2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering Education for 130 Years
Paper ID #38449

Teaching Computer Architecture with Spatial Ability Considerations

Dr. Geoffrey L. Herman, University of Illinois at Urbana - Champaign

Dr. Geoffrey L. Herman is the Severns Teaching Associate Professor with the Department of Computer Science at the University of Illinois at Urbana-Champaign.

Mr. Muahmmad Suleman Mahmood

Suleman Mahmood is a Ph.D. student in the Department of Computer Science at the University of Illinois, Urbana Champaign. Suleman completed his MS in Computer Science from Lahore University of Management Sciences and BS in Electrical Engineering from University of Engineering and Technology. He is interested in exploring how students learn computer science concepts and developing tools to assist them in the learning process

Teaching Computer Architecture with Spatial Ability Considerations

1 Introduction

Students' spatial ability or ability to reason about visual images is highly correlated with success and retention in Science, Technology, Engineering, and Mathematics (STEM) fields. Wai et al. [1] found that this correlation is particularly strong for computer science and engineering disciplines[1]. Many studies indicate that female students have lower spatial ability than male students and wealthy students have higher spatial ability than students from poorer backgrounds [2, 3, 4, 5] . The causes of gender and socioeconomic differences in spatial ability are not known but changing how we teach computer science to reduce our reliance on spatial ability may be a pathway to diversifying computer science. Instructors in other STEM disciplines have promoted the success of all students, regardless of spatial ability, by changing their instructional materials [6, 7, 8]. The first step in this process is to identify the instruction materials and concepts that heavily rely on spatial ability.

Our computer architecture course has historically had one of the largest gender-based disparities in student grades in the department and the largest for all courses that students typically take in the first two years of the CS curriculum [9]. These disparities have more or less persisted despite efforts to change the course using a variety of evidence-based pedagogies [10]. While these studies highlight differences based on gender, we hypothesized that there may be achievement differences based on spatial ability may also exist because of the correlation between gender and minority status, and spatial ability [2]. If a difference based on spatial ability is observed, then strategically changing instructional materials to close the gap between high and low spatial ability students can indirectly level the playing field for women and minorities.

We calculated the correlation between students' spatial ability scores and their scores in different topics in the course to discover that the correlation for the topic of caches was stronger than all the other topics with a Pearson's correlation coefficient [11] of 0.33. The topic of number representations had the second highest correlation coefficient with 0.29. Both correlations were statistically significant with $p < 0.001$. In both topics, the spatial arrangement of bits can be used to explain these topics and may be the underlying reason for the significant dependence of students' performance on spatial ability. For example, Figure 1 shows one of the slides from our course material for how addresses from a 16-byte memory map to a small cache. In Figure 1, the spatial orientation of the least-significant bit on the right of the memory address (highlighted in yellow) indicates that this bit corresponds to the Block Offset of the cache. Likewise, the middle address bits, colored in the diagram, indicate the index bits for the cache, and the left-most

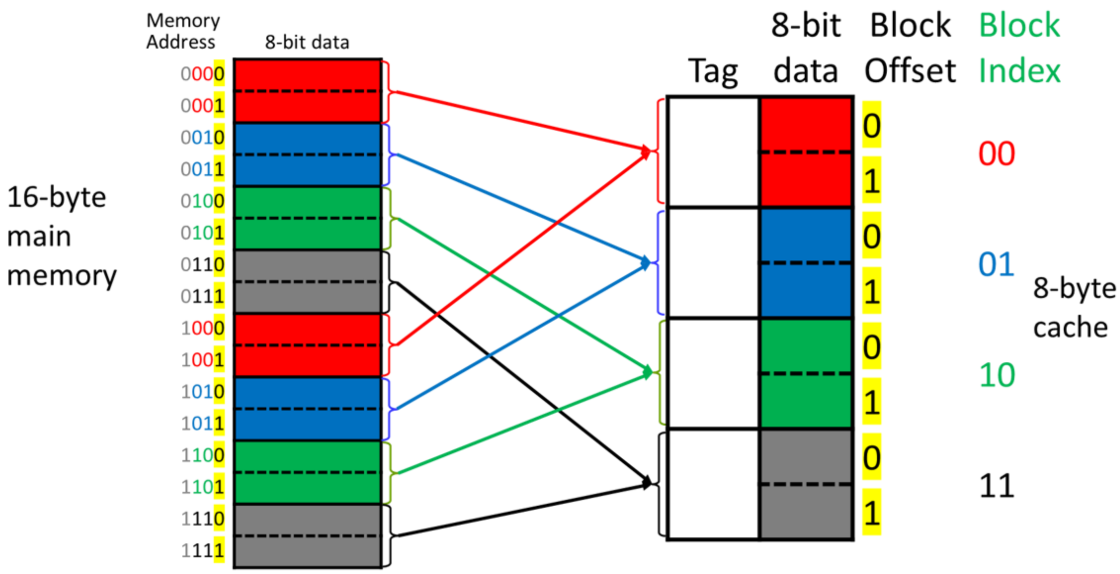


Figure 1: Slide showing how to map addresses to cache. Different parts of the addresses are highlighted and colored to show index and offset.

address bits indicate the tag. This process requires students to decode spatial information and mentally translate visual information across the diagram (e.g., the arrows), likely requiring high spatial ability.

Thus, we decided to redesign our instructional materials for caches and number representations to reduce the need to use spatial reasoning to understand the diagram, a technique that others have found effective [6, 7, 8]. In particular, the addition of more algorithmic approaches to highly spatial topics was found to be most effective [8], so our efforts focused on changing our teaching resources to emphasize algorithmic approaches rather than spatial approaches for explaining number representations and cache behaviors.

An analogy between hash tables and caches offers one such algorithmic approach for teaching caches: Addresses can be converted into the tag, index, and offset of a cache using a mathematical hashing function (modular arithmetic and division) rather than the spatial arrangement of bits. This analogy and mapping function are discussed in popular textbooks [12, 13] and was an approach we confirmed that other instructors used through a survey we conducted. Modular arithmetic is a technique that can also be helpful for understanding overflow issues for the topic of number representations.Therefore, we focused on teaching the use of modular arithmetic with the topic of number representations and then used the same concept to define the hashing function to map data addresses to caches.

We hypothesized that this algorithmic approach would 1) have no detrimental effects for students' learning of number representations and caches in general and 2) help students with low spatial ability better learn number representations and caches. In particular, we expected to see this benefit to be most prominent on questions where students need to map addresses to the tag, index, and offset of the cache, which we believe is the task that requires the most spatial ability.

Checking the effect of this technique on students' understanding of number representations provides a synergistic robustness check. To test these hypotheses, we ask the following research questions.

1. Does teaching a more algorithmic approach to number representations and caches, as opposed to a spatial approach, lead to a significant difference in students' performance on quiz questions on these topics?
2. If there are changes in student performance on these assessments, is it affected primarily by students' spatial ability?

2 Background

Lohman [14] defines spatial ability as the ability to generate, retain, retrieve, and transform well-structured visual images. As indicated by Lohman's definition [14], spatial ability represents multiple skills. Therefore, a comprehensive measurement of spatial ability must use multiple types of tasks. The most popular test for spatial ability is the Purdue Spatial Visualization Test [15]. This test is designed to test the ability to visualize rotated shapes. The skill of rotating shapes is important in many STEM fields [8, 16, 17]. This test shows large individual variance and gender differences [2]. Another common test is the Educational Testing Service (ETS) Hidden Figures Test [18]. This test is designed to test the ability to find simpler shapes inside more complex shapes. Performance on this test is also correlated with retention in STEM fields [19]. Domain specific spatial ability tests can also be developed to meet the needs of a particular field. Ormand et. al. developed Geologic Block Cross-sectioning Test, which tests the students' ability to recognize the correct vertical cross-section through a geologic block diagram. [16]. In this work, we did not develop any domain specific tests for spatial ability. We used both Purdue Spatial Visualization Test [15] and Educational Testing Service (ETS) Hidden Figures Test [18] to measure spatial ability for our students. We used these two tests because manipulating caches spatially can require mentally translating and rotating blocks of data from memory to the cache and parsing addresses requires identifying smaller parts of an address from a larger address.

There have been multiple attempts at reducing the spatial ability gap between genders and how it affects participation in STEM. Sorby and Baartmans [20] developed a course for enhancing the 3-D spatial visualization aimed at first year engineering students and reported lower dropout rates for student who completed their program. Sorby also developed a spatial training intervention aimed at middle school students [20] and reported better performance in introductory programming for students taking part in the training program. These studies show that a students' spatial ability is malleable and spatial skills can be learned [21]. However, additional classes that only a subset of student is required to attend may be perceived as remedial. This perception can discourage students and push them away from STEM fields [22].

Another method for reducing the effect of spatial ability in STEM education is to augment the representations used in STEM education with other techniques that do not rely as much on spatial skills. Stull [6] and Stieff[23] used concrete models of molecules to help students reason about the structure of molecules. Hegarty [7] and Stieff[8] reported that introducing algorithmic approaches for reasoning about molecular structure can be an effective way to reduce gender gap in test scores related to molecular structure. In particular, Stieff[8] showed that teaching a combination of algorithmic and metal rotation strategies was the most effective in helping low spatial ability students. The advantage of local changes to course materials, that Stull, Stieff and Hegarty used, is that the students do not have to take any additional courses or training, reducing perceptions that the intervention is remedial and the implication that students with low spatial ability need to be fixed.

There has not been much work studying how students learn binary number representations or how to help students learn them. Herman et al. documented students' misconceptions about binary number systems and found that students struggle to understand the role that different bit positions play in interpreting positional notations, frequently using improper weights [24]. Likewise, students misunderstand how to detect overflow in fixed-width binary representations, focusing inappropriately on the physical position of carry out bits and not on what those bits mean [24]. While they don't consider spatial reasoning in their analysis, it may be possible that students with low spatial ability struggle with number representations because the spatial arrangement of bits is so vital to the interpretation of binary numbers.

The topic of caches is commonly taught with the help of many visual aids. For example, periodic mapping of different memory addresses to caches is described with the help of coloring and highlighting in diagrams in our instruction material as well as popular textbooks [12, 13]. We also use analogies comparing how 2-D matrices are stored in memory to describe caches with multi-byte cache blocks. Chunks of a linear memory are separated and rotated to fit a 2-D picture of a cache. The reliance on visual aids has been shown to be taxing for students with low spatial ability [25, 16]. This offers an explanation why students with low spatial ability may find it harder to understand caches. We used data from Fall 2019 semester and divided the students into two equal sized groups based on their spatial ability scores. Students with low spatial ability had an average of 65% on the cache quiz and students with high spatial ability had an average of 76% on the cache quiz. The difference between the groups is statistically significant ($p < 0.001$) with a moderate effect size (Cohen's $d = 0.52$) [26], representing a full letter grade difference between students of high and low spatial ability for the topic of caches. In this paper, we seek to translate the research findings of Stieff et al. [8] from the domain of organic chemistry to computer science. As far as we know, this would the first attempt to replicate their efforts in computer science. Examining the generalizability of their findings across disciplines may help us determine whether the strategy of emphasizing algorithmic approaches is potentially an effective pedagogical approach that could be applied across many different domains to mitigate the effects of differing spatial ability across student populations.

3 Methods

3.1 Participants

In this study we are using a quasi-experimental design with Fall 2021 students going through the computer architecture course with original, spatially focused content and Spring 2021 students using the updated course with the algorithmic approach. The course is primarily taken by second-year undergraduate students and enrolls between 300 to 400 students every semester. The course uses a flipped classroom model, where students are required to watch recorded video lectures before class, and solving problems in groups during class, and then complete a longer lab

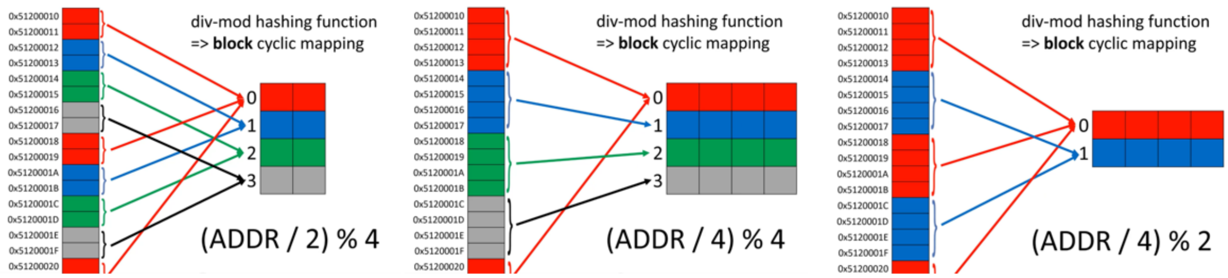


Figure 2: Three examples from videos showing how the hash function changes with cache configuration

exercise after class based on the in-class material. The course grade is based on homework, lab exercises, 11 proctored quizzes, and one comprehensive final exam. Two quizzes include questions on number representations and arithmetic and two quizzes focus on caches. The remaining seven quizzes cover topics that are not related to the instructional material modified in this study. The aggregated scores of these seven quizzes was used to establish that students in both semesters were otherwise similar. There were 45 students who registered for the course in Fall 2021 and were also registered in Spring 2022. These students could have been influenced by both versions of the course threatening the validity of the results. Therefore, we decided to limit the study to students who registered for the course for the first time in each semester. This gave us a final sample size of 371 students for Fall 2021 semester and 340 students for Spring 2022 semester. The same instructor taught both versions of the course.

3.2 Timeline

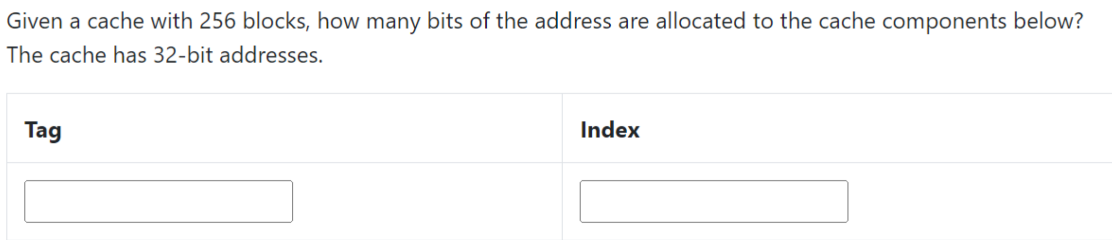
We followed the same timeline for both semesters and changes to the instruction material for number systems and caches were the only changes between semesters. At the beginning of the semester, students were asked to take a spatial ability assessment based on Purdue Spatial Visualization Test [15] and Educational Testing Service (ETS) Hidden Figures Test [18] to measure spatial ability for our students. This was an optional assessment and extra credit equal to 0.5% of the course grade was offered to students to take the test. In the first week of the course, students learned about number systems and binary arithmetic. They took the quiz for number systems in week 2 of the semester. The topic of caches was taught in weeks 11, 12 and 13 in both semesters. Students took the first cache quiz in week 14 and the second one in week 15 of the semester.

3.3 Content Changes

3.3.1 Video Lectures

In prior video lectures on binary arithmetic, overflow for unsigned binary addition was explained by showing how the carry-out from the left-most bit position (spatial explanation) could not be stored, creating mathematical errors. The video lecture was re-recorded to emphasize that computers perform modular arithmetic due to fixed width representations and overflow was explained to occur when traditional addition did not match modular addition.

Figure 3: This slide shows the parts of the binary address extracted by the functions.


Given a cache with 256 blocks, how many bits of the address are allocated to the cache components below? The cache has 32-bit addresses.

Tag	Index

Figure 4: Old question asking students to calculate sizes of tag and index fields before parsing addresses.

In prior video lectures on mapping memory addresses to the tag, index, and offset of a cache, we taught students a spatially focused approach to first calculate the number of bits needed for tag, index, and offset based on the cache configuration, and then parse the address into tag, index, and offset based on bit positions. We changed the video lecture to teach students how to derive a set of hashing functions based on the cache configuration, and then how to apply those hashing functions to derive the tag, index, and offset as decimal or hexadecimal numbers. Our mapping function is presented slightly differently from Hennessy and Patterson's textbooks [12]. We include division within the hashing functions whereas the textbooks show calculating a block index as a separate step. Figure 2 shows some screenshots from the video that show how the mapping function changes with cache block size and number of sets. One slide was used to explain how the hashing function parsed the binary bits (See Figure 3) into tag, index, and offset. Similar changes were also required for videos that focused on temporal locality and set-associative caches.

3.3.2 Group work and Homework

Prior to the experiment, homework questions related to overflow showed students n-bit binary numbers and asked students to add them in binary and determine whether overflow occurred by checking the value of certain bits in certain positions (spatial approach). We changed these homework problems so that they showed students decimal numbers and asked students to add them in binary and determine whether overflow occurred by using the modulus operation (algorithmic approach). In-class group exercises were likewise changed.

Cache questions on homework and in-class group exercises were changed in the same way as the

Consider a direct-mapped cache that has 8 blocks and 4 bytes per cache block

Determine the missing values in the equations below that you need to calculate the tag, index, and offset of an address for this cache configuration

Offset = Address % [integer]

Index = Address / [integer] % [integer]

Tag = Address / [integer]

Figure 5: New questions asking students to calculate the formula to map addresses to caches

Given a direct-mapped cache with **8 sets** of **4-byte blocks** and a main memory that has **16-bit addresses**.

A load instruction is trying to load from address `0xe0c7`. If there is a cache hit, determine what data would be retrieved from the cache (be sure to add the 0x prefix). If there is a cache miss due to a tag mismatch, type "mismatch". If there is a cache miss due to an invalid cache block, type "invalid". If a miss is caused by both an invalid cache block and a tag mismatch, type "invalid".

Index	**Valid**	**Tag**	**Block offsets**			
			0	1	2	3
0	1	0x21a	0x35	0x63	0x6c	0xfe
1	1	0x706	0x6e	0xbf	0xf4	0xa9
2	1	0x780	0xe9	0x44	0x11	0xed
3	0	0x29c	0x38	0x24	0x58	0xeb
4	0	0x62f	0xc6	0x95	0xb3	0x35
5	1	0x06b	0x62	0xce	0x34	0x0a
6	0	0x784	0x3a	0x87	0xe0	0xcf
7	0	0x36e	0x7f	0x69	0x6a	0x78

0

HEX

DEC

1	2	3	CE
4	5	6	+
7	8	9	-
A	B	C	÷
D	E	F	×
COPY	0	=	%

Enter all hex digits in your answer. Enter 0x before the number, except for misses (enter "invalid" or "mismatch" without the quotation marks).

Answer = 0x

Figure 6: This question asks students to read from the cache when a specific address is accessed. The students are required to calculate tag and index fields for the address and compare them with the given cache state. This comparison is easier if students calculate the tag and index field values directly using the formulas introduced earlier. A calculator is provided to assist in the calculation

video lectures. Prior to the experiment, cache homework questions focused on asking students to calculate the number of bits needed to encode the tag, index, and offset of a cache (shown in Figure 4) and other questions asked students to parse binary addresses based on bit positions once the number of bits for tag, index, and offset were known to determine whether memory accesses resulted in hits or misses (spatial approach). These questions were changed to emphasize deriving the hashing functions (shown in Figure 5) and then other questions required students to apply those hashing functions to calculate the tag, index, and offset values (algorithmic approach). Figure 6 shows the new question. Note that students are provided with a calculator and the tag field in the cache is specified in hexadecimal with is easier to compare if students calculate tag for the address using the hashing functions. Explanations and solutions for these questions were changed to reinforce the new algorithmic approach.

3.4 Measures

There are three measurements for this study. Purdue Spatial Visualization Test [15] and ETS Hidden Figures Test [18] to measure spatial ability for our students. The score from both tests is aggregated to create a single measure for spatial ability. These tests are administered through an optional assessment so the sample size for which we have spatial ability data is smaller than the number of students enrolled in the course in each semester.

The second measurement is from quiz questions on the topic of number representations and arithmetic. For this measurement we selected six questions from the first two proctored quizzes that focus on students being able to calculate the results of an arithmetic operation on a computer and flag if an overflow has occurred. These questions were developed before Fall 2021 semester so they could be used in both semesters for a consistent measurement. We aggregated the scores on these questions to create a measure for students' performance on number representations and arithmetic.

The third measurement is for the topic of caches. We considered all question related to caches including the comprehensive cache performance analysis questions which require students to calculate the number of hits and misses for a given code using large data structures. This meant that all the questions from quiz 10 and three questions from quiz 11 were aggregated to create a measure for students' understanding of caches. One of the quiz questions is discussed in detail in Appendix A.

A major underlying assumption in this quasi-experimental study is that student samples from each semester belongs to the same population and the content changes are the only major factor that could influence the scores. Since the content changes are only supposed to affect four quizzes (two for number systems and two for caches), the scores for other topics should not be affected. We calculated average score for each student across the remaining seven quizzes to create a baseline measure for each students' ability level. Averaging the quizzes for each student was a reasonable choice because the quizzes carry equal weight for grading purposes in the course. The average quiz scores for the students were used to verify the underlying assumption that students come from the same population.

4 Results

As a quasi-experimental study, we first sought to verify that the student samples from both semesters could reasonably be considered as belonging to the same population. The mean average quiz score for Fall 2021 semester was 75.8% with a standard deviation of 33.4%. The mean average quiz score of Spring 2022 was 79.5 with a standard deviation of 31.2. Levene's test for variance indicated that both samples' variances are not significantly different with a p-value of 0.17. We conducted an independent samples t-test for average quiz scores with equal variance assumption to compare the populations. The test indicated that the null hypothesis that population means are different cannot be rejected at 95% significance level (p = 0.13). This means that it is reasonable to assume that both samples belong to the same population for the purpose of this study.

4.1 RQ1: Overall Student Performance

To evaluate the effect of changes in the instructional material on students' overall performance on quiz questions, we compared students' mean score for the topics of number systems and caches for the two semesters. We used independent samples t-test with equal variance (See Table 1 for details). The instructional changes did not result in a significant difference ($p = 0.674$) in student scores on the number systems quiz questions. The instructional changes did result in a significant difference ($p = 0.028$) in student scores on the cache quiz questions with a small effect size (0.19), corresponding to approximately a quarter of a letter grade.

4.2 RQ2: Spatial ability and instruction changes

For this question we want to find out if the instructional changes affected students with low and high spatial ability differently. Ideally, we would want to see a bigger improvement for students for low spatial ability as compared to students with high spatial ability because the changes are geared towards removing reliance on spatial ability. We defined our high and low spatial ability groups for Fall 2021 semester based on the median score. For Fall 2021 the median score was 89 (out of 100) and each group had 122 students. For Spring 2022, we used the median from Fall 2021 to divide the students into high and low spatial ability groups so that students with similar spatial ability are compared with each other. Spring 2022 high spatial ability group had 180 students and low spatial ability group has 106 students.

Measure	Fall 2021		Spring 2022		p-value
	N	mean(sd)	N	mean(sd)	
Number systems	342	81.7(18.0)	305	82.3(19.6)	0.674
Caches	299	80.3(13.1)	249	82.8(13.1)	0.028

Table 1: RQ1: Results from t-tests comparing overall student performance across the two semester: Fall 2021 (spatially focused) and Spring 2021 (algorithmically focused). There is no significant difference in student performance on number systems quiz questions. There is a small ($d = 0.19$) but significant improvement in student performance on cache quiz questions in the algorithmically focused group.

Measure	Spatial Ability	Instruction Method	Interaction
Number systems	< 0.001	0.793	0.620
Caches	< 0.001	0.406	0.500

Table 2: RQ2: Results from two-way ANOVA tests where spatial ability (high vs. low) and instructional method (spatial vs. algorithmic) are used as two independent variables and scores on quiz questions are the dependent variables. p-values indicate that differences in spatial ability correspond to differences in quiz performance, but that changes in instruction and the interaction of instructional change and spatial ability did not yield a significant difference in performance. The non-significant interaction term means that students with high and low spatial ability were affected similarly by the instructional change.

Student's spatial ability (high/low) and whether they took the course with or without the instructional changes, provides two independent variables. The dependent variables are the scores on number systems and cache quiz questions. We performed a two-way ANOVA to analyze the effect of spatial ability and instruction changes on number systems score and cache score separately. Table 2 summarizes the results of the two tests. For both cases, there was no statistically significant interaction between the effects of spatial ability and instruction changes. Simple main effects analysis showed that spatial ability had a statistically significant effect on the scores ($p < 0.01$ for both cases). The effect of instruction changes was not statistically significant for both number systems and caches. Note that this result is slightly different than the prior t-test because many students did not take the optional spatial ability quiz.

Results from the ANOVA (Analysis of Variance) confirm that spatial ability is an important determinant for performance in both topics, but our shift to an algorithmic instructional approach did not affect the students' performance. The non-significant interaction term shows that both low and high spatial ability were affected similarly by the instruction changes and students with low spatial ability did not benefit more than the students with high spatial ability.

4.3 Additional Analysis

The analysis for overall cache scores showed a small, statistically significant improvement with the algorithmic approach. We performed a follow-up analysis looking at how students performed on individual questions on the cache quizzes to understand the effect of changes in instruction and whether there was any particular value to the algorithmic approach or the spatial approach. We conducted a two-way ANOVA for each question (See Table 3). Two questions (Q4 and Q6) that involved 32-bit hexadecimal addresses showed statistically significant improvement at 95% significance level. One question (Q5) showed a significant decrease. None of these questions showed a deep dependence on spatial ability. Two questions (Q2 and Q3) showed significant dependence on spatial ability but were not affected by the instruction changes. None of the questions had a significant interaction term.

Question	Fall 2021 mean (sd)	Spring 2022 mean (sd)	Spatial Ability	Instruction Changes	Interaction
Q1: Read data from a given cache state. 8-bit hexadecimal addresses.	95.7 (16.0)	95.6 (14.5)	0.615	0.775	0.786
Q2: Determine number of hits/misses for code with simple memory access patterns.	70.7 (36.9)	68.4 (37.8)	< 0.001	0.254	0.495
Q3: Determine number of hits/misses for code with complex memory access patterns.	65.3 (19.7)	69.3 (19.6)	< 0.001	0.178	0.546
Q4: Determine number of hits/misses for a given address sequence. 32-bit hexadecimal addresses.	91.8 (23.4)	97.9 (4.25)	0.866	< 0.001	0.684
Q5: Update state of cache after multiple memory accesses. 8-bit hexadecimal addresses.	96.7 (10.2)	94.1 (14.5)	0.506	0.002	0.760
Q6: Find the address sequence with given code. 32-bit hexadecimal addresses.	77.1 (38.6)	87.8 (24.9)	0.086	0.003	0.328

Table 3: Two-way ANOVA analysis for questions used for caches quizzes. Sample size for Fall 2021 = 213 (Low Spatial Ability = 108, High Spatial Ability = 105). Sample size for Spring 2022 = 233 (Low Spatial Ability = 88, High Spatial Ability = 145). None of the questions showed significant interaction between spatial ability and instruction changes. Two question that show significant dependence on spatial ability were not significantly affected by the instruction changes. Three questions were significantly impacted by instruction changes with two of them showing improvement and one question showing a decrease in score.

5 Discussion

The results show that students' performance on the cache quizzes improved slightly due to instructional changes when taught using algorithmic hashing functions, but this performance did not help students with low spatial ability more than students with high spatial ability. The analysis of the individual questions revealed that students taught algorithmically got better at solving questions that involved sequences of large addresses (Q4 and Q6) presented in hexadecimal format, but not because of any effect on spatial ability. In contrast, these students performed worse on a question with small addresses (Q5) and a cache visualized as a table.

Q6 suggests that the improvements for students taught algorithmically may simply be because they got more practice in working with large hexadecimal addresses, increasing their comfort with the base and notation. Q6 asks students to calculate a sequence of memory addresses accessed for a given code without dealing with hashing functions but focusing only on locality observations. Additional practice with hexadecimal addresses would best explain the better performance on Q6, so that may also be the case for Q4.

The juxtaposition of Q5 and Q4 suggests another explanation. Q4 has 32-bit addresses for which position-based parsing is quite cumbersome and error prone, whereas Q5 has short, 8-bit addresses and shows the students how many bits are allocated to the tag, index, and offset by displaying a pre-filled cache visualized as a table like in Figure 1. Hashing functions scale easily with large addresses while bit position parsing does not. This contrast suggests that scaffolding students with different representations at different times may be best, which may be part of why Stieff et al. found the hybrid spatial and algorithmic instructional condition to be most effective [8] .

Anecdotally, the undergraduate course staff helping the students during group activities spontaneously reported that they found it easier to explain and answer the questions related to hash functions using modular arithmetic (in Spring 2022) as compared to the previous semester (Fall 2021) when answering questions about determining the number of bits for tag, index, and offset and then parsing. Determining the number of bits for tag, index and offset for a cache has been consistently tricky for students, so differences in performance may have been due to students' difficulties with this particular sub-task. Q4 in algorithmic instruction replaces the number of bits calculations with a slightly easier hash function, improving performance. In contrast, Q5 in spatial instruction gave students the number of bits for free, removing the tricky sub-task, but Q5 in algorithmic instruction still required determining and executing the hashing function, raising the difficulty bar slightly.

The questions that depend significantly on spatial ability are the cache performance analysis questions (Q2 and Q3). Our earlier study showed that understanding how large data structures map to the cache in a cyclic manner is an important factor in determining students' performance on this type of question [10]. These questions also used large hexadecimal addresses like Q4 and Q6, but that improvement on Q4 and Q6 did not transfer to Q2 and Q3. Future studies will need a more targeted focus on how spatial reasoning is a barrier for these specific types of tasks. Is the barrier reading/visualizing code? Are students struggling with creating visualizations or tools to aid their analysis? Is the challenge specifically with how large data structures in memory lead to many-to-one mappings and these many-to-one mappings are spatially difficult to comprehend? The lack of any difference in performance for number representations, suggests that the algorithmic hashing function is unlikely to be a promising direction for helping students with low spatial ability unfortunately.

A limitation of this study is the optional nature of the spatial ability quiz, which introduced self-selection bias into RQ2. The sample for RQ2 and additional analysis for questions in the cache score is limited to students who took the spatial ability quiz. The fact that the statistically significant difference in cache scores in RQ1 disappears as a main effect in the RQ2 results demonstrates the potential effect of this self-selection bias. This bias reduces the robustness of our findings, especially at the individual question level, making strong conclusions impossible. As a quasi-experimental study, there are almost certainly some unobserved differences between students in the two different semesters. We controlled for baseline performance and instructor as best we could, but there are probably still additional confounders that we cannot see.

6 Conclusion

In this paper, we described how we tried to address the achievement gap between students with low spatial ability and students with high spatial ability for the topics of number representations and caches within our computer architecture course. We shifted from spatially focused bit position reasoning to algorithm focused modular arithmetic functions. While there were some localized benefits to this shift in instructional methods, none of these improvements were tied to an interaction between the instructional change and spatial ability. We did not find evidence that this particular shift from spatial to algorithmic approach improved students' performance, though there might be other shifts away from spatial approach or a more intentional hybrid approach that may still be effective. For this reason, picking and reinforcing appropriate tools and skills for challenging topics may be a productive route for instructors to explore in their efforts to improve their teaching.

References

[1] Jonathan Wai, David Lubinski, and Camilla P. Benbow. Spatial ability for stem domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance. *Journal of Educational Psychology*, 101(4):817–835, 2009.

[2] Yukiko Maeda and So Yoon Yoon. A meta-analysis on gender differences in mental rotation ability measured by the purdue spatial visualization tests: Visualization of rotations (psvt: R). *Educational Psychology Review*, 25(1):69–94, 2013.

[3] S. Brownlow, T.K. McPheron, and C.N. Acks. Science background and spatial abilities in men and women. *Journal of Science Education and Technology*, 12:371–380, 2003.

[4] Miranda C. Parker, Amber Solomon, Brianna Pritchett, David A. Illingworth, Lauren E. Marguilieux, and Mark Guzdial. Socioeconomic status and computer science achievement: Spatial ability as a mediating variable in a novel model of understanding. In *2018 ACM Conference on International Computing Education Research*, pages 97–105, Espoo, Finland, August 2018. ACM.

[5] Susan C. Levine, Marina Vasilyeva, Stella F. Lourenco, Nora S. Newcombe, and Janellen Huttenlocher. Socioeconomic status modifies the sex difference in spatial skill. *Psychological Science*, 16(11):841–845, 2005.

[6] Andrew T. Stull, Mary Hegarty, Bonnie Dixon, and Mike Stieff. Representational translation with concrete models in organic chemistry. *Cognition and Instruction*, 30(4):404–434, 2012.

[7] Mary Hegarty, Mike Stieff, and Bonnie L. Dixon. Cognitive change in mental models with experience in the domain of organic chemistry. *Journal of Cognitive Psychology*, 25(2):220–228, 2013.

[8] M. Stieff, B. L. Dixon, M. Ryu, B. C. Kumi, and M. Hegarty. Strategy training eliminates sex differences in spatial problem solving in a stem domain. *Journal of Educational Psychology*, 106(2):390–402, 2014.

[9] Jonathan H Tomkin, Matthew West, and Geoffrey L Herman. An improved grade point average, with applications to cs undergraduate education analytics. *ACM Transactions on Computing Education (TOCE)*, 18(4):1–16, 2018.

[10] Suleman Mahmood and Geoffrey L Herman. A modular assessment for cache memories. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pages 1089–1095, 2021.

[11] Scipy.stats.pearsonr¶.

[12] John L Hennessy and David A Patterson. *Computer architecture: a quantitative approach*. Elsevier, 2011.

[13] David A Patterson and John L Hennessy. *Computer Organization and Design MIPS Edition: The Hardware/Software Interface*. Newnes, 2013.

[14] David F Lohman. Spatial ability and g. *Human abilities: Their nature and measurement*, 97(116):1, 1996.

[15] George M Bodner and Roland B Guay. The purdue visualization of rotations test. *The chemical educator*, 2(4):1–17, 1997.

[16] Carol J Ormand, Cathryn Manduca, Thomas F Shipley, Basil Tikoff, Cara L Harwood, Kinnari Atit, and Alexander P Boone. Evaluating geoscience students' spatial thinking skills in a multi-institutional classroom study. *Journal of Geoscience Education*, 62(1):146–154, 2014.

[17] Mary Hegarty, Madeleine Keehner, Peter Khooshabeh, and Daniel R Montello. How spatial abilities enhance, and are enhanced by, dental education. *Learning and Individual Differences*, 19(1):61–70, 2009.

[18] Ruth B Ekstrom and Harry Horace Harman. *Manual for kit of factor-referenced cognitive tests, 1976*. Educational testing service, 1976.

[19] Herman A Witkin, Carol Ann Moore, Donald R Goodenough, and Patricia W Cox. Field-dependent and field-independent cognitive styles and their educational implications. *Review of educational research*, 47(1):1–64, 1977.

[20] Sheryl A Sorby and Beverly J Baartmans. The development and assessment of a course for enhancing the 3-d spatial visualization skills of first year engineering students. *Journal of Engineering Education*, 89(3):301–307, 2000.

[21] Lauren E Margulieux. Spatial encoding strategy theory: The relationship between spatial skill and stem achievement. *ACM Inroads*, 11(1):65–75, 2020.

[22] Eric P Bettinger. Remedial and developmental courses. *Economic inequality and higher education: Access, persistence and success*, pages 69–100, 2007.

[23] Mike Stieff, Stephanie Scopelitis, Matthew E. Lira, and Dane DeSutter. Improving representational competence with concrete models. *Science Education*, 100(2):344–363, 2016.

[24] Geoffrey L. Herman, Craig Zilles, and Michael C. Loui. How do students misunderstand number representations? *Computer Science Education*, 21(3):289–312, 2011.

[25] Mike Stieff and Sonali Raje. Expert algorithmic and imagistic problem solving strategies in advanced chemistry. *Spatial Cognition & Computation*, 10(1):53–81, 2010.

[26] Jacob Cohen. *Statistical power analysis for the behavioral sciences*. Academic press, 2013.

A Example Quiz Question

In this section, we show a complete example of a question where students performance improved. Figure 7 shows the prompt for the question. The students are given a series of addresses which are accessed by a program. The students need to determine which of the accesses are cache hits or cache misses. The students need to calculate tag and index fields for each access. These values are then used to decide if an access is a hit or a miss. In the spatially focused method, students would first calculate the size of tag and index fields. For the given example tag field size is 20-bits and the index field is 5-bits. Then they would convert the addresses to binary representation to parse the addresses into tag and index fields. At this point students could maintain the cache state using binary values for the tag or convert it back to hex. If cache state is maintained using binary values then they need to compare long binary numbers to determine hit or miss for each access. In the given example the binary values to compare would be 20-bits long.

In contrast, using the algorithmic approach, the students can derive the mapping formulas from the cache size. For tag this translates to a modulus with cache size i.e. $address\%4096$. And for index the calculation would be $(address/128)\%32$ where 32 is the number of set is the cache. These calculations result in hexadecimal tag value

Consider a 32-bit processor that uses a **direct-mapped** cache with a total capacity of 4KB with 128B cache blocks. Reminder: K = 2^{10}

The table below provides a list of memory address that a program accesses, where the top address is accessed first. Assume the cache is initially empty, so that the first access is a miss. Which of these accesses result in a cache hit?

Address	Hit/Miss
Load `0xf5f3fb39`	**Miss**
Load `0xf5f3fa55`	○ (a) Hit ○ (b) Miss
Load `0xc53f8b43`	○ (a) Hit ○ (b) Miss
Load `0xf5f3fb06`	○ (a) Hit ○ (b) Miss
Load `0xf5f3fa70`	○ (a) Hit ○ (b) Miss
Load `0xc53f8b30`	○ (a) Hit ○ (b) Miss

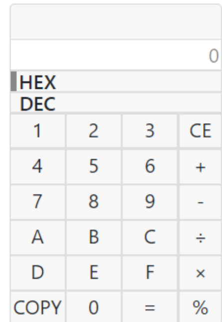


Figure 7: The figure shows prompt for one of the quiz questions. The students are required to determine if the given accessed addresses are cache hits or misses. The calculator supporting hexadecimal arithmetic is provided to help the students in calculating the tag and index fields using the cache mapping formulas.

which is just 5 digits. This makes hexadecimal values easier to compare than binary values. Students also do not have to image dividing the addresses into different fields. The solution for the problem where we show the calculated tag and index values for each accessed address is shown in Figure 8.

Address	Hit/Miss
Load `0xf5f3fb39`	**Miss**
Load `0xf5f3fa55`	(b) Miss
Load `0xc53f8b43`	(b) Miss
Load `0xf5f3fb06`	(b) Miss
Load `0xf5f3fa70`	(a) Hit
Load `0xc53f8b30`	(b) Miss

- `0xf5f3fb39`: Miss (Invalid) with Tag 0xf5f3f, Index 0x16
- `0xf5f3fa55`: Miss (invalid) with Tag 0xf5f3f, Index 0x14
- `0xc53f8b43`: Miss (tag mismatch) with Tag 0xc53f8, Index 0x16
- `0xf5f3fb06`: Miss (tag mismatch) with Tag 0xf5f3f, Index 0x16
- `0xf5f3fa70`: Hit with Tag 0xf5f3f, Index 0x14
- `0xc53f8b30`: Miss (tag mismatch) with Tag 0xc53f8, Index 0x16

Figure 8: Solution to the exam question. This solution is visible to the students only after student have exhausted all the attempts for the question. We show the tag and index fields for the accessed address and we also show the reason why the access was a cache hit or a cache miss.