[Final] Past Exams - 2019 and Older #1699

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 2019 final walkthrough

Summer 2019 final walkthrough

- Q1 Potpourri: https://youtu.be/FY5dAMrXvxo
- Q2 FSM: https://youtu.be/gmHbw6LSeSw
- Q3 C Coding: https://youtu.be/v4B1WTs5UNU
- Q4 RISC-V: https://youtu.be/2VHjG-gy9Dk
- Q5 Data-Level Parallelism: https://youtu.be/oG9Rrzmi0M4
- Q6 RAID and ECC: https://youtu.be/rfCNTIzNZ2M
- Q7 Caches: https://youtu.be/xojc8YZaO3Q
- Q8 Spark: https://youtu.be/A37BFXRXmm0
- Q9 Datapath: https://youtu.be/q-T4N3hBhUM
- Q10 Digital Logic: https://youtu.be/3RI36lsDSg4
- Q11 Virtual Memory: https://youtu.be/5_2fKsK4I34

Anonymous Opossum 2y #1699bdc ✓ Resolved

Why were denorm numbers not included in the calculation for 1.8?

R Rosalie Fang ADMIN 2y #1699bfc

which exam?

Anonymous Opossum 2y #1699bfd

Summer 2019

E Erik Yang TA 2y #1699bff

Replying to Anonymous Opossum

it is included, denorm values have an exponent of 0

♡ ···

A Aarin Salot 2y #1699bdb Unresolved

Q7: Caches

For this question, we will analyze the hit rates of 4 different caches:

- Cache A, a direct-mapped cache
- Cache B, a 4 way set-associative cache with LRU replacement
- Cache C, a fully associative cache with LRU replacement
- Cache D, an N-way set-associative cache with LRU replacement

No matter its type, each cache will have 64B cache blocks with a total capacity of 2 KiB, and have a **write through, no write allocate policy**. We will work with a **32-bit word-aligned**, physical memory space. Integers and floats are **1 word** long. Consider the following program:

```
struct Student {
      int sid;
      float epa;
      char login[3];
      char* ta;
};

#define LEN 128*32
int main() {
      int i;
      struct Student arr[LEN]; // initialized correctly with LEN
                                  Student structs
      // PART I
      for (i = 0; i < LEN; i += 128) {
            arr[i].epa += 10;
            arr[i + 2].epa += 10;
      }
      // PART II
      for (i = 0; i < LEN; i += 128) {
            arr[i + 1].epa = 15;
            arr[i + 3].epa = 15;
      }
}
```

1. How many bytes is the `Student` struct when compiled for the RV32 ISA?
   A. 4 B. 7 C. 8 D. 12 E.15 F. 16
2. If we were to run the for loop, PART I, starting with cold caches, what would be the hit rate of the loop if we used cache A? How about if we used cache B instead?
   `75%; 75%`
3. If we were to run the for loop, PART II, starting with cold caches, what would be the hit rate of the loop using cache C? Write your answer in your answer packet.
   `50%`
4. If we were to use Cache D to run both PART 1 and PART 2 sequentially, starting with a cold cache before PART 1, what value of N would maximize the hit rate? `32. There are only 32 unique blocks being accessed and the cache can only have a maximum of 32 cache blocks.`

11

why would the caches able able to achieve a 75% hit rate in part 2? For every iteration we're doing 2 reads and 2 writes, and 1 read will always be a miss. The 2 writes aren't being stored in the cash since we're doing no write allocate. Should this give us a hit rate of 25%?

♡ ···

Anonymous Mallard 2y #1699bfe

Write allocate applies when we haven't accessed the block before. But each write comes directly after a read of the same block. So we have 1 read and 2 writes that are hits, and 1

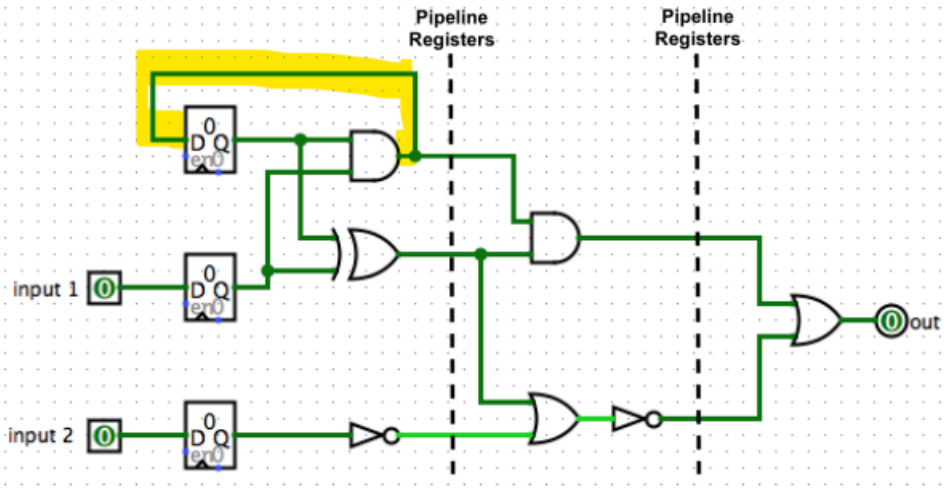
read that is a miss.

♡ ···

Anonymous Swallow 2y #1699bda ✓ Resolved

Fa17 Final Q 5.2

What is the initial value at this node assuming that input 1 starts at 0 at time 0? If we don't know the initial value of the output of the topmost register, is it just 0 since the reset is zero I think? Thank you



A. 0 B. 5 C. 10 D. 15 E. 20 F. 25 G. 30 H. 35

Hold time <= clk-to-q delay + gate delay + setup time = 5+15+0 = 20 ns

♡ ···

R Rosalie Fang ADMIN 2y #1699bfb

Yes if unspecified usually the wires contain the value 0... This really depends on the value in the register usually. We will specify it if needed.

♡ ···

Anonymous Swallow 2y #1699bcf ✓ Resolved

Fa17 Q4.5

Would it be possible to explain why we need USER_IN_2 to be 21? I think line 16 modifies the line jal read_input, but I'm not sure why having USER_IN_2 as 21 would make the code print what is at Skraa. Thank you!

Q4: This question might be a RISC

Consider the following RISC code. The function **read_input** will prompt the user to provide a 32-bit integer and stores it in a0. As a reminder, the **ecall** instruction will call an OS function (determined by the ecall number stored in a0), with the value stored in a1 as the function's argument. **ecall numbers are as follows: 1 = print integer, 4 = print string, 10 = exit.**

```
1.  .data
2.  Boom:  .asciiz "Ayy, man's not dumb." # strlen(this string) == 20
3.  Skraa: .asciiz "The ting goes skkkraaa." # strlen(this string) == 23
4.
5.  .text
6.  MAGIC:          # prologue
7.                  la s0, Risc-tery
8.                  la s1, Boom
9.                  addi s2, x0, 0x61C
10. Get:            jal read_input  # provide either 0 or 1 (USER_IN_1)
11.                 beq a0, x0, Default
12. Risc-tery:      jal read_input  # provide any integer (USER_IN_2)
13.                 beq a0, x0, QuickMaths # Q2
14.                 addi t0, x0, 9
15.                 slli t0, t0, 2
16.                 add s0, s0, t0
17.                 lw t1, 0(s0)
18.                 slli a0, a0, 20   # shift user input by 20
19.                 add t1, t1, a0
20.                 sw t1, 0(s0)
21. QuickMaths:     addi a1, s1, 0
22.                 addi a0, x0, 4
23.                 ecall
24.                 j Done
25. Default:        addi a0, x0, 1
26.                 add a1, s2, x0
27.                 ecall
28. Done:           # epilogue
29.                 jalr ra
```

5. Assume we can both read and write to **any** valid memory address. Please specify the input values to `read_input` such that calling `MAGIC` prints out "The ting goes skkkraaa."
   a) USER_IN_1:
      A. 0 B. 1 C. Not Possible

   b) USER_IN_2:
      A. Any integer B. Any nonzero integer C. 0 D. Exact value ___21_____
      `strlen(string at Boom) + 1 (null terminator) = 21`

♡ ···

R Rosalie Fang ADMIN 2y #1699bfa

Line 21 and 22 prints the string located at s1. Line 8 sets s1 to be the address of Boom, hence we want to adjust the value in s1, which is what Risc-tery is doing. We want to skip all of Boom's string and go to Skraa instead, hence why we skip by 21 bytes.

♡ ···

Anonymous Wolf 2y #1699bce ✓ Resolved

Fa19-Final-Q8b

I am confused why we do not need to have our PTEs byte-aligned.

Q9) We've got VM! Where? **(15 pts = 2 + 3 + 5 + 5*1)**

Your system has a 32 TiB virtual address space with a single level page table. Each page is 256 KiB. On average, the probability of a TLB miss is 0.2 and the probability of a page fault is 0.002. The time to access the TLB is 5 cycles and the time to transfer a page to/from disk is 1,000,000 cycles. The physical address space is 4 GiB and it takes 500 cycles to access it. The system has an L1 physically indexed and tagged cache which takes 5 cycles to access and a hit rate of 50%. On a TLB miss, the MMU checks physical memory next.

a) How many bits is the Virtual Page Number?

27 bits

SHOW YOUR WORK

Number of reachable virtual addresses: log2(32 TiB) = 45
Bits needed to reach all addresses in a page: log2(256 KiB) = 18
So the virtual page number bits are: 45 - 18 = 27

b) What is the total size of the page table (in **bits**), assuming we have **no** permission bits or any other metadata in a page table entry, just the translation?

14×2^{27} bits

SHOW YOUR WORK

We need to figure out the number of bits in the physical page number. It is the same method except we use the physical address space:
Number of reachable physical addresses: log2(4 GiB) = 32
So PPN size is 32 - 18 = 14. We do not have any metadata bits so the total number of bits in a PTE is 14. To figure out how many entries we need, we need to look at the total number of virtual page numbers we have = 27. This means we need 2^{27} entries in the page table. This means we need a total of 14×2^{27} bits in our page table.

♡ ···

R Rosalie Fang ADMIN 2y #1699bef

This is a specific course offering that didn't care about alignment. On the final, we will indicate how we want you to align your addresses.

♡ ···

Anonymous Lark 2y #1699bcd ✓ Resolved

Su19, Q1 - Part 8: I understand why we are looking at the case of 1. * 10 ^ (exponent that is less than 1023), but why aren't we also looking at the case of 0.mantissa * 2(-1022)

♡ ···

R Rosalie Fang ADMIN 2y #1699bee

We are looking at all those cases... the solution doesn't discount denormalized numbers. There're still 4 significand bits for the denormalized numbers.

♡ ···

Anonymous Lark 2y #1699bcc ✓ Resolved

Su19, Question 1 - Part 11: The person keeps saying how if the real exponent is 0, then we can represent the values [2^0, 2^1]. I wanted to run my explanation to see if I'm understanding it correctly. Would it be 1.mantissa * 2^0, so would it range from 1.0 * 2^0 to 2 * 2^0, which would be [1, 2]?

♡ ···

R Rosalie Fang ADMIN 2y #1699bed

Yes. It would range from 1.0 * 2^0 to 1.0 * 2^1.

♡ ···

Anonymous Lark 2y #1699bcb ✓ Resolved

In SU19, for Q1 problem 7 on converting 4.75 in our new floating point scheme, when figuring out the exponent, I'm a little confused. I thought that in order to convert our real exponent to the way we want to store it, we would subtract the bias, which is why in the lecture slides when we had a bias= -127, to convert from real exponent to stored exponent, we would take our real exponent - (-127) to get our stored version. But here the bias is positive, so I'm confused.

♡ ···

R Rosalie Fang ADMIN 2y #1699bec

This question might be from an offering of the course where the bias is positive. Please treat it as if it's negative.

♡ ···

Anonymous Camel 2y #1699bca ✓ Resolved

sp 18 f q1c: how do we approach this question?

(c) Given the following function in C:

```
int shifter(int x, int shift) {
    if (x > 0) {
        return x >> shift;
    }
    return -1 * (x >> shift);
}
```

Given y is a negative integer, and that `shifter(y, 2)` outputs 4, what is the range of values of y?

hint: `-8 >> 1 = -4`

Solution: -16 ∼ −13

♡ ···

R Rosalie Fang ADMIN 2y #1699beb

We know y is a negative integer, so we know

```
-1 * (y >> 2) = 4; so y >> 2 = -4
```

At this point, I would write out the bits of y. -4 is 111111...100, so y has to be 111....100 _ _, where the bottom 2 bits can be anything. And you convert to an integer from this 2's complement representation.

♡ ···

Anonymous Camel 2y #1699bbf ✓ Resolved

sp18 f q5: How do we get 30 or 41? Aren't there 2 pipelined stages so it would be 11*2?

(d) How long does it take to compute the output for a given set of inputs? Assume the clock period is 11ps.

- ○ 22ps
- ○ 27ps
- ○ 28ps
- ○ 31ps
- ○ 42ps
- ○ 47ps
- ○ 50ps
- ● other: 30ps or 41 ps

Solution: Technically, 30ps is the time it will take for the values of one set of inputs to propogate to the output, We also accepted 41ps (3 clock cycles + clk-to-q + adder)

♡ ···

R Rosalie Fang ADMIN 2y #1699bea

It's not exactly 2 pipelined stages...

It takes 1 clock cycle to go from input -> first set of registers, another clock cycle to go from first set of registers to second set of registers, and then to get the stable output you have to account for the time for register 2/3's value to travel to the output, therefore it's 2 * 11 + clk-to-q + adder.

♡ ···

Anonymous Tarsier 2y #1699bbe ✓ Resolved

Su19 MT2 Q4 : Why is there a data hazard here?

`j loop` `beq t0 a2 end`	[X] yes [ ] no	[X] Must Stall FROM: [ ] ID [ ] IF [ ] EX1 [ ] MEM [ ] EX2 [ ] WB TO: [ ] ID [ ] IF [ ] EX1 [ ] MEM [ ] EX2 [ ] WB

♡ ···

R Rosalie Fang ADMIN 2y #1699bdf

Here it's not a data hazard but a control hazard. This is because in a pipeline, we don't know the next line to execute until the EX stage. therefore the lines right after `j loop` will be fetched into the pipeline if we don't stall.

♡ ···

Anonymous Swallow 2y #1699bbd ✓ Resolved

Fa19 Q5d

Why does this simplify to B!C(A+1) instead of B!C(!A+1)? Thank you

d) Draw the **FULLY SIMPLIFIED** (*fewest* primitive gates) circuit for the equation below into the diagram on the lower right. You may use the following primitive gates: AND, NAND, OR, NOR, XOR, XNOR, and NOT.

$out = \overline{(C + AB\overline{C} + \overline{B}\,\overline{C}D)} + \overline{(C + \overline{B + D})}$ **SHOW YOUR WORK IN THIS BOX**

out = $\overline{C}\,\overline{(AB\overline{C})}\,\overline{(\overline{B}\overline{C}D)} + \overline{C}(B + D)$ **(Demorgan's)**

out = $\overline{C}(\overline{A} + \overline{B} + C)(B + C + \overline{D}) + \overline{C}(B + D)$ **(Demorgan's)**

out = $(\overline{A}\,\overline{C} + \overline{B\overline{C}} + \overline{C}C)(B + C + \overline{D}) + B\overline{C} + \overline{C}D$ **(Distributive)**

out = $(\overline{A}\,\overline{C} + \overline{B\overline{C}})(B + C + \overline{D}) + B\overline{C} + \overline{C}D$ **(Inverse)**

out = $\overline{A}B\overline{C} + \overline{A}C\overline{C} + \overline{A}\,\overline{C}\,\overline{D} + B\overline{B\overline{C}} + BC\overline{C} + \overline{B}\,\overline{C}\,\overline{D} + B\overline{C} + \overline{C}D$ **(Distributive)**

out = $\overline{A}B\overline{C} + \overline{A}\,\overline{C}\,\overline{D} + \overline{B}\,\overline{C}\,\overline{D} + B\overline{C} + \overline{C}D$ **(Inverse)**

out = $B\overline{C}(A + 1) + \overline{A}\,\overline{C}\,\overline{D} + \overline{B}\,\overline{C}\,\overline{D} + \overline{C}D$ **(Distributive)**

out = $\overline{C}(B + \overline{A}\,\overline{D} + \overline{B}\overline{D} + D)$ **(Distributive)**

out = $\overline{C}((B + \overline{B}\,\overline{D}) + (\overline{A}\,\overline{D} + D))$ **(Associative)**

out = $\overline{C}(B + \overline{D} + \overline{A} + D)$

♡ ···

R Rosalie Fang ADMIN 2y #1699bde

That looks like a typo in the solutions

♡1 ···

Anonymous Raccoon 2y #1699afa ✓ Resolved

Sp18 Final 10d

(d) Given a message of length `n` characters, how many instructions are needed after loop unrolling? Express your answer in terms of `n`, such as `3n + 4`. In addition, what is the speed up when `n` is approaching infinity in comparison to the **original non-optimized function** `obfuscate`? Count pseudo-instructions as 1 instruction. You do not need to simplify your expressions.

# of Instructions: $(7 + (n/8) * 10 + 1 + (n\%8) * 6 + 1)$ Speedup: 7.5X

How does it get 7.5X? I think the original # of instructions is 2 + 6n + 1.

♡ ···

R Rosalie Fang ADMIN 2y #1699bbc

When n approaches infinity, the constants (such as 2 + 1 in the original # of instructions, or 7 + ... + 1... in the unrolled version) because unimportant and we only look at the parts that strictly depend on n. Therefore, you're looking at the speedup we got from 6n -> (10/8) n, which is 6/(10/8) which is... supposed to be 4.8X I think?

♡ ···

Anonymous Swallow 2y #1699aea ✓ Resolved

SU18-Final-Q6

Why does the no write allocate policy mean that no part of C enters the cache? How does write allocate policy affect the hit rate? Thank you!

The following piece of code is executed on the aforementioned machine. This code computes an outer product of a **N x 1** vector A and a **1 x N** vector B, placing the result in a **N x N** matrix C. Use this code to answer the follow questions about the hit rate the code was produced.

For all questions assume the following:

- sizeof (double) == 8
- A = 0x10000
- B = 0x20000
- C = 0x30000
- The cache begins cold before each question.
- Code is executed from left to right.

```
#define N 16

void outer_product (double *A, double *B, double *C) {
      for (int i = 0; i < N; i++) {
                  for (int j` = 0; j < N; j++) {
                              C [j + i * N] = A[ i ] * B[ j ];
                  }
      }
}
```

3. What is the hit rate for executing this code if it uses LRU replacement and is a write back cache with no write allocate on a miss? Fill in all blanks for credit.

HR for accesses to A: 63/64

HR for accesses to B: 63/64

HR for accesses to C: 0

OVERALL HR: 63/64

Explanation: There are many ways to do this question. The easiest barely looks at the access patterns. Since C is a different address from A and B, no part of C will ever enter the cache. The total size of the cache is 256 B, A contains 128 B and B contains 128 B so A and B both fill the cache. Each element of A and each element of B is accessed 16 times, so each Vector is accessed 256 times with only 4 compulsory misses.

As a result,

A = 252/256 = 63/64
B = 252/256 = 63/64
C = 0

♡ ...

R Rosalie Fang ADMIN 2y #1699bbb

No write allocate policy means that when we write to a memory address, we write directly to the address instead of going through the cache.

If we were to have a write-back policy though, for example, then we will keep the cache line and there will be cache hits since the line is already in the cache.

♡ 1 ···

Anonymous Swallow 2y #1699adf ✓ Resolved

Su18-Final-Q3

Hi, how do we get the answers for XYZ? I think in order for it to be resolved in assembler we need it to be a relative offset but I dont know how to calculate it. Thank you!

> Suppose that the label `Q1` is at address 0x4000 0000. If the label `end` is at address 0x40**XY Z**800, what are all the possible values for **X**, **Y**, and **Z** such that `j end` can be resolved in the assembler? Formulate your answer in the form `[A - B]` where `A` and `B` are both hexadecimal digits.
>
> **X**: 0
> **Y**: [0 - F]
> **Z**: [0 - F]

♡ ···

R Rosalie Fang ADMIN 2y #1699bba

it's true that it needs to be a relative offset. If you look on the reference card, jal only takes in immediate up to the 20-th bit, with the 20-th bit being the sign bit. Since we want this to be a positive offset, we know the 20-th bit, in this case X's lowest bit, needs to be 0. And Y and Z can be whatever. If we have a relative offset larger than that, we would have to do some kind of lui to be able to resolve addressing, or wait for the linker.

♡ 1 ···

Anonymous Llama 2y #1699ace ✓ Resolved

I'm confused as to what this question is asking for and how you would need two pages for code and histogram?

Spring 2013 #F2:

```
int histogram[MAX_SCORE];
void update_hist(int *scores, int num_scores) {
      for (int i = 0; i < num_scores; i++)
            histogram[scores[i]] += 1;
}
```

d) In the best-case scenario, how many iterations of the loop can occur before a TLB miss? You can leave your answer as a product of two numbers.

Iterations = $30*2^{18}$

- Firstly, your TLB has 32 entries, one per page.
  - You need at least one for code, scores, and histogram.
- Now, continuing being nice, what if score[i] was always the same number?
  - Then we'd only need one page for histogram (and only one page for code)
- This leaves 30 pages to iterate through the score array!
  - Each page can hold $(2^{20} / 2^2) = 2^{18}$ ints, so we can iterate for **$30*2^{18}$ i values**

HKN Spring 2023

♡ ···

R Rosalie Fang ADMIN 2y #1699baf

We need one page for code because the code section (try converting this loop logic into RISC-V then binary) doesn't take that many bytes so can definitely fit into a 2^20 byte page.

We need one page for the histogram because, assuming that score[I] always give you the same value, so we keep accessing the same element of histogram[i], therefore, only one page needs to be allocated because none of the other addresses will actually get accessed.

♡ ···

Anonymous Lark 2y #1699abc ✓ Resolved

Fa18-Final-QF4a

It just seems out of the box to assume that a lookup table is one of the options since the given "schemes" are number representations. Is this question just asking what would use the least bits? Not sure if this question is in scope, since I don't recall what fixed point implies and had to look it up and apparently it's where you can have a certain number of digits after the decimal point? Just curious why wouldn't unsigned fixed point be an answer?

F4) *What's that smell?! Oh, it's Potpourri...* (20 points=2 each, 30 minutes)

a) We build a small Internet-of-things device to measure dog body temperature and send it to a receiver. It will only send the following temperatures: {100.0, 100.1, 100.2, 100.4, 100.8, 101.6, 103.2, 106.4}, and any time the temperature is not those exact values, it'll send whatever value is the closest one. What encoding/decoding *scheme* would you use for these numbers and how many total *bits* would you need?

Scheme: ○Unsigned fixed point ○Bias fixed point ○2s complement fixed point ●Other

Bits: ●3 ○4 ○5 ○6 ○7 ○8 ○9 ○10 ○11 ○12 ○13 ○14 ○15 ○16

Other (just have a lookup table) using **3 bits** to choose from the 8 values

♡ ···

R Rosalie Fang ADMIN 2y #1699bae

I do agree that it's out of the box and most times I don't think we'll ask you to come up with it... in fact I don't think we did it again in later exams.

Unsigned fixed point cannot be used because the question asked "total bits" meaning this is a binary notation, so even though the decimal notation looks really good, it actually doesn't translate well to binary notation (how to represent 0.2 in binary?)

♡ ···

Anonymous Lark 2y #1699abb ✓ Resolved

Fa18-Final-QF3

Question about control hazards with branches. It seems we will always have a control hazard with 2 NOPs with branches because we do not know whether it is jumping or not until the EX (execute) stage? So it will **always** be 2 NOPs, or am I misunderstanding something?

```
start: lw    t0, 0(a0)          Hazard (circle one): S D C None   # of nop _____2_____
```
t0 being written to and read from the next op causes a data hazard and 2-cycle stall (or introduction of 2 NOPs so the W and R of the register file lines up -- thankfully we can read and write registers the same cycle)

```
       beq   t0, x0, end        Hazard (circle one): S D C None   # of nop _____2_____
```
We need to wait until after the EX stage to know whether t0==x0 before we can load the correct next instruction, so that causes a control hazard and a 2-cycle stall (or introduction of 2 NOPs)

```
       addi  t0, t0, 2          Hazard (circle one): S D C None   # of nop _____2_____
```
t0 being written to and read from the next op causes a data hazard and 2-cycle stall (or introduction of 2 NOPs so the W and R of the register file lines up -- thankfully we can read and write registers the same cycle)

```
       sw    t0, 0(a0)          Hazard (circle one): S D C None   # of nop _____________
end:
```

inst	1	2	3	4	5	6	7	8	9	10	11	12	13	14
lw t0 0(a0)	IF	ID a0	EX	MEM r	WB t0									
beq t0,x0,end		NOP	NOP	IF	ID t0	EX	MEM	WB						
addi t0, t0, 2					NOP	NOP	IF	ID t0	EX	MEM	WB t0			
sw t0, 0(a0)								NOP	NOP	IF	ID t0 a0	EX	MEM w	WB

♡ ···

R Rosalie Fang ADMIN 2y #1699acd

If we're talking about the number of stalls needed to ensure no control hazards, sure your logic is correct. However, if the branch instruction happen to be an instruction that never takes the branch, we don't have to stall at all.

♡ ···

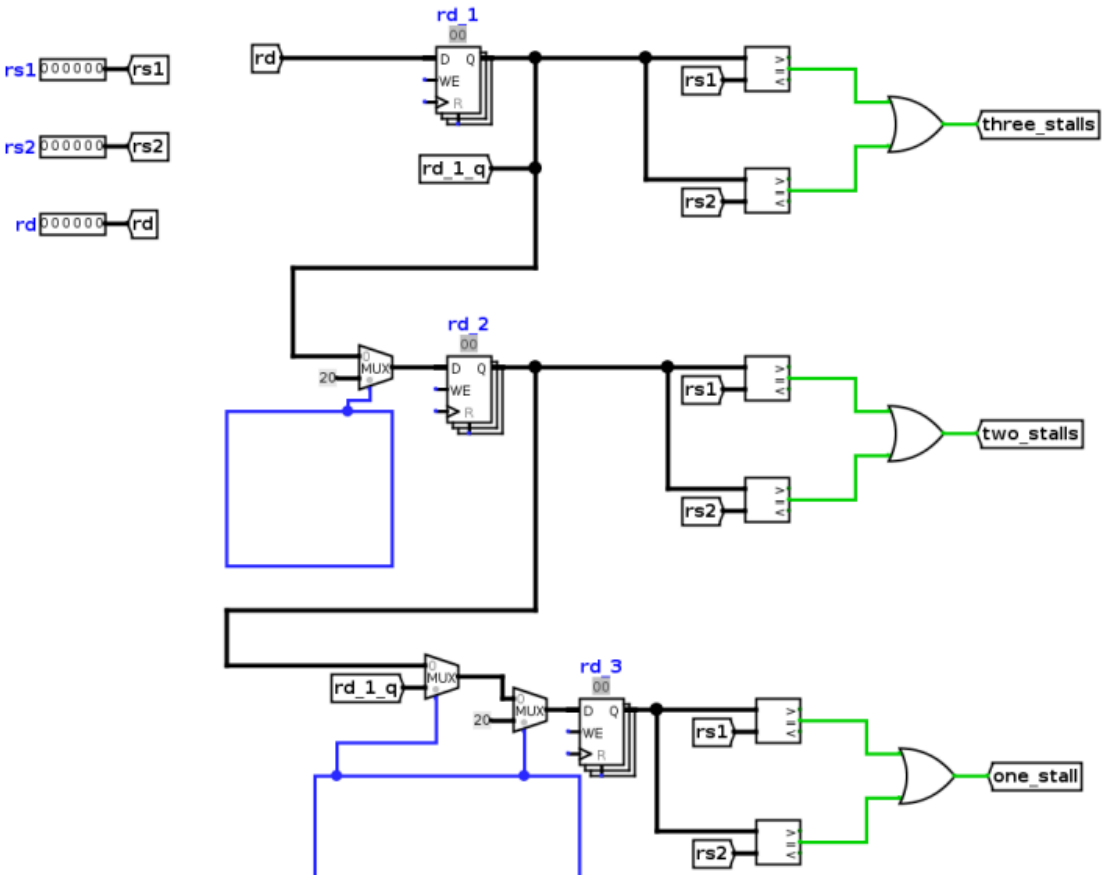
Anonymous Raccoon 2y #1699aba ✓ Resolved

SP19 MT2 7

Problem 7 ***Stalling*** **(18 points)**

The missing part of a basic five-stage RISC-V pipeline before we added forwarding that we dont cover in lecture is the hardware logic that decides when to insert stalls if we have a data hazard. Lets put it together now! Heres an outline of a hardware unit that will decide if we need a stall due to a data hazard in that youll fill in the missing pieces for. Each clock cycle, it takes in the 5-bit signals for rd, rs1, and rs2 that come into the Instruction Decode phase and it outputs the number of stalls that are necessary. For this question we will assume all of our instructions are R-Type Instructions, every instruction will write to rd and use rs1 and rs2. Additionally we assume that you cannot read and write in the same clock cycle, which means you could need to stall as many as 3 clock cycles.

In the diagram below you will see that we store the value of rd for each of the past 3 instructions, which are from top to bottom, $rd[i-1]$, $rd[i-2]$, and $rd[i-3]$ respectively.



I'm really confused about how to approach this problem. I don't get what's the function of constant **20** and why rs1 and rs2 are comparing with each register outputs.

♡ ···

R Rosalie Fang ADMIN 2y #1699acc

The premise of this problem is data hazard detection. So if you look at the top logic, you're comparing rd (after going through a register) to rs1 and rs2. So this basically compares rd of instr1 to rs1/rs2 of instr2. And then it goes through another register, and compares again with rs1 and rs2, which is basically comparing rd of instr1 to rs1/rs2 of instr3, so on.

The **20** is in hex so it's actually 0b00100000, which is not a valid register. The wire is hardcoded to always pick the 0 I believe, so the first **20** is just there as a placeholder.

♡ ···

Anonymous Raccoon 2y #1699adb

I don't get why compare rd to rs1/rs2. Why is it associated with determing the number of stalls? Could you give me more details about this? Thanks :)

♡ ···

R

Rosalie Fang **ADMIN** 2y #1699ade

Replying to Anonymous Raccoon

Think about the condition of a data hazard: it's when the value needed is not updated.

For example, code that looks like

```
addi s0 x0 1
addi t0 s0 4
```

creates data hazard between the rd of the first instruction and rs1 of the second instruction.

and code like

```
addi s0 x0 1
addi x0 x0 0
addi t0 s0 5
```

creates data hazard between rd of the first instruction and rs1 of the third instruction.

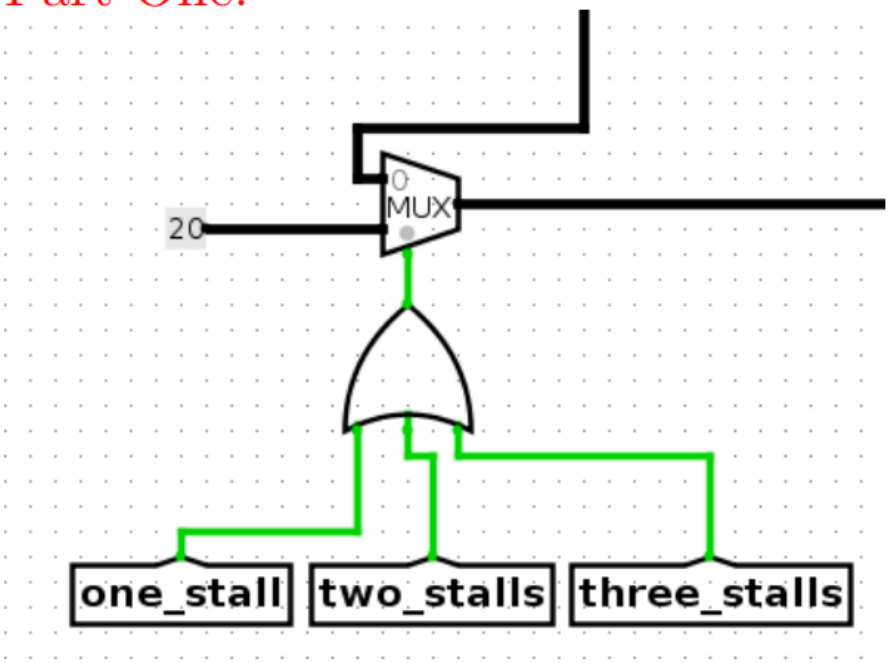
♡ ···

Anonymous Raccoon 2y #1699aed

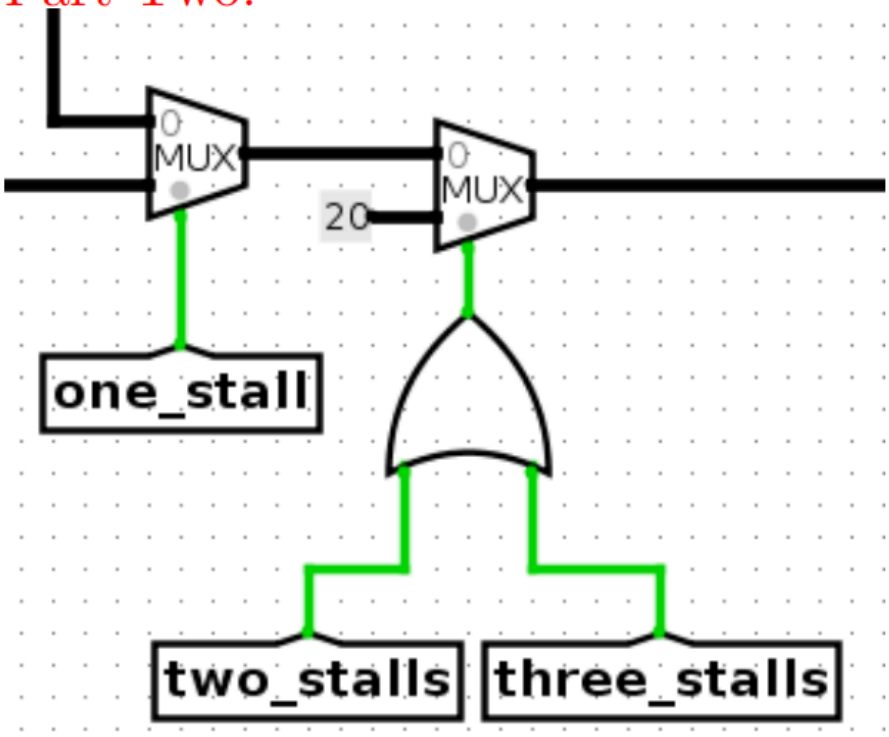
Replying to Rosalie Fang

Solution:

Part One:



Part Two:



Now I understand why we need to compare but still don't understand the logic here. Could you give me more explanations?

Thanks!

♡ ···

R

Rosalie Fang ADMIN 2y #1699bad

Replying to Anonymous Raccoon

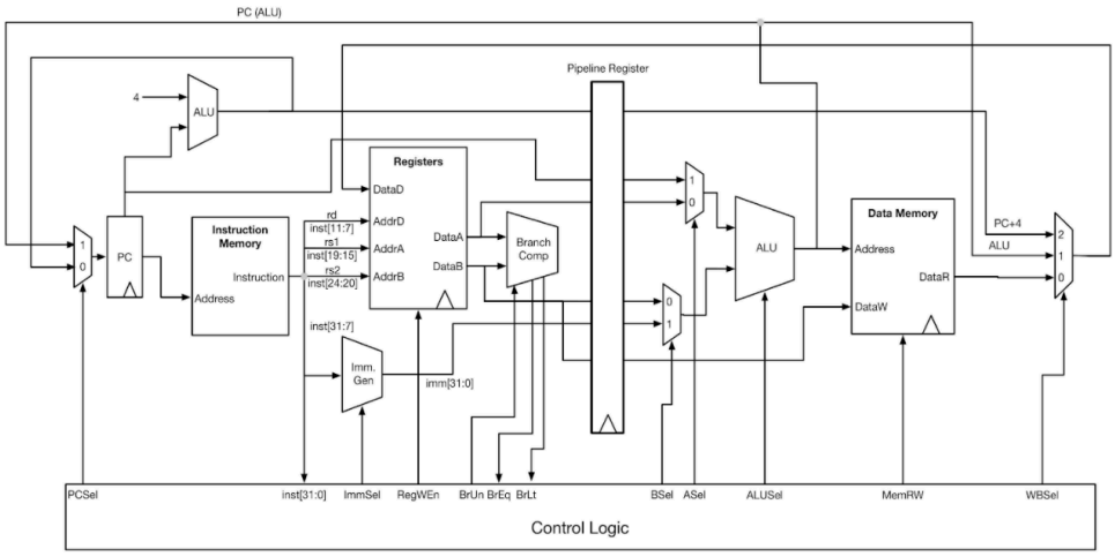
For part 1, we're trying to see if our instruction should be executed, or if we need to stall. With rd_1, we compared the rd of instr1 with rs1/rs2 of instr2, so if there're any stall needed at all, we need to insert a nop, hence why it's simply an or.

For part 2, we're now comparing rd of instr1 with rs1/rs2 of instr3, so we only stall if there's at least 2 stalls. Also, if there was 1 stall, we already inserted a nop in part 1, so we now need to execute that instruction from a tilmestep ago.

♡ ...

Anonymous Raccoon 2y #1699aaf ✓ Resolved

SP18 MT2 4d



In this 2 stage piplined datapath, PCSel signal is set in which stage?

♡ ...

R Rosalie Fang ADMIN 2y #1699acb

PCSel depends on 1. the instruction, 2. the branch comparator. Since both parts are determined in stage 1, we can obtain the correct PCSel in stage1.

♡ ...

Anonymous Raccoon 2y #1699aeb

I'm still confused about why REgWEn is determined by stage 2.

I think it can be updated as soon as the instructions are parsed.

♡ ...

R Rosalie Fang ADMIN 2y #1699bab

Replying to Anonymous Raccoon

It's true that RegWEn is obtainable once the instructions are parsed, however RegWen should correspond to the data that it's trying to write into rd. Since that data is not ready at stage 1, we need RegWEn to be in stage2.

♡ 1 ...

Anonymous Penguin 2y #1699aad ✓ Resolved

fa 19 last prob, h: is there a way we should the conversions from million/billion/trillion to gibi/tebi etc? is there a chart or something given to us on the ref sheet, or are we expected to just know it .

h) You are designing a 64-bit ISA for a simplified CPU with 3 bit-fields: `immediate | register | opcode`. You reserve enough of the rightmost bits to handle 1,500 opcodes, and enough of the leftmost bits to encode unsigned numbers up to 500 trillion. What's the greatest number of registers can you have?

16	**SHOW YOUR WORK** 1500 opcodes requires 11 bits (2048) 500 trillion requires 49 bits (512 Tebi) 64 - (11 + 49) = 4 bits for registers 2^4=16

♡ ···

R Rosalie Fang **ADMIN** 2y #1699aca

It's kind of given on the official 61C reference card. Go check it out and see if that's sufficient conversion for you (it maps 10^3 to ~2^10, 10^6 to ~2^20, etc.)

♡ ···

Anonymous Penguin 2y #1699aac ✓ Resolved

is fall 19 last prob part g in scope?

g) You have an SSD which can transfer data in 32-byte chunks at a rate of 64 MB/second. No transfer can be missed. If we have a 4GHz processor, which takes 200 cycles for a polling operation, what fraction of time does the processor spend polling the SSD drive for data? Leave your answer in the box provided as a percentage.

10%	**SHOW YOUR WORK** 64 MB/sec / 32 B/poll = 2M polls/sec 2M polls/sec * 200 cycles/poll = 400M cycles/sec 4 GHz clock ⇒ 4000M cycles/sec; 400M c/s / 4000M c/s = 10%

♡ ···

R Rosalie Fang **ADMIN** 2y #1699abf

Technically I don't believe this is out of scope since there is not really formulas for questions like this but rather just understanding the high-level ideas of data transferring.

♡ ···

Anonymous Penguin 2y #1699aab ✓ Resolved

fa19 q8b: why is i+256 a conflict miss? since we have 4 offset bits, each entry in the cache should be less than 16 bits apart, but 256>16?

Q8) This is for all the money! **(15 pts = 3 + 7 + 5)**

Assume we have a single-level, 1 KiB direct-mapped L1 cache with 16-byte blocks. We have 4 GiB of memory. An integer is 4 bytes. The array is block-aligned.

```
#define LEN 2048

int ARRAY[LEN];
int main() {
    for (int i = 0; i < LEN - 256; i+=256) {
       ARRAY[i] = ARRAY[i] + ARRAY[i+1] + ARRAY[i+256];
       ARRAY[i] += 10;
    }
}
```

a) Calculate the number of tag, index, and offset **bits** in the L1 cache.

T:22	I:6	O:4

SHOW YOUR WORK

Offset: log2(block size) = log2(16) = 4
Index: log2(cache size / block size) = log2(1 KiB / 16) = log2(64) = 6
Tag:
First find total address bits log2(4 GiB) = log2(4 * 2^30) = log2(2^32) = 32
Then 32 - Index - Offset = 32 - 6 - 4 = 22

b) What is the hit rate for the code above? Assume C processes expressions left-to-right.

50%

SHOW YOUR WORK

Every iteration it's
ARRAY[i] read MISS
ARRAY[i+1] read HIT
ARRAY[i+256] read CONFLICT→ MISS
ARRAY[i] write CONFLICT→ MISS
ARRAY[i] read HIT
ARRAY[i] write HIT
3 MISSES, 3 HITS. 50% hit rate.

♡ ···

R Rosalie Fang ADMIN 2y #1699abe

#1699ee

♡ ···

Anonymous Penguin 2y #1699aaa ✓ Resolved

fa19 q7, case 2c: why can't you forward execute to execute? you can take s0 from the first instruction's execute stage, and forward to the execute state of the third instruction so that bge always has the right value?

Case 2: *After fixing that hazard,* the following case fails:

```
      addi  s0 x0 4
      slli  t1 s0 2
      bge   s0 x0 greater
      xori  t1 t1 -1
      addi  t1 t1 1
greater:
      mul t0 t1 s0
```

When this test case is run, `t0` contains `0xFFFFFFC0`, which is not what it should have been.
Pro tip: you shouldn't even need to understand what the code does to answer this.

c) What caused the failure? (select ONE)
- ● Control Hazard
- ○ Structural Hazard
- ○ Data Hazard
- ○ None of the above

d) How could you fix it? (select all that apply)
- ■ Insert a nop 3 times if you detect this specific error condition
- ☐ Forward execute to write back if you detect this specific error condition
- ☐ Forward execute to memory if you detect this specific error condition
- ☐ Forward execute to execute if you detect this specific error condition
- ■ Flush the pipeline if you detect this specific error condition

♡ ...

Anonymous Crane 2y #1699aee

Adding to this, how are the first two lines not a data hazard, since we try to use s0 before the WB of the addi instruction finishes.

♡ ...

R Rosalie Fang ADMIN 2y #1699aff

the first 2 are a data hazard, but this is case 2 so this is after fixing the data hazard.

♡ ...

R Rosalie Fang ADMIN 2y #1699baa

We want to be able to generalize this control hazard case into others. The problem here isn't just forwarding, in fact control hazards can't be solved by forwarding at all. This is because without any implementation to deal with control hazards, we will only know whether we take the branch or not in the EX stage, by then the 2 instructions immediate after it would've started executing already. But if we take the branch we don't want those instructions to execute. Hence why only nop/flushing the pipeline would work.

♡ ...

 Stella Kaval 2y #1699ff ✓ Resolved

How is this fully associative?

Problem 12 ***[F-4] Virtual Memory*** **(20 points)**

Demand paging (storing part of a process' memory on disk) is yet another example of caching in computer systems. If we think of main memory as a cache for disk, what are the properties of this cache? Assume a machine with 64 bit addresses, 16KB pages, a 4-way fully associative TLB, and 8B words.

(a) Associativity?

○ Direct Mapped ● Fully Associative

○ N-Way Set Associative

(b) Block size:

Solution: 16KB

(c) Address layout. Your answer should be of the form [N:M] where N is the bit number of the most significant bit of the field and N is the bit number of the least significant bit of the field. For example, if the tag consists of the first 4 least-significant bits, you should write [3:0]. If the field is not applicable to paging, you may write "N/A".

Solution: Tag bits:[63:14] Index bits:N/A Offset bits:[13:0]

♡ ...

R Rosalie Fang ADMIN 2y #1699abd

Notice that we never specified the size of the TLB!! When we say "fully-associative" we mean there is 1 set, and 4-way means there are 4 entries per set. So this just says that we have a total of 4 entries in the TLB.

♡ ...

Anonymous Tarsier 2y #1699fe ✓ Resolved

Su18 11.9 Why did we not replace the second row in the TLB? At this point, aren't 0X6 and 0X5 in the TLB so we'd replace the smaller VPN (row 2)?

Also why does the valid bit for 0x1 in the Page Table change to 0?

♡ ···

R Rosalie Fang ADMIN 2y #1699afe

#1699afd ;-;

♡ ···

Anonymous Tarsier 2y #1699bac

slay thanks Rosalie 🫶

♡ ···

N Neeraj Rattehalli 2y #1699de ✓ Resolved

FA19-Final-Q8b

Why is array[I] read and array[I] read 2 separate cache misses. Isn't array[I] block loaded into cache immediately after read?

♡ ···

R Rosalie Fang ADMIN 2y #1699fd

#1699ca

♡ ···

Anonymous Hamster 2y #1699cf ✓ Resolved

Sp19-Final-Q6a

Wouldn't line 5 also error since size is of type size_t whereas i is of type int?

♡ ···

R Rosalie Fang ADMIN 2y #1699fc

This would be a compiler warning, not an error. size_t is just uint8_t usually so comparison is still allowed.

♡ ···

Anonymous Swallow 2y #1699ce ✓ Resolved

Su18-Final-Q12

Hello, why do we use burst mode for playing the video? Is it not better to have continuous transfer of the data since we want a smooth-playing video? Or would we rather have a buffer?

Consider adding a DMA controller to facilitate bringing in data from our external disk/devices into main memory. The DMA controller can either make transfers in **burst mode**, where it stalls the CPU and writes 32B of data per clock cycle, or **cycle-stealing mode** where it doesn't stall the CPU (i.e. it runs normally) and writes 4B of data per clock cycle. The CPU returns to normal instruction execution after the transfer is completed. Which mode would be better for the following situations? Assume every instruction uses 1 clock cycle with no stalls.

- Dealing with page fault; the page size is 4KiB — Ⓐ burst — Ⓑ cycle stealing
- Playing a 200MB 5-minute video — Ⓐ burst — Ⓑ cycle stealing
- Processing 10 keyboard strokes per second — Ⓐ burst — Ⓑ cycle stealing

♡ ···

R Rosalie Fang ADMIN 2y #1699fb

It's mostly because the data we want to transfer is really big. stalling the CPU doesn't mean we don't have a smooth-playing video necessarily, but here burst mode is critical to transferring large amount of data.

♡ ···

Anonymous Dinosaur 2y #1699cd ✓ Resolved

Su18-Final-Q11.9

I'm not sure how caches deal with page fault. Why does the TLB and Page table now map 0x2 to 0x2, and why did the 0x1 in the page table turn invalid?

9. 0x2F4

TLB

VPN	PPN
0x2	**0x2**

PAGE TABLE

VPN	Valid Bit	PPN
0x1	**0**	**X (doesn't matter)**
0x2	**1**	**0x2**

TLB

VPN	PPN
0x1	0x2
0x5	0x3

PAGE TABLE

VPN	Valid Bit	PPN
0x0	0	0x3
0x1	1	0x2
0x2	0	0x1
0x3	1	0x0
0x4	0	0x0
0x5	1	0x3
0x6	1	0x1
0x7	0	0x2

♡ ···

R Rosalie Fang ADMIN 2y #1699afd

tbh I'm not sure. don't worry about it for now maybe?

♡ ···

Anonymous Porcupine 2y #1699cc ✓ Resolved

Fa19-Final-Q6

I'm not sure how exactly immediate generator not extending yields the given output.

For your CPU project, you followed the datapath diagram we gave you exactly and built a single-cycle CPU. However, something is not working correctly. *All instructions besides some of the I-types and SB-types are working.* You start by testing with an **`addi a0 a0 -3`** instruction. The **`a0`** register initially holds a value of **7** and all other registers initially hold **0**. This instruction is stored in IMEM at address **`0x00000004`**. DMEM reflects the initial IMEM. Undefined ImmSel outputs an I-type imm. You put a probe at the data read from IMEM and find the instruction is correct. You next put a probe at **wb**, and see the output is `0b0000_0000_0000_0000_0001_0000_0000_0100` (**`0x00001004`**).

a) Since the output is incorrect, what errors alone would cause the erroneous behavior? (select all that apply)

☐ The RegWEn is set to false.	☐ PCSel is set to PC + 4.
■ The Immediate Generator is not sign extending.	☐ The writeback MUX is selecting PC + 4.
☐ Read Reg1 and Read Reg2 are flipped.	☐ MemRW is set to write.
☐ The writeback MUX is selecting DMEM.	☐ BSel is selecting rs2 and not imm.

a) RegWEn does not change the state of wb since this is the only inst.
b) This is the issue!
c) If the two registers were flipped, we would get a different value since we would find we would be using a register with 0 in it thus would be getting back 0xFFC thus these are actually correct.
d) This is not correct since we would be getting the machine code of the current instruction since the correct address would be 0x4 which is this instruction. Thus this is not a root issue.
e) We would not be able to detect an issue here by looking at wb since this does not affect the wb of the current program.
f) If the write back mux was set to PC + 4, we would receive an output of 0x0000000c thus this is not correct either.
g) MemRW will not change the output of wb. Even if we were outputting the read of the address, it would be incorrect.
h) If BSEL selected RS2, we would have selected t6 which we know has a value of 0 and would have had 0 + 5 which is not the result we got.

For option 4, do we get machine code reading DMEM because it carries forward what's read from IMEM

For option 6, why does PC+4 give 0xC

For option 8, why rs2 (value -3) gives register t6, isn't 11101 = 29 => t4?

♡ ···

R Rosalie Fang ADMIN 2y #1699fa

- Option 4: it mentioned that the DMEM reflects the initial IMEM, so if write back MUX is selecting DMEM we would see machine code of the instruction addi a0 a0 -3 which is not the output shown
- Option 6: The instruction is stored in IMEM at address 0x00000004, that means PC=4, I think there's an error, but PC+4 should give 0x8.
- Option 8: Also typo I think... yeah it'd be t4.

♡ ···

Anonymous Lark 2y #1699cb ✓ Resolved

Fa18-MT-Q3C (https://tbp.berkeley.edu/exams/6323/download/)

Can I have some clarification on how to calculate the maximum value to get 63, the explanation's logic is kind of hard to follow.

c) Let **`A = 0x100061C0`**. If the cache has a hit at **`i=0`** in the loop, what is the maximum value returned by **`random`**? ___63___
If we look at the 8 bits of offset, the value if C0, which is 0b1100 0000, (¼ of the way from the left since the top two bits are 11) so any value of random between 0 and 0b11 1111 would leave us in that block (before the 8 bit offset bubbles over, and another index is randomly touched), and 0b11 1111 is **63**.

♡ ···

R Rosalie Fang ADMIN 2y #1699ef

cache has a hit at `i = 0` means that touch(A) must've accessed the same block as `A[0]`. Since block size of 256B, that means offset takes 8 bits. So when we access 0x1000 61C0, what gets loaded in the cache is all the data from 0x1000 6100 - 0x1000 61FF. Therefore, what we need to find is maximum value of `x` such that `&A[x] < 0x1000 61FF`, and since each integer is 1 byte because A is a uint8_t array, the largest address of the integer is located at 0x1000 61FF. Calculating the difference between 0x1000 61FF and the head of the array at 0x1000 61C0 we get 0x3F = 3 * 16 + 15 = 63 (divide by the size of each element which is 1 byte, it's still 63).

♡ ···

Anonymous Llama 2y #1699ca ✓ Resolved

FA19-Final-Q8b

Why is ARRAY[i + 256] a conflict miss? Wouldn't ARRAY[0] block and ARRAY[256] have different indices?

```
#define LEN 2048

int ARRAY[LEN];
int main() {
    for (int i = 0; i < LEN - 256; i+=256) {
       ARRAY[i] = ARRAY[i] + ARRAY[i+1] + ARRAY[i+256];
       ARRAY[i] += 10;
    }
}
```

SHOW YOUR WORK

Every iteration it's
ARRAY[i] read MISS
ARRAY[i+1] read HIT
ARRAY[i+256] read CONFLICT→ MISS
ARRAY[i] write CONFLICT→ MISS
ARRAY[i] read HIT
ARRAY[i] write HIT
3 MISSES, 3 HITS. 50% hit rate.

♡ ···

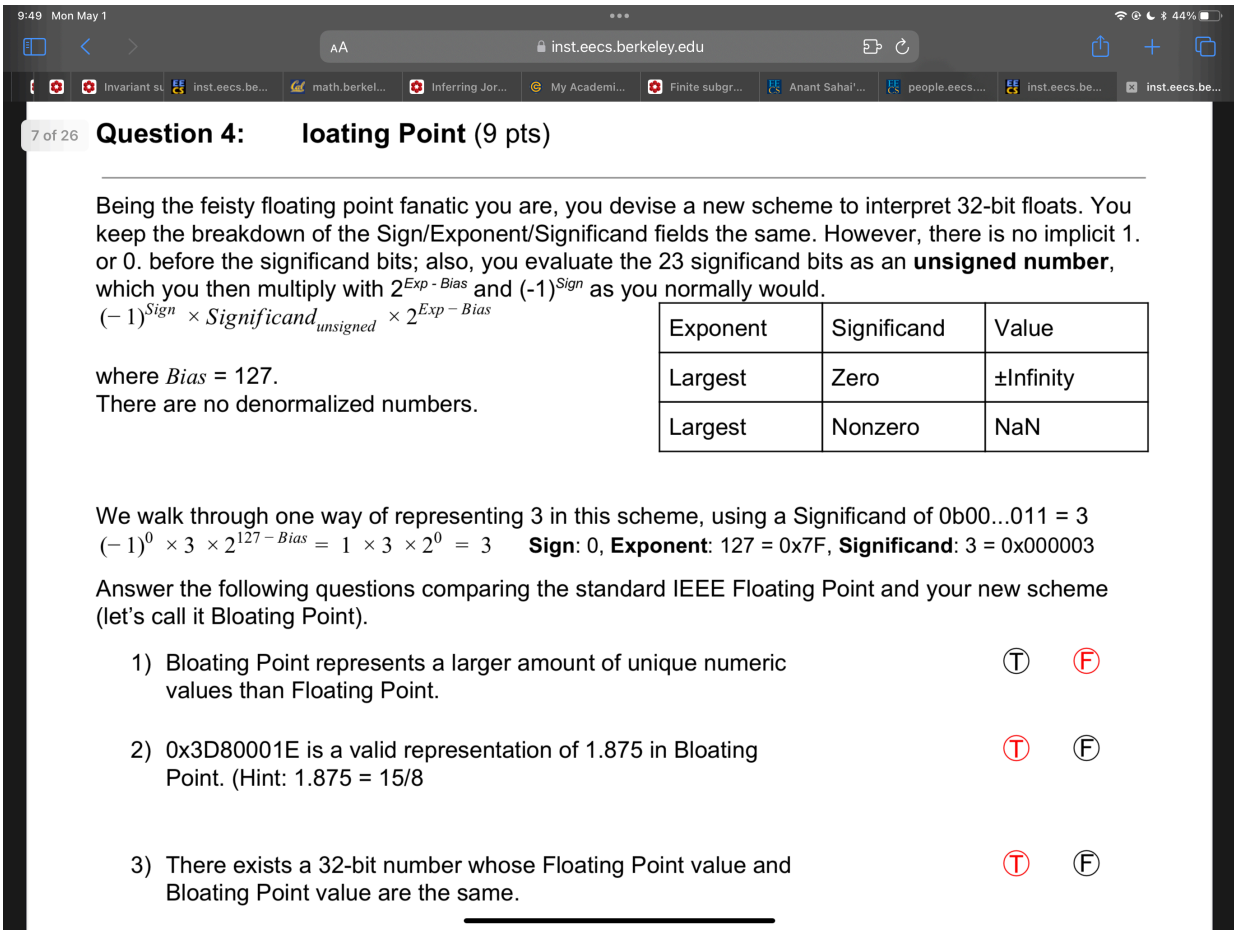
R Rosalie Fang ADMIN 2y #1699ee

#1798ae

♡ ···

Anonymous Monkey 2y #1699bf ✓ Resolved

SU18 Final Q4 2
Why is it ...1E instead of ...0F?



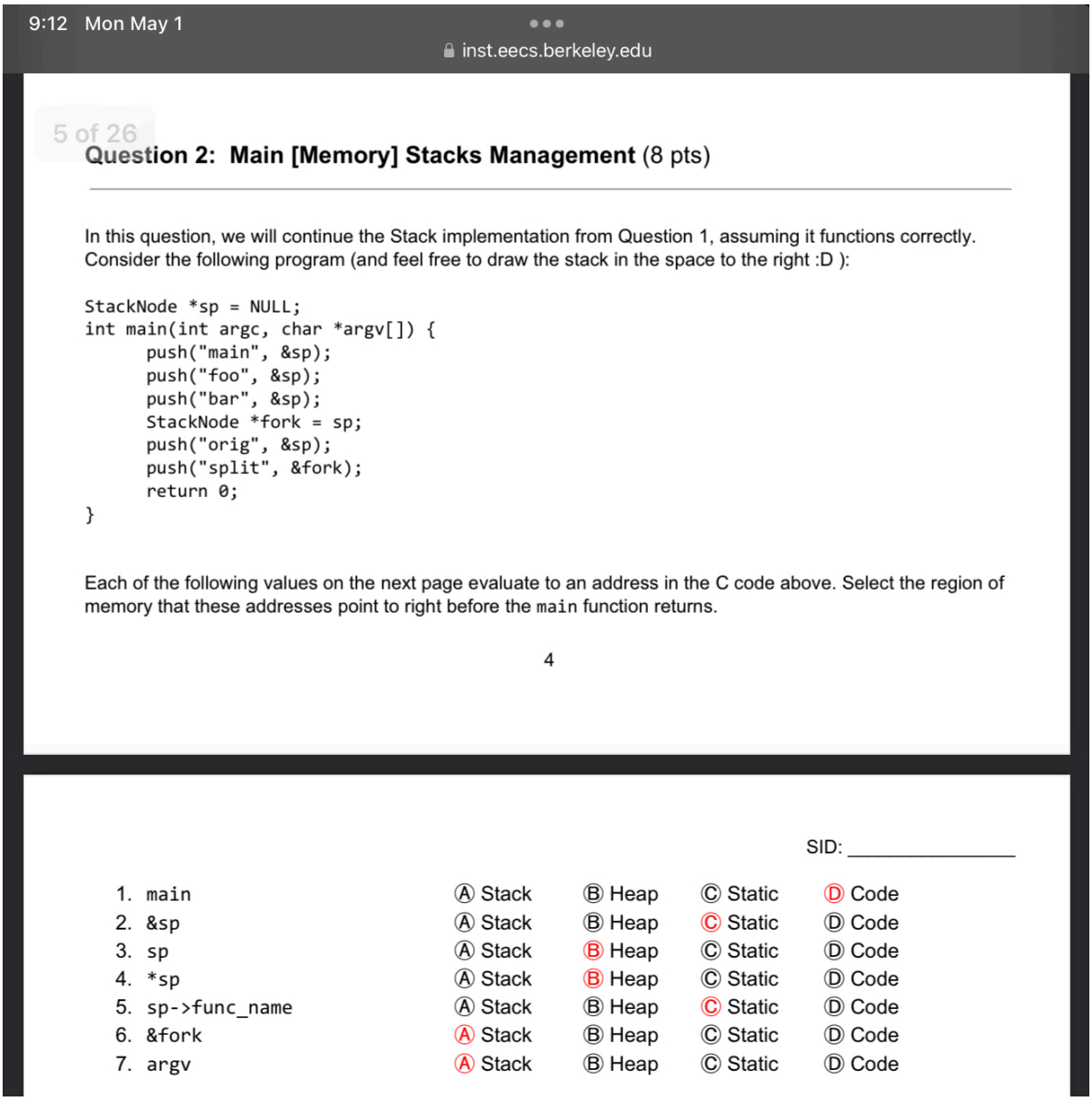
R Rosalie Fang ADMIN 2y #1699ed

We evaluate the 23 significand bits as an unsigned number. so we evaluate 0x00001E as an unsigned number which gives 16 + 14 = 30. Exponent is 0b 11 1101 1 = 1 + 2 + 8 + 16 + 32 + 64 = 123, and sign bit is 0, so we have `(-1)^0 * 30 * 2^(123-127) = 30 / 2^4 = 30 / 16 = 15/8.`

Anonymous Monkey 2y #1699be ✓ Resolved

SU18Final Q2 2

Why is the thing &sp points to stored in static?



Jero Wang ADMIN 2y #1699dd

It is declared outside of any functions, so it is part of static.

Anonymous Swallow 2y #1699add

Hello, why is sp and *sp on the heap? Thank you

Anonymous Tarsier 2y #1699bd ✓ Resolved

FA17 Q13 4D. Why do small random reads have higher throughput on RAID 5 than RAID 1? Doesn't reading from RAID 5 require accessing the parity information distributed across the disks?

Jero Wang ADMIN 2y #1699dc

RAID 5 has more disks than RAID 1, so the overall throughput is higher. Yes, it requires accessing parity information, but it can still read from more disks at once.

Anonymous Tarsier 2y #1699bc ✓ Resolved

FA15 F-2 c: Even with the data dependancy on resullt[i-1], why is this always correct while it was sometimes incorrect for part b? Is it because of `reduction`?

♡ ···

R Rosalie Fang ADMIN 2y #1699afc

It's not because of `reduction` but rather what you chose to parallelize. part b parallelized outer loop which could cause a race condition precisely because of data dependency, where part c only parallelized inner loop, so the outer loop is single threaded and got computed sequentially.

r

♡ ···

Anonymous Swallow 2y #1699bb ✓ Resolved

Is problem 4 in Sp19 Final in scope? Thank you!

Problem 4 ***Nick Goes Nuclear - Atomics*** **(17 points)**

In class you learned about OpenMP and got to experience speedups on the Hive Machine. However after your time in 61C you developed a deep-seated hatred for X86 and have determined that you want to employ OpenMP on RISC-V machines using atomic instructions.

You decide to start small and you seek to implement the following parallelization of summing a loop.

```
int sum = 0;
#pragma omp parallel for {
for (int i = 0; i < n; i++) {
    #pragma omp critical
    sum += A[i];
}
```

When executing the for loop, each thread holds its local starting and terminating byte offset in t0 and t1 respectively. You store the address of `sum` in s1 and the address of `A` in s2. Now you are tasked with implementing the actual `sum` update. You develop the following code which WORKS:

```
loop_start:
    beq t0 t1 end
    add t2 s2 t0
    lw t2 0(t2)
retry:
    lr.w t3 (s1) # Load sum and place our reservation
    add t3 t3 t2
    sc.w t4, t3 (s1)
    bne t4 x0 retry # Check if our store failed
    addi t0 t0 4
    j loop_start
```

(a) Your friend, however, took 61C back in Fall 2017, so he only understands amoswap. Your friend asks if you could reimplement the same piece of coding using amoswap instead, without needing any values other than those in t0, t1, s1, and s2. Is this possible? Why or why not?

♡ ···

R Rosalie Fang ADMIN 2y #1699ec

Not exactly. We didn't teach lr.w and sc.w and we did teach amoswap.ac and amoswap.rl...

♡ ···

Anonymous Swallow 2y #1699ba ✓ Resolved

Su18 MT2 Q2

Hi, would it be possible to explain how to solve the following problem? Thank you

2) Now we have a different machine with two caches, an L1 and an L2 cache. Both caches are direct mapped caches. The L1 cache can hold 256 B of data and the L2 cache can hold 4 KiB of data. Assume the following code is run on this machine:

```
#define ARR_SIZE 2048

uint16_t sum (uint16_t *arr) {
    total = 0;
    for (int i = 0; i < ARR_SIZE; i++) {
       total += arr[i];
    }
    return total;
}
```

This produces a hit rate (HR) of 7/8 for the L1 cache and 3/4 for the L2 cache. Given that `arr` is a block aligned address and `sizeof (uint16_t) == 2`:

A. What is the blocksize of the L1 cache **in bytes** that produces its hit rate?

♡ ···

R Rosalie Fang ADMIN 2y #1699eb

Every line of `total += arr[i]` is 1 memory access. Our access pattern is therefore `arr[0], arr[1], arr[2], ... arr[ARR_SIZE-1]` . In order to achieve a 7/8 hit rate for L1 cache, we know arr[0] must be a compulsory miss, but the next 7 sequential accesses need to be hits before we have another compulsory miss, therefore there are 8 elements per cache line, 4 bytes each.

♡ ···

Anonymous Llama 2y #1699ada

isnt each element in the cache line 2 byte. b/c it is an array of uint16_t, so would it be 16 bytes per cache line instead of 32?

♡ ···

Anonymous Raccoon 2y #1699af ✓ Resolved

Fall 2015 Final mt2-4 c

Is it possible for part I to have other types of misses?

Such that capacity miss and conflict miss

The information for one student in regards to clobbering a single midterm is captured in the data of the following *tightly-packed* struct:

```
typedef struct student {
   int studentID;
   float oldZScore;
   float newZScore;
   int clobber;          /* a value equal to 1 if a student clobbers,
                            0 if otherwise */
} student;
```

We run the following code on a 32-bit machine with a 4 KiB write-back cache. `importStudent()` returns a `struct student` that is in the course roster and that has not been returned by `importStudent()` previously. For simplicity, assume `importStudent()` does not affect the cache.

```
int ARR_SIZE = 512; //Class size rounded down for simplicity
student *61CStudents = (student *) malloc (sizeof(student) * ARR_SIZE);

/* Assume malloc returns a cache block aligned address */
for (int i = 0; i < ARR_SIZE; i++) {                        <=== part I
   61CStudents[i] = importStudent()  <- what does import student do?
}

for (int i = 0; i < ARR_SIZE; i++) {                        <=== part II
   if (61CStudents[i].oldZscore > 61CStudents[i].newZscore){
     61CStudents[i].clobber = 0;
   } else {
```


c. At the label part I, assume that 61CStudents is filled with the correct data. What type of misses will occur from memory accesses during the process? Why?

The misses that will occur from executing the for loop at label part I will be compulsory misses, because the loop will be accessing student structs that will be accessed for the first time.

♡ ···

R Rosalie Fang ADMIN 2y #1699ea

No, because part I is when we read in all the data for the first time so it's impossible to have any other types of misses.

♡ 1 ···

Anonymous Monkey 2y #1699ab ✓ Resolved

SP18 Final Q12 f, g, h: Where should we start when thinking about these questions?

Problem 12 ***[F-4] Virtual Memory*** **(20 points)**

Demand paging (storing part of a process' memory on disk) is yet another example of caching in computer systems. If we think of main memory as a cache for disk, what are the properties of this cache? Assume a machine with 64 bit addresses, 16KB pages, a 4-way fully associative TLB, and 8B words.

(a) Associativity?

- ○ Direct Mapped
- ● Fully Associative
- ○ N-Way Set Associative

(b) Block size:

Solution: 16KB

(c) Address layout. Your answer should be of the form [N:M] where N is the bit number of the most significant bit of the field and N is the bit number of the least significant bit of the field. For example, if the tag consists of the first 4 least-significant bits, you should write [3:0]. If the field is not applicable to paging, you may write "N/A".

Solution: Tag bits:[63:14] Index bits:N/A Offset bits:[13:0]

(d) Write policy?

- ○ Write Through
- ● Write Back

(e) Allocation policy?

- ● Write Allocate
- ○ Write No Allocate

TLB Reach. We have written a strange and mysterious summation function. It uses a mystery constant called `T`. You may assume that `T` is defined (but you don't know what to) and that `arr` will always have enough elements (the function will never access outside of `arr`). The function is run on a machine with the following properties:

- 64 bit addresses
- 4 byte words
- 4KiB pages
- 4GiB of main memory
- 1MiB fully-associative cache with 64 byte blocks
- 2 entry fully associative TLB

- 4 level page table with 8 byte entries
- The OS uses LRU when paging to disk

```
#define NITER 10*1024*1024
#define T ???       // see below

int MysterySum(int *arr) {
     int i = 0;
     int sum = 0;
     for(; i < NITER / 2; i++)
          int p = (i % T)*4096;
          int b = i % 4096;
          sum += arr[p + b];
     }

     /* Timer starts here*/
     for(; i < NITIR; i++) {
          int p = (i % T)*4096;
          int b = i % 4096;
          sum += arr[p + b];
     }
     /* Timer ends here */

     return sum;
}
```

(f) **Performance of T**
Rank the the following values of T based on how fast the second loop only executes (assuming the first loop has already ran). You should state whether pairs of values are $<$ or $=$. For example, you should write $1 < 2$ if T=1 causes the second loop to run strictly slower than T=2. Likewise, you could write 8=2 if 8 is about as fast as 2.
T = 1, 2, 3, 4

Solution: $3 = 4 < 1 = 2$

(g) **System Design**
What system parameter would you change in order to maximize system performance for T=27. You must mark only one of the following (pick the one with the largest performance gain):

○ Address Size	○ Cache Block Size
○ Page Size	● TLB Capacity
○ Word Size	○ TLB Associativity
○ Main Memory Size	○ Page Table Depth
○ Cache Capacity	○ Page Table Entry Size

(h) **Page Table Walk**

Given the list of virtual addresses, find the corresponding physical addresses. For each address, you must also note whether the access was a TLB hit, Page Table hit, or Page Fault (by writing yes/no for each). If the access is a page fault, you should leave the PPN and PA fields blank. Do not add this entry to the TLB.
Our virtual memory space has 16-byte pages and maintains a fully-associative, two-entry TLB with LRU replacement. The page table system is hierarchical and has two levels. The two most-significant bits of the VPN index the L1 table, and the two least-significant bits of the VPN index the L2 table.

Solution:

Virtual Address	Virtual Page Number	Physical Page Number	Physical Address	TLB Hit, Page Table Hit, Page Fault?
0x10	0x1 = 0b00 01	0x12	0x120	Page Table Hit
0x5C	0x5 = 0b01 01			Page Fault
0x39	0x3 = 0b00 11	0x5C	0x5C9	Page Table Hit
0x1F	0x1 = 0b00 01	0x12	0x12F	TLB Hit

TLB:

VPN	PPN
0x1 → 0x2	0x12 → 0x9
0x3 → 0x1	0x5C → 0x12

♡ ···

R Rosalie Fang ADMIN 2y #1699df

(f). whenever I think about performance analysis for cache questions, I like write it out, so in this case specifically which values are accessed, for example what would arr[p + b] access look like for T = 1, 2, 3, 4.

(g). in this case I think the question to ask is, what is causing the most number of misses with this access pattern, you might have to actually write out the TLB, page table, etc. to see

(h). this is just a regular page table walk, but it has a L2 page table which you don't need to worry about!! make sure to understand HW9.2 and 9.3 for VM questions.

♡ ···

Anonymous Monkey 2y #1699aa ✓ Resolved

SP18 Final Q8 an and b: Why can we add the time together? Shouldn't the minimum clock period be 350ps and we need 5*350 and 4*350 ps respectively?

Problem 8 ***[M2-4] Datapath*** **(10 points)**

Recall the standard 5-stage, single cycle datapath contains stages for Instruction Fetch, Decode, Execute (ALU), Memory, and Write-back. Datapath designers are interested in reducing the phases necessary for execution such that instead of accessing both the Execute (ALU) phase and the Memory phase, instructions access either one or the other, but not both. This would create a 4-stage, single cycle datapath with the following stages: Instruction Fetch, Decode, Execute OR Memory, and Write-back.

Instr Fetch	Instr Decode	Execute (ALU)	Memory	Write-back
100ps	150ps	200ps	350ps	150ps

(a) Given the table above and the described datapath above, what is the time it takes for a single instruction that utilizes all stages to execute on the typical 5-stage, single cycle datapath?

Solution: $100 + 150 + 200 + 350 + 150 = 950\text{ps}$

(b) What is the time it takes for a single instruction that utilizes all stages to execute on the new 4-stage, single cycle datapath?

Solution: $100 + 150 + MAX(200, 350) + 150 = 750\text{ps}$

♡ ...

Jero Wang ADMIN 2y #1699db

Since it is a single cycle datapath, it isn't pipelined. The stages are there for the sake of timing calculations and do not refer to pipeline stages.

♡ ...

Anonymous Monkey 2y #1699e ✓ Resolved

SP18 Final Q6 part a

Problem 6 *[M2-2] Read and Write* **(15 points)**

Recall in class we learned that we can optimize our CPU pipeline by having register writes then reads within the same cycle. Let's call this implementation **write-read**.

Consider a new implementation where register reads happen before register writes within the same cycle. Let's call this implementation **read-write**.

Now consider the following RISC-V code and answer the following questions about a 5-stage RISC-V pipeline. **Assume no forwarding and no branch prediction**.

You are given that there needs to be at least one stall after line 4 for both implementations.

	loop:
1	slli t0 a1 2
2	or t2 a1 t1
3	add t0 t0 a0
4	lw t1 4(t0)
5	beq t1 x0 loop
6	addi t2 t2 5
7	sw t2 8(t0)
8	add a0 t2 x0

(a) Consider the code above and the **write-read** implementation. Which lines should be followed by a stall to guarantee correctness? (You are given that there needs to be at least one stall after line 4). For example, if an instruction on line A causes an instruction on line B to stall, bubble A.

- ● 1
- ○ 2
- ● 3
- ● 4
- ● 5
- ○ 6
- ○ 7
- ○ 8

Solution: line 1, 3, 4 (given), 5

Why doesn't line 6 cause a stall since in line 7 we need the calculated t2 value yet there is no forwarding.

♡ ...

Anonymous Monkey 2y #1699f

(g) If we decide to reorder instructions, which instruction is the best choice to replace a nop after line 4? Choose the line number of that instruction.

- ○ 1
- ○ 2
- ○ 3
- ○ 4
- ○ 5
- ○ 6
- ○ 7
- ● None - no reordering needed

Same question part g

Why can't there be any optimization through reordering?

♡ ...

R Rosalie Fang ADMIN 2y #1699afb

line 6 should cause a stall I think...

We can't reorder because there's no line that we can move between line 4 and 5 since line 5 is a branch instruction, pulling line 6, 7, or 8 up would mean that the register values might be incorrect in the case that we do take the branch.

♡ ...

Anonymous Monkey 2y #1699d ✓ Resolved

SP18 Final Q4 part a

Why is the first red line .Data not .String? Thanks

Problem 4 ***[MT1-4] CALL*** **(14 points)**

Consider the following C code and assembly code:

```
#include <stdio.h>

int main() {
     int i, sum = 0;
     for (i = 100; i !=0; i--)
          sum = sum + i * i;
     printf ("The sum of sq from 100 ..  1 is %d\n", sum);
}
```

Address	Assembly
	.data
	str:
0x80	.string "The sum of sq from 100 .. 1 is %d\n"
	.text
	main:
0x00	addi sp, sp, -4
0x04	sw ra, 0(sp)
0x08	mv a1, x0
0x0c	li t1, 100
0x10	j check
	loop:
0x14	mul t2, t1, t1
0x18	add a1, a1, t2
0x1c	addi t1, t1, -1
	check:
0x20	bnez t1 loop
0x24	la a0, str
0x28	jal printf
0x2c	mv a0, x0
0x30	lw ra, 0(sp)
0x34	addi sp, sp 4
0x38	ret

Figure 1: Assembly Code with Address

(a) Please fill in all lines in the above assembly code.

♡ ...

E Erik Yang TA 2y #1699ae

here, we're defining str to be a constant that is equivalent to "The sum ..." , which is done in a .data section

♡ ...

Anonymous Llama 2y #1699acf

What is the line la a0, str do and why do we have to reset a0 to 0 after we print?

♡ ···

E Erik Yang TA 2y #1699adc

Replying to Anonymous Llama

La takes in a label and stores the address of the label into a0. This allows printf to take in a0 as an argument and print str which is from the top. We put x0 into a0 immediately after because I'm assuming we want a0 as a return value so we put 0 inside it. The question doesn't want you to write anything but that's what I'm assuming.

♡ ···

Anonymous Lark 2y #1699c ✓ Resolved

Sp07-MT-Q3 Part B (https://tbp.berkeley.edu/exams/156/download/)

1. In the line under `strcpy(new_significand, b->significand);` why is it `free(b->significand)` ? I am wondering why the solution did not free `new_significand` .
2. Also is `b->significand[min_sigfigs] = '\0';` valid since we assume that it is on the heap (malloc'd) already since the function is passing in a pointer to a vector that contains a pointer to the elements? Usually indexing into a char * pointer is not valid (such as `char *p = "hello"` and then doing `p[3] = '\0'`) since it is read-only (not on heap)?

```
/* truncate all elts to have min_sigfigs */
for (unsigned int i = 0; i < vec->num_elts ; i++) {
                          vec->elts + i;
    sci_bignum_t *b = _____________________________ // convenient reference
             b->num_sigfigs                  min_sigfigs
    if (_____________________________ > _____________________________) {
                               (char *) malloc((min_sigfigs + 1)*sizeof(char))
        char *new_significand = (______) malloc(_____________________________);
        b->significand[min_sigfigs] = '\0';
        ____________________________________________________________ // for strcpy

        strcpy(new_significand, b->significand);
        free(b->significand);
        ____________________________________________________________________
        b->significand = new_significand;
        ____________________________________________________________________
        b->num_sigfigs = min_sigfigs;
        ____________________________________________________________________
```

3. Below is my solution for the lines under `strcpy(new_significand, b->significand);` . I did not see the need to free `b->significand` as it was not malloc'd in the function?

```
b->significand = new_significand;
b->num_sigfigs = min_sigfigs;
free(new_significand);
```

♡ ···

E Erik Yang TA 2y #1699ac

1. Since we're changing what b->significand is, we have to free what was in there before. The old value was from the parameter, vec, so that's why we have to free that pointer.

2. That is valid, because we're really mallocing space for an array on the heap, and adding a null terminator at the end. This is possible, as we've seen in project 1.

♡ ···

Anonymous Swallow 2y #1699a ✓ Resolved

Fa18 MT Q5e

How do we get 21 from this code? Thank you

e) The value of memory pointed to by x1 is 10. Two cores run the following code concurrently:

`lw x2,0(x1)`
`addi x2,x2,1`
`sw x2,0(x1)`
	`lw x3,0(x1)`
	`add x3,x3,x3`
	`sw x3,0(x1)`

...what are possible values of the memory afterward? (select ALL that apply)

☐10 ■11 ☐12 ☐13 ☐14 ☐15 ☐16 ☐17 ☐18 ☐19 ■20 ■21 ■22

♡ ···

Y Yile Hu TUTOR 2y #1699b

consider the following execution sequence:

```
lw x3, 0(x1)        0(x1) == 10
add x3, x3, x3
sw x3, 0(x1)        0(x1) == 20
lw x2, 0(x1)
addi x2, x2,1
sw x2, 0(x1)        0(x1) == 21
```

♡ 1 ···