ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE

Paper ID #36778

Work in Progress: Exploring Students' Misconceptions of Cache Memories

Muhammad 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.

Hongxuan Chen

Hongxuan Chen is a Ph.D. student in the Department of Computer Science at the University of Illinois, Urbana-Champaign, where he also completed his B.S. in Computer Science. He is broadly interested in how students learn computer science and broadening participation in computer science.

Morgan M Fong

Morgan M. Fong is a Ph.D. student in the Department of Computer Science at the University of Illinois, Urbana-Champaign and an NSF Graduate Research Fellow. Prior to starting her Ph.D. Morgan completed her B.A. in Computer Science at the University of California, Berkeley. Her current research focuses on developing methods for observing and analyzing cooperative learning in undergraduate computing courses.

Geoffrey Herman

Dr. Geoffrey L. Herman is the Severns Teaching Associate Professor with the Department of Computer Science at the University of Illinois at Urbana-Champaign. He earned his Ph.D. in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign as a Mavis Future Faculty Fellow and conducted postdoctoral research with Ruth Streveler in the School of Engineering Education at Purdue University. His research interests include creating systems for sustainable improvement in engineering education, conceptual change and development in engineering students, and change in faculty beliefs about teaching and learning. He is an associate editor with the Journal of Engineering Education and a board member of the Computing Research Association Education committee.

Work in Progress: Exploring Students' Misconceptions of Cache Memories

Introduction

Caches are small memories inside a computer's processor that reduce the average time to access memory for a program. Caches store a small amount of recently accessed data inside the processor so that it can be accessed quickly by the processor. ACM Computer Science Curricula 2013 [1] classifies the purpose and operation of caches as a core topic. Programmers need to know how caches work and its implications for the order in which data should be accessed to maximize computing performance. Prior research shows that cache concepts are difficult to learn for students. Porter et al. [2] found that fewer than 40% of students were able to correctly answer basic questions related to cache structure after completing a full computer architecture course. Our earlier quantitative investigations showed that students struggled in finding the location where data from the main memory should be stored in cache. They also struggled tracking the state of set-associative caches (caches where data from one memory address can be stored in more than one place in the cache) and their replacement policies which decide what data must be removed from the cache to make room for new data [3].

In this paper, we describe our ongoing qualitative exploration of the methods that students use to map data to caches, especially when cyclic mapping (memory addresses at regular intervals mapping to the same location in the cache) needs to be considered. We also explore how students track Least Recently Used (LRU) replacement polices for set-associative caches. Implementing LRU replacement policy requires tracking when each block of data stored in cache was last accessed. We investigate the processes students use for the tasks mentioned above and the mistakes that hinder their progress. We also analyze the mistakes to discover the underlying misconceptions that lead to those mistakes. We believe that this investigation will help us in designing representations and tools to help students develop a better understanding of caches. We intend to answer the following research questions.

RQ1: What processes, mistakes, and misconceptions do students show in calculating where a memory address maps to cache?

RQ2: What processes, mistakes, and misconceptions do students show in calculating the mapping of different elements of a large array to the cache when cyclic mapping needs to be considered?

RQ3: What processes, mistakes, and misconceptions do students show in tracking the Least Recently Used (LRU) replacement policy in set-associative caches?

Background

Cache Concepts and Misconceptions

There is little research in identifying students' processes and misconceptions related to caches. Grigoriadou et al. [4, 5] interviewed 20 students to find the difficulties they faced related to cache design and operation. They found that students exhibited partial or faulty knowledge of cache memory operation. Sahuquillo et al. [6] identified three major tasks related to caches that students must be able to perform. The tasks were mapping addresses to cache, identifying spatial and temporal locality, and tracing replacement policies. To facilitate students' learning for these tasks, they extended SPIM [7], a commonly used simulator for MIPS architecture, to include cache simulation. The work done by Sahuquillo et al [6] and Grigoriadou et al. [5] provide a start but neither provides rich descriptions of students' processes for working, or misconceptions of, caches. We seek to address this gap.

Exploring Student's Misconceptions

There is a large body of research documenting common student misconceptions in science, technology, engineering, and math (STEM) fields [8, 9, 10, 11, 12]. Identifying common misconceptions is an essential step in developing concept inventories and instruments for assessing different pedagogical approaches [8, 13, 14, 15].

Research Methods

We are using think-aloud interviews to explore our research questions. Think-aloud interviews [16] are commonly used for exploring students' misconceptions [8]. Students are given a set of questions to solve and asked to explain their reasoning for the steps that they perform while solving the questions. In computing contexts, this method has been used for exploring students' misconceptions about introductory programming concepts [17], state of systems [18], data structures [17] among other computing concepts [19, 20, 15]. These interviews are commonly analyzed using inductive coding techniques to form theories about students' misconceptions [21].

This research is being conducted in a computer architecture course at University of Illinois Urbana-Champaign. Computer Architecture is a required course for computer science majors and enrolls between 300 and 400 students each semester.

Interview Protocol

The interview protocol included one question requiring students to map given addresses to a cache, two questions about the cyclic mapping of 1-D and 2-D arrays to a cache, and two questions about the replacement policy in 2-way and 4-way set-associative caches. The first question (mapping addresses) also served as a warm-up for students before they tackled more complicated questions. The protocol was designed so that the whole interview process was expected to take students one hour to complete, and three pilot interviews with students who had recently completed the course showed that one hour was sufficient for the interviews. The questions related to replacement policy were updated based on the pilot interviews. The final interview protocol document showed one question per page to minimize the effect that one question might have on students' reasoning on other questions.

Since we had two questions each for exploring cyclic mapping and replacement policies, the order of the questions could have influenced the information that students were able to recall and use in later questions. We addressed this by flipping the order of the questions related to the same concepts while interviewing different students. This gave us samples where students approached each question without any conditioning.

Data Collection

Initial interviews were conducted in Spring 2021 semester. Computer architecture students learned about caches in Weeks 11 and 12 with a quiz in the middle of Week 13. We invited students to participate in the interview at the end of week 13. Interviews were conducted in Weeks 14 and 15 of the semester. Ten students volunteered for the interview and were compensated $10 for their participation. The interview was conducted over Zoom. Video and audio were recorded during the interview. At the start of the interview, the interviewees were briefed about the purpose of the study. Students were asked to direct the camera towards their scratch paper so that their process to solve the problem could be observed. The average time for each interview was approximately 30 minutes. The interview protocol, recruitment, and interview procedures were approved by the Institutional Review Board at the University.

Our preliminary analysis of the data collected in the first round indicated that it was possible to solve questions related to LRU policy with minimal tracking of the cache state. We updated these questions so that tracking full state of the cache was required to solve the questions. The updated protocol was used in data collection in Fall 2021 semester. Analysis of Fall 2021 data is still in progress.

Data Analysis

Interviews conducted in Spring 2021 were transcribed before analysis. We used oTranscribe [22] to help with transcriptions. The transcripts contained pictures of the scratch paper and descriptions of students' actions. The transcripts were analyzed through the constant comparative method [23]. Constant comparative method is an inductive coding technique that takes a disciplined approach to comparing the different actions that an individual participant does over time and across participants to make sure that we are making robust comparisons between participants so that we can draw more robust conclusions. For our data there were two levels of comparisons that needed to be performed: comparisons within a single interview and comparisons across interviews.

Our research team consists of four members. First, each team member inductively coded each transcript independently. The team members could access the videos if something was not clear in the transcript. At this time, each researcher identified the steps and mistakes in each question. They also formed hypothesis about the misconceptions that led to the mistakes based on data. At this point, first level of comparison, i.e., comparisons within a single interview showed how students' approach to later questions changed as they recalled more information about caches while solving the questions.

After the independent analysis, the team members met to discuss each interview sequentially.

Discussing each interview completely before moving on to the next made it easier to compare a student's approach to different questions. During the discussion, all team members shared what they believed about the student's processes, mistakes, and the underlying misconceptions. All disagreements at this point were resolved through discussion. In some cases, this required watching the original recorded interview to get a better sense of the student's tone and what the student was writing at that specific point. During this discussion, the agreed upon interpretation of the student's actions and words were documented in copies of the transcripts in Microsoft Word. These annotated copies were used in the second phase of the analysis.

In the second phase of the analysis, each team member independently synthesized their codes to identify cross-cutting themes, considering all interviews to identify common methods and mistakes. This step involved the second level of comparisons, i.e., comparison across interviews. After independent analysis for themes, the research team met to discuss the identified themes. During this discussion, descriptions of agreed upon themes were written down. Annotations that contributed to each theme were also listed with the theme. The themes followed a general pattern of identifying an observed step, mistakes related to that step and the underlying misconceptions. We intend to use the same steps to analyze data from Fall 2021.

Trustworthiness and Reliability

Every team member brings their own experiences that can bias their interpretation of the data. In our team, two team members had been working to improve instructional methods for caches for several semesters. The efforts mostly focused on developing assessments and tools to help students learn caches. These members of the research team may have developed hypotheses about student processes which may bias their analysis. To mitigate the effect of prior hypotheses, we had two additional team members who previously worked as teaching assistants for the computer architecture course but had not worked on developing material for caches. We also maintained an audit trail to track the evolution of our interpretation of data. We did this by setting the documents to use the track changes feature. This audit trail can be examined to find any major changes in our interpretation of the data during the discussion phase.

Limitations

The results of this study are limited to the questions explored in a single protocol, limiting the study to explore only a few concepts in limited contexts. It is also a possibility that thinking aloud will change a student's thought process. One student commented during the interview that "thinking aloud really helps". This means we may not be observing the thought process that students may have while solving problems during an actual assignment or exam. We kept interactions limited to prompting the students to think aloud during the interview process to minimize any interruption and consequent changes in a students' train of thought. We also recognize that the themes identified in this study may not be the most frequently observed misconceptions and processes used by students, but the purpose of this study is to find misconceptions and processes rather than their prevalence.

Preliminary Results

We present a summary of the common observed processes and misconceptions. Mistakes are specific to the questions in the interview protocol, so they have been omitted for the sake of brevity.

RQ1: Processes and Misconception in Mapping Addresses

Students started by calculating cache parameters like the size of tag, index, and offset (sizes of parts of a memory address used to find where the data must be stored in cache) even in questions where it was not necessary. To map an address to a cache, students either converted the hexadecimal address to binary before breaking it down into tag, index, and offset fields using the sizes of their respective parts, or they used division and modulus operations to determine only the index and offset. There were two major misconceptions regarding the data that gets cached when an address is accessed: 1) only the part of the block following the accessed address is cached, and 2) the cached block always starts at the accessed address. Confusing the tag for the index, e.g., using tag size calculation formula for index size calculation, and off-by-one errors in calculating block boundaries were also observed.

RQ2: Cyclic Mapping for Large Arrays

The questions in the protocol asked only for mapping of specific elements, but many students used drawings to visualize how arrays filled the cache. Students made two major assumptions while drawing these diagrams: 1) the start of array maps to the start of the cache even though the address of the array was not given, and 2) the array is being accessed sequentially even though only a single element was being accessed. Off-by-one errors were common in drawings and calculations. A common misconception for set-associative cache was the belief that blocks in distinct cache ways have a different set of indices. Some students also mapped the array indices directly to the cache set index.

RQ3: Set-associative Caches and Replacement Policies Tracking

Students recognized that both the tag and index needed to match for a cache hit. Tracking least recently used state was a required step, but students commonly did not perform this step while updating that state of the cache on each access, particularly for 2-way set associative caches. Some students went back to the beginning of the question to recalculate the state for the replacement policy when data needed to be replaced in the cache. Some students completely ignored the replacement policy and either assumed that cache ways are accessed and replaced in a cyclic order or assumed that everything that had been accessed would stay in the cache.

Work in Progress

As mentioned in the methods section, a second round of interviews with updated protocol was conducted in Fall 2021 semester. We intend to analyze the data collected from these interviews using the same methods. We hope that the additional data will help us in refining our understanding of the processes and misconceptions that were extracted from Spring 2021 data. The results of this study will be used to design new representations and pedagogical approaches to address the misconceptions identified here.

References

[1] ACM Computing Curricula Task Force, editor. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, Inc, jan 2013.

[2] Leo Porter, Saturnino Garcia, Hung-Wei Tseng, and Daniel Zingaro. Evaluating student understanding of core concepts in computer architecture. In *Proceedings of the 18th ACM conference on Innovation and technology in computer science education*, pages 279–284. ACM, 2013.

[3] 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.

[4] Maria Grigoriadou, Evangelos Kanidis, and Agoritsa Gogoulou. A web-based educational environment for teaching the computer cache memory. *IEEE Transactions on Education*, 49(1):147–156, 2006.

[5] Maria Grigoriadou, Maria Toula, and Evangelos Kanidis. Design and evaluation of a cache memory simulation program. In *Proceedings 3rd IEEE International Conference on Advanced Technologies*, pages 170–174. IEEE, 2003.

[6] Julio Sahuquillo, Noel Tomas, Salvador Petit, and Ana Pont. Spim-cache: A pedagogical tool for teaching cache memories through code-based exercises. *IEEE Transactions on Education*, 50(3):244–250, aug 2007.

[7] James Larus. Spim: A mips32 simulator. http://spimsimulator.sourceforge.net/. Accessed on Mar 22, 2022.

[8] Ruth A. Streveler, Shane Brown, Geoffrey L. Herman, and Devlin Montfort. *Conceptual Change and Misconceptions in Engineering Education: Curriculum, Measurement, and Theory-Focused Approaches*, page 83–102. Cambridge University Press, 2014.

[9] Ross H Nehm and Leah Reilly. Biology majors' knowledge and misconceptions of natural selection. *BioScience*, 57(3):263–272, 2007.

[10] Musa Dikmenli. Misconceptions of cell division held by student teachers in biology: A drawing analysis. *Scientific Research and Essays*, 5(2):235–247, 2010.

[11] Haluk Özmen. Some student misconceptions in chemistry: A literature review of chemical bonding. *Journal of Science Education and Technology*, 13(2):147–159, 2004.

[12] Peter A Tanner, Lei Zhou, Changkui Duan, and Ka-Leung Wong. Misconceptions in electronic energy transfer: bridging the gap between chemistry and physics. *Chemical Society Reviews*, 47(14):5234–5265, 2018.

[13] Geoffrey L. Herman, Craig Zilles, and Michael C. Loui. A psychometric evaluation of the digital logic concept inventory. *Computer Science Education*, 24(4):277–303, 2014.

[14] A. E. Tew and M. (2011) Guzdial. The fcs1: A language independent assessment of cs1 knowledge. In *In Proceedings of the 41st ACM technical symposium on computer science education (SIGCSE)*, page 111–116, Dallas, TX, 2011.

[15] Leo Porter, Daniel Zingaro, Soohyun Nam Liao, Cynthia Taylor, Kevin C. Webb, Cynthia Lee, and Michael Clancy. Bdsi: A validated concept inventory for basic data structures. In *Proceedings of the 2019 ACM Conference on International Computing Education Research*, ICER '19, page 111–119, New York, NY, USA, 2019. Association for Computing Machinery.

[16] K Anders Ericsson and Herbert A Simon. *Protocol analysis: Verbal reports as data.* the MIT Press, 1984.

[17] Lisa C. Kaczmarczyk, Elizabeth R. Petrick, J. Philip East, and Geoffrey L. Herman. Identifying student misconceptions of programming. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, SIGCSE '10, page 107–111, New York, NY, USA, 2010. Association for Computing Machinery.

[18] Geoffrey L. Herman, Craig Zilles, and Michael C. Loui. Flip-flops in students' conceptions of state. *IEEE Transactions on Education*, 2012.

[19] Geoffrey L Herman, Michael C Loui, Lisa Kaczmarczyk, and Craig Zilles. Describing the what and why of students' difficulties in boolean logic. *ACM Transactions on Computing Education (TOCE)*, 12(1):1–28, 2012.

[20] G. L. Herman, M. C. Loui, and C. Zilles. Students' misconceptions about medium-scale integrated circuits. *IEEE Transactions on Education*, 54(4):637–645, 2011.

[21] Kathy Charmaz and J Smith. Grounded theory. *Qualitative psychology: A practical guide to research methods*, 2:81–110, 2003.

[22] Elliot Bentley. otranscribe. https://otranscribe.com/. Accessed on May 12, 2022.

[23] Hennie Boeije. A purposeful approach to the constant comparative method in the analysis of qualitative interviews. *Quality and quantity*, 36(4):391–409, 2002.