has it translated to 0b00011. also, im not sure how -256 was translated into 11110000000. thank you in advance!

♡ 1 ···

R Rosalie Fang ADMIN 2y #1701dcd

jalr ra t1 -256:

ra = x1 = 0b00001

t1 = x6 = 0b00110

-256 => 256 = 0b000100000000 => flip bits = 0b111011111111 => add 1 = 0b111100000000

Then translate into 32 bits, read the hex from right to left.

♡ ···

Anonymous Newt 2y #1701bed ✓ Resolved

sp21-final-Q8

I don't quite understand this question's setup. the input is 0111, 0100, and why is the output 0,0,1,0? (are we adding the two strings bit by bit? like 1+0, 1+0, 1+1, 0+0?)

♡ ···

Anonymous Jellyfish 2y #1701bfa

The output bit is whether there is an overflow from adding the two bitstrings, from the LSB. Here I'm adding the two bitstrings from right to left and checking the output. So looking at 0111 and 0100, would be:

1. 1+0 = 1
2. 1+0 = 1
3. 1+1 = 2 (overflow!)

4. 0+0 = 0

We can see the overflow occurs at the third operation, hence 0, 0, 1, 0.

♡ 1 ···

Anonymous Parrot 2y #1701beb ✓ Resolved

is this n byte alignment rule in scope?

Q6.1 (2 points) What is `sizeof(struct foo)` (Answer as an integer, with no units)?

Solution: 12
This question (and solution) uses the alignment rule where an n-byte struct field must start at an n-byte-aligned boundary. For example, `char *c` is a 4-byte field, so it must start at a 4-byte-aligned boundary (the offset of `c` relative to the start of the struct must be a multiple of 4).

♡ ···

E Erik Yang TA 2y #1701dae

yes

♡ ···

Anonymous Newt 2y #1701bea ✓ Resolved

SP21-Final-Q2, F/G

how do we calculate the hit rate for those 2 for loops?

for F, I understand the HR for the first iteration would be 2/3, but why starting at the 2nd outer iteration, the hit rate becomes 1? (how does eviction work here?)

for G, I also don't understand why the hitting rate is all 1, I thought you would always have compulsory miss at first.

```
arr[j] = arr[j] * arr[i]
```

1

The entire array can fit in memory. Because the access pattern is effectively the same, all accesses are hits.

♡ ···

Anonymous Jellyfish 2y #1701bfd

Notice that the size of the cache is 512Kib = 2^19 while the ARR_SIZE = 2^12. This is what the answer key means by "entire array can fit in memory" since cache size > ARR_SIZE. So on the second iteration, the hit rate becomes 1 because the entire array is loaded in the cache because of the first iteration.

For G, it is the same reasoning. Since it continues after the first loop, all the elements of the array are already loaded in the cache and thus always a hit.

♡ 2 ···

Anonymous Duck 2y #1701bdf ✓ Resolved

FA21-Q3.6

I think there is a discrepancy in the answer key and the explanation. The answer key selects "jalr ra t2 -256" and "jalr ra t0 16" as the correct choices, but the explanation says "jalr s0 t1 -256 and

jalr ra t0 16 are valid"

♡ 3 ···

R Rosalie Fang ADMIN 2y #1701dcc

You're right. Thanks for the catch. jalr s0 t1 -256 should be correct instead.

♡ ···

N Neeraj Rattehalli 2y #1701bdd ✓ Resolved

FA21-Q2. Why is the answer 36 and not 38? why round down as opposed to up?

♡ ···

A Alson Chan 2y #1701dba

I had this question too!

♡ 1 ···

R Rosalie Fang ADMIN 2y #1701dcb

"In the event that the number is exactly halfway between two floating point numbers, round to the number with a least significant bit of 0."

♡ ···

Anonymous Duck 2y #1701bdc ✓ Resolved

FA21-Q9

For the last blank, the solution says "output[0]+output[1]+output[2]+output[3] == 0 is incorrect, since we could have received outputs that happened to add to 0, even if they aren't all 0." Since we are using unsigned integers, though, how would this be possible? Wouldn't every output be greater than or equal to 0?

♡ ···

R Rosalie Fang ADMIN 2y #1701dca

Look at the example that was provided. There might be overflow even if all outputs are unsigned.

♡ ···

Anonymous Panther 2y #1701bda ✓ Resolved

SP21-Q2

I'm a little confused as to when to realize when we should evict a block. The example here was a 4-way associative block, so how would I know when I need to evict a block?

♡ 1 ···

R Rosalie Fang ADMIN 2y #1701dbf

Eviction happens when your set is full.

♡ ···

Anonymous Panther 2y #1701bcf ✓ Resolved

SP21-Q1 (Potpourri - Q4)

I get why D would be sped up by pragma omp for, but I don't get why A would also be sped up. Wouldn't the results be wrong since different threads might be performing the operation at different times so A[i] would not be loaded? If A is included why aren't the other ones included too?

♡ 2 ···

R Rosalie Fang ADMIN 2y #1701dbe

Are you looking at the rewritten answers? There's a "Note: We think this question had a typo and meant to say #pragma omp parallel for instead of #pragma omp for.
"

♡ ···

Anonymous Panther 2y #1701bce ✓ Resolved

SP21-Final-Q3: what does "generate parse trees" refer to, and why does the compiler take care of that?

♡ 3 ···

R Rosalie Fang ADMIN 2y #1701dbd

This is kind of a specific term used in the compiler that was probably brought up in sp21's lectures... keyword here is "parse", compiler handles parsing the input programs.

♡ ···

Anonymous Otter 2y #1701bac ✓ Resolved

SP21-final-1 (Q2)

for Q2A did we ever cover MTBF in class? why is availability not 4000/4003?

for Q2B how to solve this problem. Where did the 100 come from?

♡ 1 ···

Anonymous Human 2y #1701baf

For 2a I believe that MTTF + MTTR = MTBF and Availability = MTTF/MTBF so we would subtract 3 from 4000 for the numerator and our denominator would be 4000

Following! for 2B!

♡ ···

Anonymous Otter 2y #1701bba

Thanks for 2A!

♡ ···

E Erik Yang TA 2y #1701bbc

2B: since annual fail rate is 1%, then you fail once every 100 years. MTTF = 100

♡ ···

Anonymous Duck 2y #1701afd ✓ Resolved

FA21-1.12

If our error-correcting code is able to fix 1-bit errors, doesn't this mean the Hamming distance between two valid code words is 3? So if the distance from 0x61C to another code word is 3, wouldn't the other code word be valid?

♡ ···

R Rosalie Fang ADMIN 2y #1701bcb

You are correct. both 0x71C and 0x51C have hamming distance of less than 3 from 0x61C. Which option of the solution is confusing?

♡ ···

Anonymous Human 2y #1701afb ✓ Resolved

For Fa21-Final Q5.3

If I wrote

mac a0 a1 a1

add a0 a0 4

would this be valid? Because there is data dependency between the first instruction and the second on a0?

♡ ···

E Erik Yang TA 2y #1701bbd

i don't think so, because you want a sequence where it would not cause a hazard if all mac instructions became mul instructions (from problem statement). Here, if you replace mac with mul, you will still get a hazard.

♡ ···

Anonymous Viper 2y #1701ade ✓ Resolved

FA21-1.12 I am a bit on their explanation can somebody explain how we know these aren't valid and what rule we use?

♡ 1 ···

R Rosalie Fang ADMIN 2y #1701bcc

We're trying to do 1-bit error correction so we need a hamming distance of at least 3. Try figuring out the hamming distance between 0x61C and all the options (hamming distance is the number of bits that differ) and the ones that have a hamming distance of less than 3 are the answers.

♡ ···

Anonymous Viper 2y #1701bcd

Appreciate the response so do we need a hamming distance of 6 for 2 bit error correction, or would it still be 1. More of a generic question not as much pertaining to the question

♡ ···

R Rosalie Fang ADMIN 2y #1701dbc

Replying to Anonymous Viper

hamming distance of 5 for 2 bit error correction (2 * n + 1) where n is the number of bits you want to correct.

♡ ···

Anonymous Human 2y #1701adc ✓ Resolved

Fa21-Final-Q7.3

Just to clarify in a 2-way associative cache, one block has two entries? And if the question were to ask for total amount of bytes and not just data bytes we would have to sum up the bytes of tags * entries + valid bits + data bytes? Because a cache entry has a tag, valid, and data?

♡ ···

E Erik Yang TA 2y #1701afa

2 -way cache means for every index, you can fit 2 blocks. Yeah, that sounds correct

♡ 1 ···

Anonymous Parrot 2y #1701ada ✓ Resolved

Fa21-Final-Q9

For Blank 2 I put **64 * n**, for Blank 3 I put **256**, and for Blank 7 I put **output[0] == 0 && output[64] == 0 && output[128] == 0 && output[192] == 0**

The correct answers were n, 4, and **output[0] == 0 && output[1] == 0 && output[2] == 0 && output[3] == 0.**

It seems like I am considering length of bits when I shouldn't. Does C automatically consider the size of each int when indexing the pointers?

♡ ···

Anonymous Dunlin 2y #1701adb

yes

♡ 1 ···

R Rosalie Fang ADMIN 2y #1701aee

Yes, C indexing is index by i-th element not by bit/byte

♡ ···

Anonymous Parrot 2y #1701acf ✓ Resolved

FA21-Final-Q5.3

The solution states "The extra hazard that occurs as a result of this instruction is a data hazard on rd; with mul, we don't need to wait for the value of rd to get written back, but we do need to wait for rd for a mac." What does this mean?

♡ ···

R Rosalie Fang ADMIN 2y #1701aed

The question is asking for hazards that is specifically a result of mac but not mul instructions. Therefore the solution is trying to explain exactly what the difference is between a mac and a mul instruction

♡ ···

Anonymous Cobra 2y #1701acd ✓ Resolved

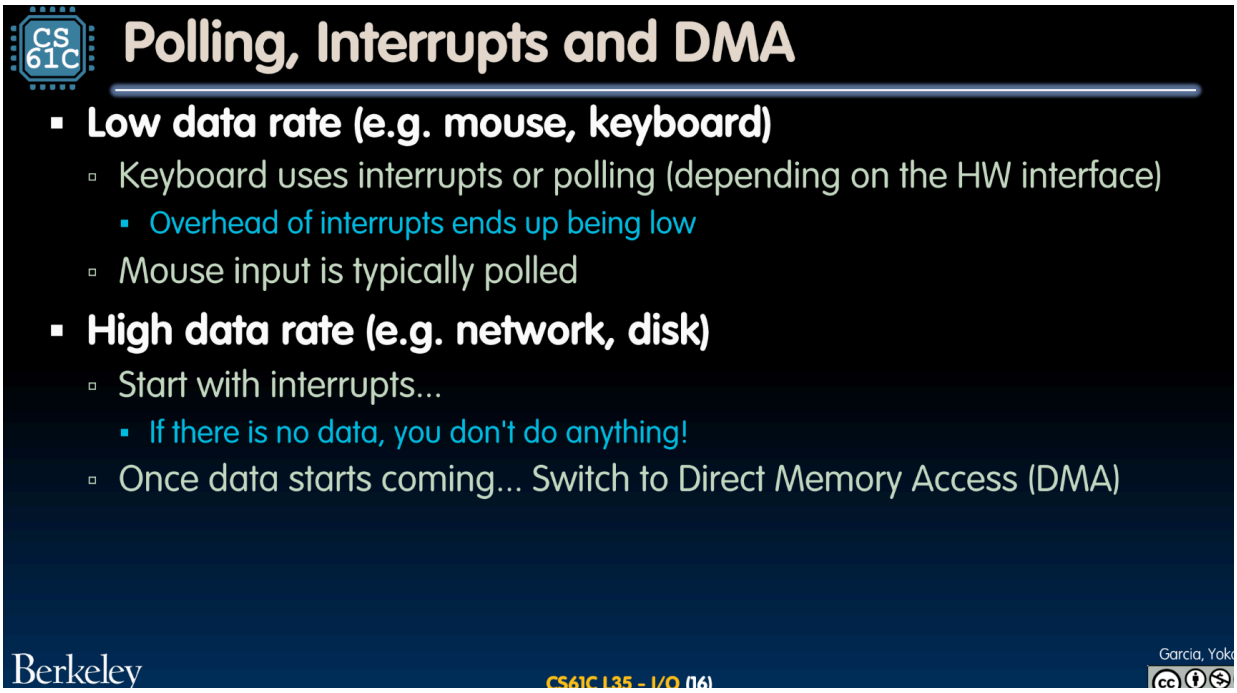
FA21 Q1, 1.4)

I'm confused as to why the answer key says that polling is for high data rate when the slides say otherwise?

Q1.4 (1 point) TRUE or FALSE: For high-performance network devices, polling tends to be used when there's a low data rate, while interrupts tend to be used when there's a high data rate.

○ TRUE ● FALSE

Solution: False; generally, the opposite tends to happen.



Am I missing something?

♡ 5 ···

R Rosalie Fang ADMIN 2y #1701aec

high data rate is just "start with interrupt". Usually when you think about high data rate that means we have a lot of data, so polling is better.

♡ ···

Anonymous Goldfinch 2y #1701ccd

Could you elaborate a bit more about this? I'm still very confused about the contradiction. So are you saying when there is a low data rate (e.g. mouse/keyboard) we use interrupts instead of polling?

♡ ···

Anonymous Jay 2y #1701ccf

Replying to Anonymous Goldfinch

I'm not a TA, but maybe I can help a little bit.

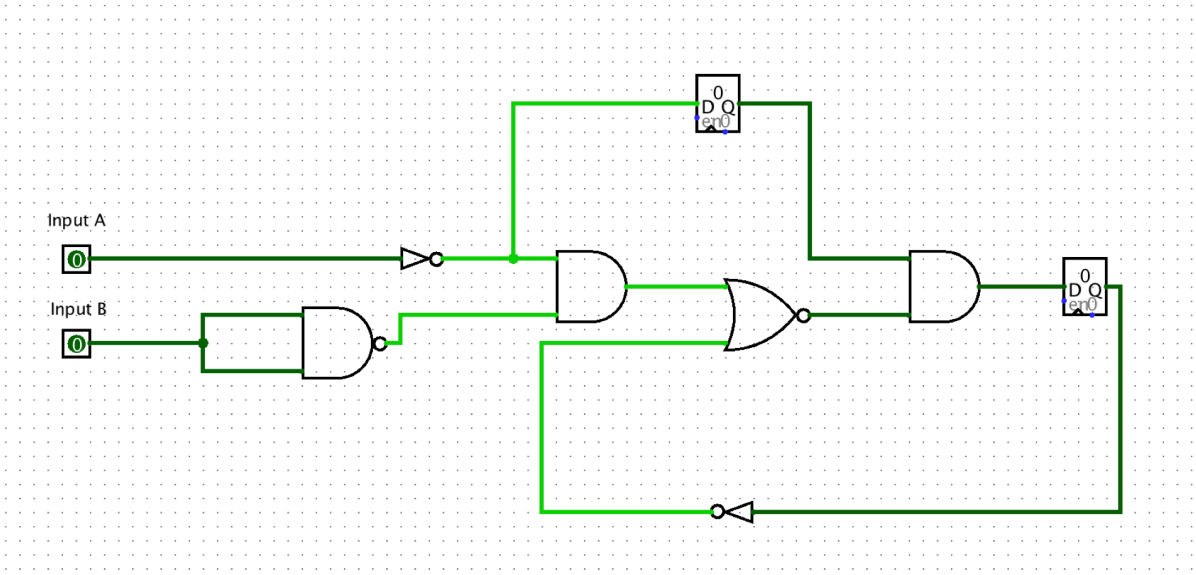
Yes, when there is a lower data rate we can use interrupts instead of polling because we know polling is always in a loop asking the I/O device if it's "ready." Thus if we use polling in a low data rate device, we're expending energy when it isn't needed compared to interrupts which only act when needed, which is better for a lower data rate device.

This is the way I think about it, hopefully it doesn't lead to any more confusion. Please feel free to correct me if I am wrong as well.

♡ ···

Anonymous Mule 2y #1701acc ✓ Resolved

SP21 Q4(a)



Are inputs also considered as "registers" so that they would also have a clk-to-q time? Because initially I thought they are not registers, so I did't consider them as valid component in the longest combination path.

♡ ···

R Rosalie Fang ADMIN 2y #1701aeb

Yes, but their "clk-to-q" time is specified. Usually you'd see something along the lines of "input A changes x ns after the rising edge of the clock"

♡ ···

Anonymous Mule 2y #1701acb ✓ Resolved

SP21 2(a)K

Is this question in scope? I think we've learned using a cache for page table. Is that the same as 2-level page table?

♡ ···

R Rosalie Fang ADMIN 2y #1701adf

2-level page tables are not in scope.

♡ ···

Anonymous Ibis 2y #1701aad ✓ Resolved

for Fa 21 1.9, shouldn't both thread 1 read x and y? So if thread 1 reads y = 0, then shouldn't x = 10 for thread 1. In addition, both y and x are shared variables, wouldn't that take affect

of 9 11B

Q1.9 (1 point) We run the following code on two threads.

```
1  int y = 0;
2  int x = 10;
3  #pragma omp parallel
4  {
5    while(x>0)
6    {
7      y = y+1;
8      x = x-1;
9    }
10 }
```

What is the smallest possible value y can contain after this runs?

Solution: This one's a bit tricky. The optimal sequence is:

Thread 1 reads y=0 and goes to sleep.
Thread 2 runs to completion.
Thread 1 wakes up and writes y=1, reads x=0, sets x=-1, then sees x == -1 and stops the loop.

♡ ···

R Rosalie Fang ADMIN 2y #1701aca

Sorry can you rephrase your question? Also this is the exact same question as HW7.5. You can look back at HW7.5 to see a more guided version of this.

♡ ···

Anonymous Jay 2y #1701bfe

Following up on this, I had a question as I understand up until thread one wakes up again and reads x = 0, but where I get confused or might have forgotten from hw 7.5 is how can x be set to -1 when the while condition is x > 0? Isn't the condition false, causing it x to stay at 0 and thus, y would just stay at 0?

I'm not sure if my understanding of threads is lacking, thank you in advance!

♡ 1 ···

Anonymous Mandrill 2y #1701cfd

Replying to Anonymous Jay

I think we're assuming that thread one has already entered the loop before it goes to sleep. Therefore, when it wakes up it doesn't have to check the while condition until after it does everything inside the body. At that point, y is incremented, x is set to -1 and we don't satisfy the condition anymore.

♡ 1 ···

Anonymous Jay 2y #1701cfe

Replying to Anonymous Mandrill

Thanks for your response. So does thread 1 enter the loop without reading x = 10, because then when it wakes up, why wouldn't it just read as y as y = 10 after thread 2 has completed the same way thread 1 reads x as x = 0?

♡ ···

Anonymous Mandrill 2y #1701cff

Replying to Anonymous Jay

from the hw7 problem: "the expression y = y + 1 is equivalent to three instructions: load value of y, add 1, store result to y. Another thread can execute in between those instructions."

so in this case i'm assuming we first load y = 0, then we go to sleep. once we wake up, we add 1 and y becomes 1. So thread 1 probably does read x = 10 for the branch instruction when entering the loop, but after it wakes up and increments y, it still has to load in the new x before x is decremented.

♡ 1 ···

Anonymous Ibis 2y #1701aac ✓ Resolved

Fa 21

using ahmdal's law I got 10/1.9. How did we get the extra 0.1 for for 10/2

Q1.13 (1 point) A program originally takes 1 second to run. We manage to parallelize 90% of our code to be 10 times faster, at the cost of 10 milliseconds of overhead. How many times faster is our new code?

Solution: 5 times faster. 1s / ((1s * 10%) + (1s * 90%) / 10 + 0.01s)

The easiest way to do this question is not to use the formula for Amdahl's law, but rather to treat it as a word problem. 90% of our code is 0.9 seconds worth of runtime, and that gets sped up to 0.09 seconds. We still have 0.1 seconds from our serial section, and add 0.01 seconds of overhead. This adds up to 0.2 seconds, which is 5 times less than our original runtime.

A useful note: Amdahl's law can almost always be solved in this "word problem" manner, by assigning an arbitrary runtime to the original code. I've generally found it more useful to know this instead of the formula itself.

♡ ···

R Rosalie Fang ADMIN 2y #1701abf

"At the cost of 10 milliseconds of overhead" means that 10 milliseconds is added to our overall program runtime, which is the denominator so we add the 0.1.

♡ ···

Anonymous Caterpillar 2y #1701aab ✓ Resolved

FA21-Final-Q8

Why is it that with the same Virtual address shared between the two programs, we use the same page for 8.7 but not for 8.9? What is the distinction between these two cases?

♡ ···

R Rosalie Fang ADMIN 2y #1701abd

Try looking at homework 9.2. It walks through an exact case of this. Two programs don't share physical memory, so they have different PTBR (page table base pointer), which means the process of translation will give you different results since they're looking at different page tables.

♡ ···

R Rosalie Fang ADMIN 2y #1701abe

Specifically, for 8.7 the address is 0xABCDEFAB, so VPN is 0xABCDE. Program 2 has never accessed that virtual page before so we need a new mapping. for 8.9 the address is 0x12345664, VPN is 0x12345 and program 1 already had an access to that exact same virtual page in 8.5

♡ ···

Anonymous Caterpillar 2y #1701aaa ✓ Resolved

FA21-Final-Q8.4

Q8.4 (1 point) Program 1: 0xABCDEFAB

Solution: This is our first page, so we get the physical page $0x00$. The page offset is the last 12 bits of our address (since our page size is 2^{12} bytes), so we take the last 12 bits of the virtual address: 0x00FAB.

Q8.5 (1 point) Program 1: 0x12345678

Why is it true that our first page gets physical page 0x00? There is no page table to refer to. It would be useful if I could know the procedure for which physical pages are assigned when a page table isn't present.

♡ 1 ···

R Rosalie Fang ADMIN 2y #1701abc

"Assume that no physical pages are in use prior to the first memory access, and that physical pages get assigned in order of physical page number"

That means there is no usable page table, or basically everything has a valid bit of 0 since there currently exist no virtual to physical mappings. That means whatever address we try to access are going to be page faults, in which case we assign physical pages to this program. Since here we assigned physical pages in order of PPN, the first PPN is 0x0.

♡ ···

Anonymous Caterpillar 2y #1701ff ✓ Resolved

FA21-Final-Q4.1

The answer is 16 because it only includes the OR and AND gate. However, i thought clock periods must include the clock to q of the initial register and the setup of the final register as well. What is the difference between "largest combinational delay", "largest combinational path", and "clock period", and is it true that "largest combinational path" and "clock period" contain the clock to q of initial register and setup of final register, while the "largest combinational delay" does not?

♡ ···

R Rosalie Fang ADMIN 2y #1701abb

"largest combinational delay" = "largest combinational path" which is just the delays of all the **logical elements** on the critical path. That means no setup time and no clk-to-q since those are related to state elements which depend on the clock.

"clock period" refers to the critical path, which is clk_to_q + largest combinational delay + setup

♡ ···

Anonymous Mule 2y #1701fc ✓ Resolved

FA21-Final-Q1.8

I didn't really get how we use the parity. The solution says we got one parity for every 9 blocks, is that a fixed rule?

♡ 2 ···

R Rosalie Fang ADMIN 2y #1701aea

We decide to set up 10 1 TiB disks together in a single RAID configuration. Since RAID 5 has 1 parity block, the rest is used for data blocks.

♡ ···

Anonymous Dugong 2y #1701cfc

So is a general rule we can follow is effective amount of storage = total disk size - 1 * (size of singular disk which is used for parity)?

♡ ···

Anonymous Elephant 2y #1701ed ✓ Resolved

FA21-Final-Q9

Should the 2nd blank be `n/4` or `n` ?

♡ ···

E Erik Yang TA 2y #1701ee

n, we assume that n is a multiple of 4 in the problem statement

♡ ···

Anonymous Pigeon 2y #1701ec ✓ Resolved

Summer Mt2 5(d)

(d) (4.0 points) Forwarding

We now add some forwarding paths to our datapath from the previous questions as seen below (Figure 4). For our code above, how many NOPs/stalls will we need now for our code to operate correctly? What about after we add forwarding paths from the end of DMEM1 to the beginning of DMEM1 and from the end of DMEM2 to the beginning of DMEM2? Give your answers in order. Note that not all forwarding logic is depicted in the diagram given; only the important forwarding paths are given. Refer to the key in the diagram for forwarding path types.

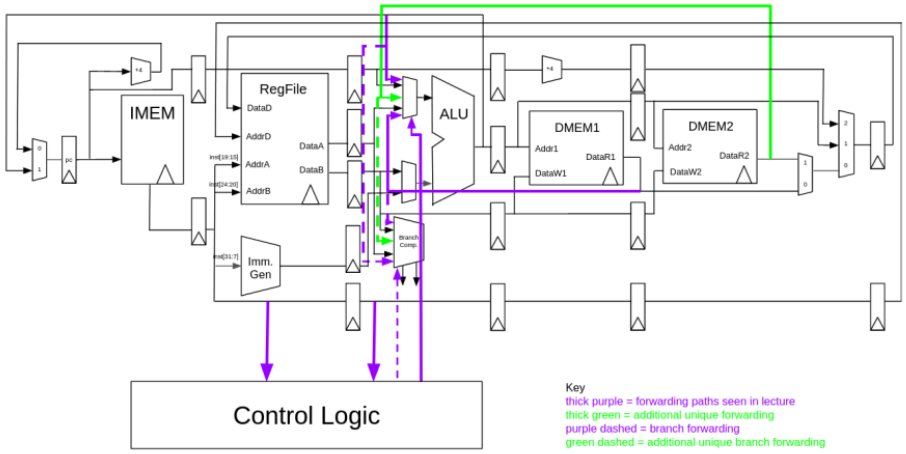


Figure 4

i. **(1.5 pt)** No memory forwarding paths NOPs/stalls count:

5

TBD

ii. **(1.5 pt)** With additional memory forwarding paths NOPs/stalls count:

5

TBD

iii. **(1.0 pt)** Looking at our datapath, what kind of hazards still remain given our pipeline? Select all that apply.

- [x] Control
- [ ] Structural
- [x] Data
- [ ] None

TBD

I'm really confused about how to approach this problem.

This problem IF and MEM are both splited into 2 stages.

♡ ···

E Erik Yang TA 2y #1701ef

i think our review session went over this problem if you want to take a look a tthat

♡ ···

Anonymous Pigeon 2y #1701fa

- Session 1: Datapath, Parallelism
  - Time: Wednesday, 5/3, 2-5 PM PT
  - Location: Cory 540AB
  - Slides:
  - https://docs.google.com/presentation/d/1QAB8j9tx5yUFxCaKz39iSHMqub1oxoOCMvEyNOD0Uh8/edit?usp=sharing
  - Recording:
    - Datapath: https://drive.google.com/file/d/1eyXj6M7DB84UWBdAH61i1T4Q1CZELTrL/view?usp=share_link
    - Parallelism: https://drive.google.com/file/d/10XJbQ3vqvBg8pCVhTjA8anIDEdm5X_3a/view?usp=sharing

Are you referring to this one? I can't find it though.

♡ ···

Anonymous Pigeon 2y #1701fb

I found it!

♡ ···

Abby O'Neill 2y #1701db ✓ Resolved

for 3.3 i understand that we are moving the stack pointer down 180 and then calling get 20 char which actually just copies all the 256 bytes worth of instructions and then call jr sp to execute those instructions, but what happens since there is no null terminator at the end how does the program know to stop?

♡ ···

E

Erik Yang TA 2y #1701eb

Are you talking about in GetChars? The null terminator comes from the input

♡ ···

Anonymous Falcon 2y #1701ce ✓ Resolved

Sp21-Final-Q1(i)

I understand that we have to use Amdahl's formula for this question but I'm not sure how exactly it was used to derive 1/7. Can someone explain for me?

♡ ···

E

Erik Yang TA 2y #1701ea

S = 4, N= 8. Solve for F using the speedup formula

♡ ···

Anonymous Ibis 2y #1701daa

sorry, what is N? I don't see an N used for Ahmdal's formula

$$\textbf{Speedup} = \frac{1}{(1-F) + \frac{F}{S}}$$

tical speedup of the program

♡ ···

E Erik Yang TA 2y #1701dac

Replying to Anonymous Ibis

N would be the number of cores

♡ ···

Anonymous Ibis 2y #1701dee

Replying to Erik Yang

So do I set N equal to the speedup formula and solve for F?

♡ ···

E Erik Yang TA 2y #1701dfa

Replying to Anonymous Ibis

You want at least 4 speed up so you want it equal to 4

♡ ···

Anonymous Dragonfly 2y #1701cd ✓ Resolved

sp21-final-2k: why is the answer 2? I don't get how the one level pg table translates to the two-level page table

♡ ···

R Rosalie Fang ADMIN 2y #1701aba

two-level page tables are out of scope :)

♡ 1 ···

Anonymous Quelea 2y #1701cc ✓ Resolved

FA21-Final-Q6.3

How does W ^ Z yield X for inputs 0 0 0 and 1 0 1?

♡ ···

A Aarin Salot 2y #1701cf

i think its a typo in the solutions

♡ ···

E Erik Yang TA 2y #1701df

X means it can output either a 0 or 1, it doesn't matter. You just need to look at the outputs that have values in them

♡ ···

Anonymous Ostrich 2y #1701cb ✓ Resolved

SP21-Final-Q6A

Are alignment rules in scope? I don't remember going over that in lecture. If so, where can I read up about it?

♡ ···

E Erik Yang TA 2y #1701de

yes, alignment rules mean like the structure is 4-byte aligned, so padding will take in effect. Also, this is sp21

♡ ···

A Aarin Salot 2y #1701afc

for part ii on this question, why will the size increase is they're swapped? is it bc the character pointer can't go directly after the 1 byte allocated to char a, so it wastes 3 bytes of space instead of 1?

♡ ···

E Erik Yang TA 2y #1701afe

Replying to Aarin Salot

exactly, if we swap, then char a needs 3 bytes of padding before char*c. Then, it is 1 byte for the 1 byte integer **b ,** and then that needs 3 bytes of padding before you add the struct poitner for 3.

♡ 1 ···

A Aarin Salot 2y #1701aff

Replying to Erik Yang

for Q6 part b.v, why are doing

slli t1 t1 4

add t0 t0 t1

lw t0 t0(4)

Shouldn't be be like

mul t1 t1 4 (multiply index of level by 4 (the number of bytes in a pointer))

addi t0 t0 t1 (location of where the pointer to next[level] is.

lw t0 0(t0) (gives us the pointer to next[level])

lw t0 0(t0) (gives us the struct the pointer is pointing too)

♡ ···

E Erik Yang TA 2y #1701baa

Replying to Aarin Salot

both do the same thing; it's just the first way was how we taught it in lab,disc

♡ ···

A Aarin Salot 2y #1701bab

Replying to Erik Yang

isn't slli t1 t1 4, the same as multiplying the index by 16 tho?

also how can we do this with 1 load since we have 2 pointers?

♡ ···

E Erik Yang TA 2y #1701bad

Replying to Aarin Salot

slli t1 t1 2

add t0 t0 t1 l

lw t0 0(t0)

Edit: this should be the correct answer

♡ ···

E Erik Yang TA 2y #1701bae

Replying to Erik Yang

also i'm pretty sure the answer has a typo, it should be slli t1 t1 2, which is every 4 bytes

♡ ···

A Aarin Salot 2y #1701bbb

Replying to Erik Yang

gotchu, thanks Erik!

♡ ···

Anonymous Dotterel 2y #1701bbf

Replying to Erik Yang

why it isn't slli t1 t1 3? The struct has two pointers, which is 8 bytes

♡ ···

E Erik Yang TA 2y #1701bca

Replying to Anonymous Dotterel

t0 already stores the **next** pointer, which is an array of 4 byte pointers. We're already operating within the struct

♡ ···

Y Yury Orlovskiy 2y #1701ca ✓ Resolved

SP21-MT-Q5B, what is the additional mux responsible for? I don't quite understand what we get from the control logic for that additional mux, and what it does

♡ ···

E Erik Yang TA 2y #1701dd

we need to decide if alu needs to pass additional data to MEM2 if MEM1 doesn't succeed

♡ ···

Anonymous Elephant 2y #1701ace

How do we decide which mem output gets put in the WB mux? Also, why to we need path from control logic to MEM1?

♡ ···

E Erik Yang TA 2y #1701aef

Replying to Anonymous Elephant

from the problem statement, if the value is found in Mem1, then we just pass that result into the WB mux. Otherwise, we need to use MEM2 in order to pass that result into WB mux. We need a path from MEM1 to control logic because MEM1 sends a signal to control logic saying whether or not the data was found in MEM1, and then this is propagated into MEM2. (This is also from the problem statement)

♡ ···

Anonymous Elephant 2y #1701bbe

Replying to Erik Yang

I understand that but why do we need one going the other way, from control logic to mem1?

♡ ···

Anonymous Elk 2y #1701bf ✓ Resolved

Fa21-final-Q7(b): if the instruction changes format, (for example: is_null, rd, rs1, imm), would we still select "Modify the control logic for parsing instr[31:0]" option? Or is it only selected because we don't have an instruction with the format name rd, rs1

♡ ···

E Erik Yang TA 2y #1701dc

this is in spring2021, we need to modify control logic because we're assuming this is R-type format

♡ ···

Anonymous Seahorse 2y #1701bb ✓ Resolved

fa21-final-q9

why do they have the vectorOr line? Since we are doing the or between total (which is a vector of all 0's) and isequal, won't the total vector always just be a vector of all 0's? What is the point of doing that?

♡ ···

J Jero Wang ADMIN 2y #1701bc

If any element in `isEqual` is 1, then the `total` vector would have a 1 in it. This is used to detect if `isEqual` ever has a 1 in it.

♡ 2 ···

Anonymous Dragonfly 2y #1701ad ✓ Resolved

fa21-final-q3.3) What is the injected instructions the answer key is referring to? I don't get why the pointer is moved 280 bytes as opposed to 256.

♡ ···

E Erik Yang TA 2y #1701ba

it's not 256 because we allocate 256 bytes for the buffer, and then from part 1, you see that we move the stack down by 24 bytes, so we need to move it down 280 bytes so that we don't overlap with anything from the stack so far

the inject instructions is from the first part

♡ ···

Anonymous Duck 2y #1701bde

Isn't the stack pointer already moved down 24 bytes by the time we jump to ra though? Meaning we would only have to move the stack pointer down by another 256 bytes

♡ ···

E Erik Yang TA 2y #1701bec

Replying to Anonymous Duck

i think these injected instructions happen outside of the original code, so you account for the changes made in the stack from the original code

♡ ···

Anonymous Duck 2y #1701bee

Replying to Erik Yang

I don't see how it would be possible for the injected instructions to happen outside the original code, though, because in order to jump to the injected code we would have to run the verifypassword function, which decrements the stack pointer by 24 in its first instruction.

♡ ···

E Erik Yang TA 2y #1701cae

Replying to Anonymous Duck

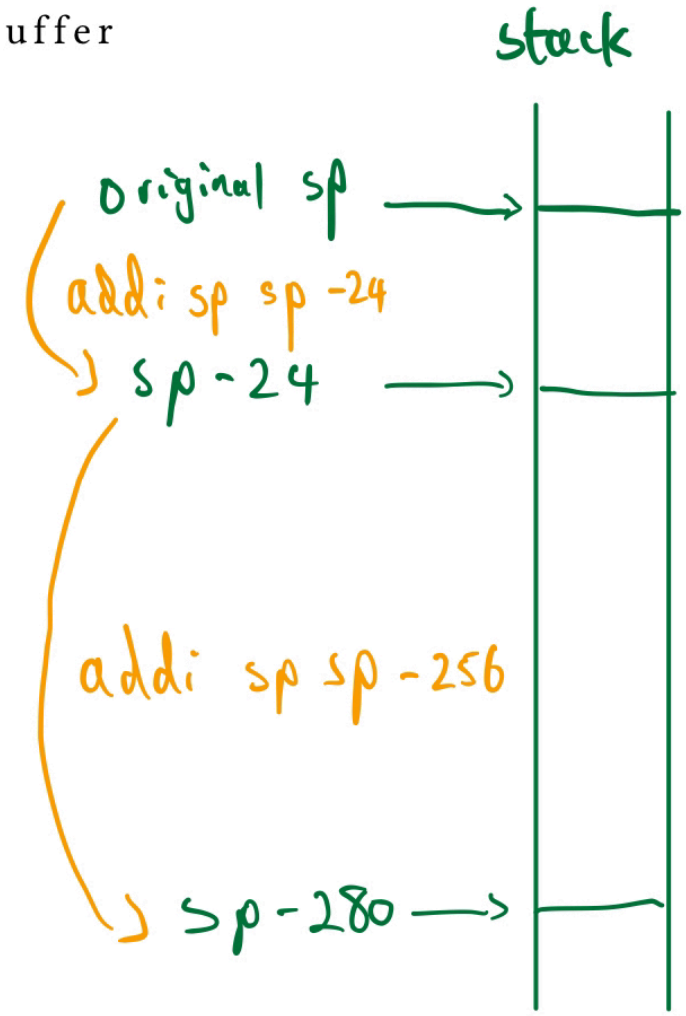
from 3.2, it says that we jump to the start of the buffer on the stack, so sp will be at the top

♡ ···

Anonymous Goldfinch 2y #1701cce

Replying to Erik Yang

How does jumping to the start of the buffer on the stack affect sp at all? Correct me if I'm wrong, but after line 2, the stack pointer points to the start of the buffer (filled by Get20chars, injected from stdin) and this is only moved back after line 23. By moving the stack pointer down another 256 bytes we never overwrite any of the buffer. Moreover, there was never an opportunity for any other code to be run before we run the jr sp line in Q3.3, because the 20 bytes of buffer space are all used. I attached a picture showing the sp changes if that helps



♡ ···

E Erik Yang TA 2y #1701cda

Replying to Anonymous Goldfinch

that's a good point, but the only register that accounts for the stack is sp, so we do have to modify sp in order to get to the top of the 20 byte buffer. The jumping is all conceptual (as stated in the problem statement) so I think you can assume that we are at the top of the buffer, without having any code that does that.

♡ ···

Anonymous Pigeon 2y #1701ac ✓ Resolved

SP 21 final Q8.4-10

I'm really confused about what's the meaning of "physical pages get assigned in order of physical page number"

♡ ···

E Erik Yang TA 2y #1701af

i'm sorry i can't find that question, are you sure it's the right number?

♡ ···

Anonymous Echidna 2y #1701e ✓ Resolved

SP22-Final-Q2F

I read the explanation for why the hit rate is 47/48. I am still a bit confused on how this fits. Like it says it misses arr[0] for a read then a hit on arr[1]? which I am not sure where the 1 comes from.

Can someone explain how the first iteration works?

arr[i] = arr[i] + arr[j]

Doesn't i stay the same the whole first inner loop? then j is just missing every subsequent one?

♡ ···

Anonymous Echidna 2y #1701f

(2/3 * 1/16) + (3/3 * 15/16) = 47/48? Does that logic make sense? I am confused about why 2/3 for the first outside iteration and why 100% hit rate for the rest of the outside iterations.

♡ ···

Anonymous Jaguar 2y #1701aa

I think its because for the first outer iteration, every time we read the arr[i] we need to bring in a new block therefore its a miss. However, arr[j] is in that block that we load so thats a hit. Writing to arr[i] is also a hit. After the first outer iteration is finished we will have all needed blocks in the cache so we will keep hitting in the following iterations. Hope this helps!

♡ ···

Anonymous Human 2y #1701b ✓ Resolved

SP21-Final-Q2

H.I How do they arrive at 1024 pages? If someone can breakdown the calculations that would be amazing as I'm not really sure where to even start

♡ 3 ···

Anonymous Jaguar 2y #1701c

same question :D Any help is appreciated (I got 2^12 pages :(

♡ ···

E Erik Yang TA 2y #1701ae

The VPN has 20 bits so there are 2^20 *entries.* Each entry has 8 bits of metadata and the PPN has 12 bits, so there are 20 bits per Page Table Entry. That means there are 2^20 · 20 bits in the page table. But since this is a 32 bit system, the 20 bits are padded to 32.

2^20 * 32 / 4 KiB = 2^20 * 32 bits / 4 * 8 * 2^10 = 1024.

♡ 2 ···

Anonymous Pigeon 2y #1701fd

Where does the extra 8 come from in your final equation?

♡ ···

E Erik Yang TA 2y #1701fe

Replying to Anonymous Pigeon

Since 4 KiB is in bytes, we need to convert to bits. 1 byte = 8 bits

♡ 1 ···

Anonymous Bee 2y #1701aae

Can we say in general for any page table entry that is less than 32 bits, we pad it to 32 bits? Thanks.

♡ ···

E Erik Yang TA 2y #1701aaf

Replying to Anonymous Bee

good question, since we didn't specify what type of alignment it is, you can go ahead and pad it to 32 bits

♡ ···

N Nick Zheng 2y #1701cde

Replying to Erik Yang

Is there an extra valid bit in each page table entry so that each page table entry takes 21 bits?

♡ ···

E Erik Yang TA 2y #1701cdf

Replying to Nick Zheng

i don't think so

♡ ···

Anonymous Human 2y #1701a ✓ Resolved

FA21-Final-Q8

For this question, I assumed that because the two programs don't have shared memory, I can assign the same ppn's again when it page faults. For example, for 8.7 I had 0x00FAB instead of 0x03FAB. I just want to confirm that this is because we still have space in memory and don't have to evict anything yet.

Also I wanted to ask how this would differ if they did share memory. We would still have space in memory so would anything change?

♡ 2 ···

Anonymous Pigeon 2y #1701ab

Same confusion

♡ 1 ···

J Jero Wang ADMIN 2y #1701bd

Pages are not evicted during a context switch, so you cannot reuse the PPNs. You can only reuse PPNs after you start evicting pages.

If they shared memory, when they accessed the same addresses, it would return the same PPN, but only if those pages are shared.

♡ ···