The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center
Portland, OR . June 23 - 26, 2024
ASEE
Paper ID #41341

An Enhanced Learning Method Used for Datapath Design Topics in Computer Engineering Curriculum

Dr. Tingjun Lei, Mississippi State University

Dr. Tingjun Lei is currently a Postdoctoral Research Fellow in the Department of Electrical and Computer Engineering at the Mississippi State University (MSU). He received his Ph.D. degree in electrical and computer engineering with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS, USA., in 2023, his M.S. degree in electrical and computer engineering from the New York Institute of Technology, Old Westbury, NY, USA, in 2016, and the B.S. degree in intelligent transportation engineering from Shanghai Maritime University, Shanghai, China, in 2014. He was Graduate Teaching Assistant for ECE1013 Foundations in ECE, ECE1022 Foundations in Design, ECE4713/6713 Computer Architecture, and ECE4753/6753 Introduction to Robotics at the undergraduate level and as a guest lecturer delivered graduate-level courses, ECE 8743 Advanced Robotics and ECE8833 Computational Intelligence. He received the ECE Best Graduate Researcher Award from the Department of Electrical and Computer Engineering, Mississippi State University in 2023. He received the Research Travel Award from Bagley College of Engineering, Mississippi State University in 2023. His two papers have been selected and featured as cover articles on Intelligence & Robotics Journal. He won six oral and poster presentation awards at multiple conferences. Dr. Lei received the Best Paper Award in 2022 International Conference on Swarm Intelligence. Dr. Lei serves as Youth Editorial Board Member of Intelligence and Robotics. Dr. Lei has served on the technical program committee for numerous international conferences, such as IEEE-CEC, IEEE-IJCNN, ICSI, and PRIS, etc. Dr. Lei has extensively published journal and conference papers in robotics, intelligent systems, and engineering education areas. His research interests include engineering education, robotics and autonomous systems, human robot interaction, deep learning, intelligent transportation systems, and evolutionary computation.

Mr. Timothy Sellers, Mississippi State University

Timothy Sellers received the B.S. degree in robotics and automation technology and applied science in electro-mechanical engineering from the Alcorn State University, Lorman, MS, USA in 2020. He is currently pursuing a Ph.D. degree in the Department of Electrical and Computer Engineering at Mississippi State University, Mississippi State, MS, USA. He is currently a Graduate Teaching Assistant for Senior Design II (ECE4542/ECE4522) and was for Advance Circuits (ECE3434) at the undergraduate level and as guest lecturer delivered graduate-level courses, Advanced Robotics (ECE 8743) and Computational Intelligence (ECE 8833). He received the ECE Outstanding Teaching Assistant Award from the Department of Electrical and Computer Engineering, Mississippi State University in 2021. He received the Research Travel Award from Bagley College of Engineering, Mississippi State University in 2024. He has also received the Bagley College of Engineering Student Hall of Fame award in 2024. He won three poster presentation awards at multiple conferences. Mr. Sellers has served on the technical program committee for numerous international conferences and journals, such as IJMLC, ICSI, and PRIS, etc. He has extensively published journal and conference papers in engineering education and robotics fields. His research interests include engineering education, robotics and autonomous systems, human robot interaction, deep learning, and computational intelligence.

Prof. Chaomin Luo, Mississippi State University

Dr. Chaomin Luo (Senior Member, IEEE) holds a Ph.D. degree in electrical and computer engineering from the Department of Electrical and Computer Engineering at the University of Waterloo, Canada in 2008. He also earned an M.Sc. degree in engineering systems and computing from the University of Guelph, Canada in 2002, and a B.Sc. in electrical engineering from Southeast University. Currently, he is an Associate Professor in the Department of Electrical and Computer Engineering at Mississippi State University. His research interests include engineering education, intelligent systems, control and automation, robotics, and autonomous systems. He is Associate Editor in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2019). He is Tutorials Co-Chair in the 2020 IEEE

Symposium Series on Computational Intelligence. Dr. Luo was the recipient of the Best Paper Awards in IEEE International Conference on Information and Automation, International Conference on Swarm Intelligence, and SWORD Conference. His research interests include Robotics, Autonomous Systems, and Control and Automation. Dr. Luo is an IEEE senior member, INFORMS, and ASEE member. Dr. Luo is active nationally and internationally in his research field. He was the Program Co-Chair in 2018 IEEE International Conference on Information and Automation (IEEE-ICIA'2018). He was the Plenary Session Co-Chair in the 2021 and 2019 International Conference on Swarm Intelligence, and he was the Invited Session Co-Chair in the 2017 International Conference on Swarm Intelligence. He was the General Co-Chair of the 1st IEEE International Workshop on Computational Intelligence in Smart Technologies (IEEE-CIST 2015), and Journal Special Issues Chair, IEEE 2016 International Conference on Smart Technologies (IEEE-SmarTech), Cleveland, OH, USA. He was Chair and Vice Chair of IEEE SEM - Computational Intelligence Chapter and was a Chair of IEEE SEM - Computational Intelligence Chapter and Chair of Education Committee of IEEE SEM. He has organized and chaired several special sessions on topics of Intelligent Vehicle Systems and Bio-inspired Intelligence in reputed international conferences such as IJCNN, IEEE-SSCI, IEEE-CEC, IEEE-CASE, and IEEE-Fuzzy, etc. He has extensively published in reputed journals and conference proceedings, such as IEEE Transactions on Industrial Electronics, IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on SMC, IEEE Transactions on Cybernetics, IEEE-ICRA, and IEEE-IROS, etc. Dr. Luo serves as Associate Editor of IEEE Transactions on Cognitive and Developmental Systems, International journal of Robotics and Automation, and Associate Editor of International Journal of Swarm Intelligence Research (IJSIR).

Prof. Gene Eu Jan, Tainan National University of the Arts

Gene Eu Jan (M'00) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1982 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, MD, USA, in 1988 and 1992, respectively. He has been a Professor with the Departments of Computer Science and Electrical Engineering, National Taipei University, New Taipei City, Taiwan since 2004, where he also served as the Dean of the College of Electrical Engineering and Computer Science from 2007 to 2009. Currently, he is the president of Tainan National University of the Arts. He has published more than 270 articles related to parallel computer systems, interconnection networks, path planning, electronic design automation, and VLSI systems design in journals, conference proceedings, and books.

Prof. Zhuming Bi, Purdue University, Fort Wayne

Zhuming Bi (Senior Member, IEEE) received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 1994, and the Ph.D. degree from the University of Saskatchewan, Saskatoon, SK, Canada, in 2002. He has international work experience in Mainland China, Hong Kong, Singapore, Canada, UK, Finland, and USA. He is currently a professor of Mechanical Engineering with Purdue University Fort Wayne, Fort Wayne, IN, USA. His current research interests include robotics, mechatronics, Internet of Things (IoT), digital manufacturing, automatic robotic processing, and enterprise information systems. He has published 6 research books and over 180 journal publications in these fields.

An Enhanced Learning Method Used for Datapath Design Topics in Computer Engineering Curriculum

Tingjun Lei[1], Timothy Sellers[1], Chaomin Luo[1], Gene Eu Jan[2], and Zhuming Bi[3]
[1]Department of Electrical and Computer Engineering, Mississippi State University
[2]National Taipei University, and Tainan National University of the Arts
[3]Department of Civil and Mechanical Engineering, Purdue University Fort Wayne

Abstract

In the Computer Architecture curriculum, teaching the complex topics of single cycle and pipelined datapath to senior students is challenging, which this paper addresses through a hybrid pedagogy combining Technological Pedagogical Knowledge with flipped learning (TPK-FL) and multiple practice exercises. TPK explores the technology-pedagogy interaction, particularly in datapath design. This approach enables students to construct knowledge by integrating technological tools with pedagogical strategies. Flipped learning (FL) reverses traditional teaching methods by providing course materials on datapath design beforehand, fostering active, self-directed learning in the classroom. The pedagogy is enriched with structured practice exercises, enhancing students' understanding of datapath design, along with their problem-solving, analytical, and critical thinking skills. The effectiveness of this method is validated through various assessments, including homework, exams, Q&A sessions, and student feedback, with a positive comparison to the instructor's previous teaching experiences. This holistic evaluation confirms the efficacy of this innovative approach in improving the learning experience in Computer Architecture and Engineering education, specifically in the areas of single cycle and pipelined datapath design. The approach improves student performance, enhancing learning outcomes in Computer Architecture and Engineering curriculum.

1 Introduction

In engineering education, there is a significant thrust towards crafting students into capable professionals. Pursuing this goal has involved delving into and embracing various pedagogical approaches, including differentiated learning [1–6], project-based [7–10], inquiry-based [11–14, 24], and collaborative [15–19]. Each approach imparts a distinctive flavor to the educational experience, thereby enriching the academic journey for those studying engineering.

Differentiated Learning customizes instruction to meet the unique needs and learning preferences of different students. Menekse *et al.* [1] study on the Differentiated Overt Learning Activities (DOLA) framework in engineering education found that interactive activities lead to higher student learning gains, supporting the ICAP hypothesis. However, the study mainly focused on short-term impacts without addressing long-term effectiveness. Kolloffel and Jong [2] compared traditional instruction with inquiry learning in a virtual lab for electrical engineering, observing significant improvements in students' conceptual and procedural skills, especially in complex problem-solving. The study's limitations included its quasi-experimental design and focus on a specific student group. Cheng *et al.* [3] meta-analysis on flipped classrooms showed a positive impact on student learning outcomes across disciplines, but the study relied on existing literature and failed to extensively cover diverse educational settings or long-term effects. Zervoudakis *et al.* [5] developed a particle swarm optimization-based algorithm for student classification in differentiated instruction, focusing on technical efficiency over practical educational application. Together, these studies underscore the effectiveness of interactive teaching methods, while also highlighting the need for more comprehensive research in varied educational contexts.

Project Learning is based on active participation in hands-on projects, cultivating practical skills through immersive learning experiences. Leite combined project-oriented research with PBL, highlighting the need for adaptability in modernizing education [7]. Hosseinzadeh and Hesamzadeh found PBL effective in specialized subjects but noted resource requirements and suitability for moderate class sizes [8]. Rodriguez *et al.* [9] implemented PBL in space education, enhancing motivation but demanding more faculty commitment and facing scalability challenges. Marasc and Bejkat interdisciplinary program aimed to boost electrical engineering interest by integrating it with other subjects [10]. These studies collectively emphasize the benefits and logistical considerations of PBL and innovative approaches in engineering education.

Inquiry Learning is centered around nurturing students' curiosity and investigative abilities, motivating them to actively pursue knowledge by engaging in questioning and exploration. Huijuan *et al.* improved problem-solving skills in an electric machinery course, with positive short-term student feedback and improved grades [11]. Yuliati *et al.* [12] found that inquiry-based learning with simulations enhanced problem-solving in physics students, though the study focused more on methods than conceptual understanding. Haryudo *et al.* [13] developed a learning tool for industrial control, receiving positive initial feedback but lacking long-term skill applicability assessment. Eppes *et al.* [14] created a course integrating inquiry-based learning to advance student research, with immediate benefits observed but no long-term career outcome evaluation. These studies collectively demonstrate the positive impact of inquiry-based learning in scientific education, albeit with a need for more extensive, long-term evaluations.

Dickerson *et al.* [20] employed a distinctive approach to foster reflection among engineering students within the context of a digital circuits course. This method integrated computer-based simulation for digital circuit design with reflective thought prompts administered after a midterm exam for post-exam analysis and contemplation. The study also underscored the significance of employing thought-provoking question prompts designed to voluntarily elicit comprehensive reflections after a significant milestone event, such as a midterm exam, as opposed to a quiz [21]. A new student-centered teaching approach founded on four pedagogical domains: scaffolded, universally designed, mastery-based, and gameful learning [22]. This approach was applied in an engineering course by initially breaking down the content into small, easily digestible topics [23]. A framework has been developed for scaffolding, monitoring, and evaluating teamwork in a capstone engineering design course with industry-sponsored projects [25].

Martin-Gutierrez *et al.* [15] implemented augmented reality in an electrical engineering course, with the study primarily assessing immediate student feedback and not delving into the long-term educational impact. In the realm of Collaborative Learning, the focus is on group dynamics, which bolsters understanding through the exchange of shared ideas. Meanwhile, Hadfield-Menell *et al.* [17] study on cooperative inverse reinforcement learning concentrated on theoretical aspects, lacking practical, real-world validation. Additionally, Vliet *et al.* [19] explored the effects of flipped-class pedagogy on student motivation and learning strategies, noting improvements in critical thinking and peer learning. However, the observed benefits were short-lived, suggesting the necessity for continuous application.

In the Computer Architecture curriculum, teaching the challenging topics of single cycle and pipelined datapath design to senior students requires an innovative approach beyond traditional textbook methods. This paper introduces a hybrid pedagogy that combines Technological Pedagogical Knowledge with a flipped learning (TPK-FL) method and multiple practice exercises. The Technological Pedagogical Knowledge (TPK) approach explores the interaction between technology and pedagogy, specifically applied to datapath design. The flipped learning method alters the conventional teaching model by delivering course materials on datapath design to students beforehand, promoting active learning and self-direction in the classroom. This method is integrated with ongoing practice exercises aimed at enhancing students' understanding of datapath design, as well as their problem-solving, analysis, and critical thinking skills. The effectiveness of this pedagogical approach is evidenced by improved student performance in various assessments, indicating a significant enhancement in learning outcomes for datapath design in the Computer Architecture and Engineering curriculum.

In this paper, a hybrid pedagogy consisting of a Technological Pedagogical Knowledge (TPK) based method associated with a flipped learning method, and multiple practice exercises is implemented to create learning opportunities that allow students to construct their knowledge (the K) of the technological/tool (the T) through the pedagogical module (the P). TPK is used to explore correlations and interactions between technological tools and specific pedagogical practices. The multi-practice-exercise enabled method (MPEM) was utilized to construct the TPK of senior students in the context of designing single cycle datapath and pipelined datapath in Computer Architecture course. In particular, in this research, a series of well-organized multiple practice exercises was prepared to cover a variety of inherently connected topics in the datapath design timely. The flipped learning method in education is an instructional approach in the classroom that reverses the traditional model of teaching on this topic. In this flipped learning, the primary shift is in the way instructional content is delivered and how class time is used. Parts of course materials on single cycle datapath and pipelined datapath are carefully planned and assigned to students early, before they can get involved in the learning, discussion of the materials in the classroom. Thus, the flipped learning method leverages technology and active learning strategies to enhance the learning experience, encouraging students to be more self-directed in the classroom and promoting deeper understanding of the course materials on single cycle datapath and pipelined datapath.

This on-going practice exercises integrated with the flipped learning pedagogy are designed to not only improve students' learning outcomes and understanding of datapath design content, but also to develop and enhance their problem solving, problem analysis and critical thinking skills. Various practice exercises corresponding to the datapath design assigned to students just after the various lectures in support of better understanding datapath design was elaborately managed. In light of performance of students through homework assignments, Q/A sessions, exams, self-assessment survey and students' input to the official course evaluation administered by the university, and a comparison to the instructor's previous years of teaching experience, the adopted hybrid pedagogy effectively and efficiently enhanced students to learn datapath design meaningfully. Final assessment and evaluation of this new hybrid pedagogy for providing Computer Architecture and Computer Engineering education through obtained valuable information supports its effectiveness.

2 Topical Guide Objective Pedagogy Fused with On-Going Assignment Learning

This paper introduces a hybrid teaching method combining Technological Pedagogical Knowledge (TPK) with flipped classroom techniques and multiple practice exercises to enhance computer architecture students' grasp of instruction set design and principles. By leveraging TPK, the approach seeks to deepen both qualitative and quantitative understanding, encouraging active participation through pre-class preparation and in-class activities. This method facilitates a more engaged learning environment, allowing students to construct their knowledge actively and apply it to complex engineering tasks. The focus is on improving students' problem-solving, critical thinking, and analytical skills, essential for their success in the electrical and computer engineering fields.

2.1 TPK Pedagogy with On-going Assignments

A comprehensive suite of in-class exercises, practice questions, homework assignments, review problems, and exams are designed to encompass a broad range of topics in this course, utilizing a hybrid pedagogical approach that integrates TPK with flipped learning (FL) methodologies. This approach emphasizes active, student-centered learning through carefully structured assignments that encourage students to construct their understanding actively and engage deeply with the material. Learning outcomes are enhanced through the analysis of ongoing assignments, fostering a deeper comprehension of the course content. Student performance and understanding are assessed through various milestone review sessions and interactive activities, ensuring a well-rounded educational experience.

In our hybrid teaching approach, leveraging TPK and flipped classroom methodologies, we target learning objectives as follows:

- Developing a comprehensive understanding of both foundational and advanced principles of computer organization and architecture.
- Gaining the ability to critically analyze and assess the performance of CPU and memory hierarchies.
- Acquiring skills to analyze, design, and evaluate CPU microprocessors, datapaths, and pipeline architectures.
- Enhancing understanding of performance analysis, memory systems, and I/O interfacing concepts.
- Cultivating the capability to propose alternative designs and conduct evaluations of control units, pipelines, arithmetic and logic units, and identify and mitigate hazards in pipelined

datapaths.

Our hybrid teaching approach, TPK-FL, yields several positive outcomes aligned with our targeted learning objectives. Firstly, students develop a comprehensive understanding of both foundational and advanced principles of computer organization and architecture. This lays a solid groundwork for their comprehension of complex computing systems. Secondly, they gain the ability to critically analyze and assess CPU and memory hierarchies, honing their skills in evaluating system performance effectively. Furthermore, students acquire practical skills in analyzing, designing, and evaluating CPU microprocessors, datapaths, and pipeline architectures, enhancing their capacity to contribute meaningfully to hardware design and optimization. Additionally, our approach fosters a deeper understanding of performance analysis, memory systems, and I/O interfacing concepts, equipping students with essential knowledge for real-world application. Lastly, students cultivate the capability to propose alternative designs and conduct evaluations of critical components such as control units, pipelines, and arithmetic and logic units, developing problem-solving skills crucial for addressing challenges in pipelined datapaths effectively. Overall, our hybrid teaching approach empowers students with a comprehensive skill set and knowledge base, preparing them to excel in the field of computer architecture.

2.2 The Objectives and Topics Connected to Course Materials with TPK Information

In line with the TPK pedagogical approach, in addition to in-class exercises, practice questions, and homework assignments, we facilitate weekly, well-structured, in-class student-centered discussions. These discussions are thoughtfully designed to align with the TPK framework, which encompasses various aspects of teaching and learning with a focus on the effective integration of technology. Each TPK component comprises a teaching-learning objective, a set of key points, and their interconnected relationships. Table 1 presents examples of objectives linked with their corresponding topics. Furthermore, ongoing assignments are carefully synchronized with learning topics and objectives, as illustrated in this Table 1. Within the TPK pedagogical framework, we diligently identify, organize, and analyze topics and objectives while considering their interdependencies. This strategic approach enables us to determine which topics require review and pre-review in preparation for the course materials.

3 The Objectives and Topics Connected to Course Materials

Adopting the hybrid teaching approach, which integrates Technological Pedagogical Knowledge (TPK) and flipped classroom methods, we extend our educational strategy beyond in-class exercises, practice questions, and homework assignments to include regular, well-organized, student-centered in-class discussions. These discussions take place weekly, focusing on concepts and materials presented in each lecture, encouraging active participation and engagement with the course content. The discussions are tailored to reinforce the understanding of the principles and applications of computer architecture, emphasizing critical analysis and problem-solving.

3.1 Master Arithmetic for Computers

The alignment of ongoing assignments with the learning topics and objectives is meticulously planned to complement the discussions, ensuring that students have the opportunity to apply what they have learned in a practical context. This approach facilitates a deeper understanding of the course material, fostering a dynamic learning environment where students can explore the

Table 1: Course Outline with Objectives, Topics, and Measurements

Objectives	Topics	Measurements
Grasp the Fundamentals of Computer Systems	☐ History of Computing ☐ Computer Abstractions and Technology	☐ Reading Assignments on History and Technology ☐ In-class Discussion on Computer Abstractions
Understand the Language of Instructions	☐ Language of the Computer ☐ Compilers and Java ☐ History of Instruction Sets	☐ Homework on Java and Compiler Design ☐ Class Exercises on Instruction Language
Delve into RISC Instruction-Set Architectures	☐ RISC Principles ☐ Fallacies and Pitfalls	☐ Reading Assignments on RISC History ☐ Practice Problems on RISC Principles
Master Arithmetic for Computers	☐ Subword Parallelism ☐ Arithmetic Operations	☐ In-class Exercises on Arithmetic Operations ☐ Homework on Subword Parallelism
Decode the Basics of Logic Design	☐ Logic Conventions ☐ Simple Implementation to Pipelined Datapath ☐ Verilog for Logic Design	☐ Practice Problems on Logic Conventions ☐ Verilog Design Projects
Explore Processor Design	☐ Pipelining Overview ☐ Hazards and Exceptions in Processor Design ☐ Parallel Processing Concepts	☐ In-class Exercises on Pipelining and Hazards ☐ Discussions on Parallel Processing in Hardware
Navigate Memory Hierarchy	☐ Mapping Control to Hardware ☐ Redundant Arrays of Inexpensive Disks (RAID) ☐ Verilog Cache Controller	☐ Homework on Memory Mapping ☐ Case Study Analysis on RAID Systems
Comprehend Parallel Processing	☐ From Client to Cloud ☐ Network Principles	☐ Network Simulation Assignments ☐ Cloud Computing Project Work

interconnections among key concepts and their practical implications in computer architecture.

In the effort to comprehending CPU performance, students are presented with a range of questions and provided with sample materials to bolster their grasp of this objective. As an illustration, consider the following scenario:

A computer M2 has the following CPIs for instruction types A thru D, and a program P3 has the following mix of instructions (Note: pct = percent).

- M2: Type A CPIA= 1.7;
- Type B CPIB= 2.1;

- Type C CPIC= 2.7;
- Type D CPID= 2.4;
- P3: Type A = 22% ;
- Type B = 29% ;
- Type C = 17% ;
- Type D = remaining %.

Answer the following questions:

(a) Calculate the average CPI of Machine M2.

(b) Calculate the runtime of P3 on M2 if IC = 22,311 and clock rate is 3.3 GHz.

By engaging with such objective inquiries and utilizing sample materials, students gain a more profound understanding of CPU performance within a practical context. Through these real-world scenarios, they acquire the skills to analyze and compute various aspects of CPU utilization and performance. In the provided example, students can apply their knowledge to ascertain crucial metrics like elapsed time, user CPU time, total CPU time, system performance, and CPU performance. This hands-on approach not only reinforces theoretical concepts but also enhances problem-solving abilities, critical thinking skills, and the capacity to apply theoretical knowledge to practical scenarios. Exercises of this nature prove invaluable in bridging the divide between theoretical comprehension and real-world applications, preparing students for the challenges that lie ahead in the realm of computer science and engineering.

3.2 Understand the Language of Instructions

In the pursuit of mastering the MIPS instruction set, students are exposed to a series of exercises and sample questions meticulously crafted to fortify their comprehension of this particular architectural framework. MIPS, being a widely-utilized instruction set architecture in academic contexts, provides a lucid and methodical approach for grasping essential concepts in computer architecture and assembly language programming. By actively engaging with practical examples and problem-solving scenarios related to MIPS, students acquire a tangible understanding of how instructions are formatted, executed, and their interactions with various components of the computer system.

Consider a pipeline with forwarding, hazard detection, and 1 delay slot for branches. The pipeline is the typical 5-stage IF, ID, EX, MEM, WB MIPS design. For the above code, complete the pipeline diagram below (instructions on the left, cycles on top) for the code. Insert the characters IF, ID, EX, MEM, WB for each instruction in the boxes. Assume that there are two levels of bypassing, that the second half of the decode stage performs a read of source registers, and that the first half of the write-back stage writes to the register file, Label all data stalls, and What is the final execution time of the code?

Consider the following assembly language code:

- I0: ADD R4 = R1 + R0;

- I1: SUB R9 = R3 - R4;
- I2: ADD R4 = R5 + R6;
- I3: LDW R2 = MEM[R3 + 100];
- I4: LDW R2 = MEM[R2 + 0];
- I5: STW MEM[R4 + 100] = R2;
- I6: AND R2 = R2 & R1;
- I7: BEQ R9 == R1, Target;
- I8: AND R9 = R9 & R1;

Through this exercise, where students must translate MIPS instructions into C code, they not only reinforce their comprehension of the MIPS architecture but also develop a deeper appreciation for the connection between high-level programming languages and their underlying assembly code counterparts. This practice material vividly exemplifies the intricacies of instruction set design and its profound influence on programming and system performance, enriching students' expertise in this pivotal realm of computer science.

3.3 Explore Processor Design

In our teaching approach, which leverages Technological Pedagogical Knowledge (TPK) and flipped classroom strategies, students delve into the intricacies of datapath and microprocessor design, learning about the sequential flow within a single datapath as conceptualized in the provided diagrams. This exploration breaks down the memory instruction diagram into four key components, starting with the register file and the Arithmetic Logic Unit (ALU), as seen in Figure 1. The register file, equipped with two read ports and one write port, facilitates the retrieval of register contents based on read register inputs and supports edge-triggered write operations to allow read and write actions concurrently within the same clock cycle. Connectivity within this setup is maintained through 5-bit lines for register numbers and 32-bit lines for data. Meanwhile, the ALU, directed by a 4-bit operation signal, executes essential computational and logical operations, featuring a Zero detection output to support branch operations, excluding the overflow output at this initial phase. These components are foundational for performing R-format ALU operations efficiently in computing systems.

Further, the design incorporates two vital units for load and store operations: the data memory unit and the sign extension unit. The data memory unit, a crucial state element, accepts inputs for the address and data to be written, providing a single output for the read result. It is designed with distinct read and write controls, where only one control is active per clock cycle, emphasizing the importance of the read signal to avoid accessing invalid addresses. This unit's design, favoring edge-triggered writes, diverges slightly from conventional memory chips, which typically rely on a write enable signal, yet it can be adapted for compatibility with standard memory chips.

The sign extension unit is designed to adjust data widths, accepting a 16-bit input and extending it to a 32-bit output, ensuring that operations involving varying data sizes are accurately processed in a system predominantly using larger data sizes. Together, the data memory and sign extension

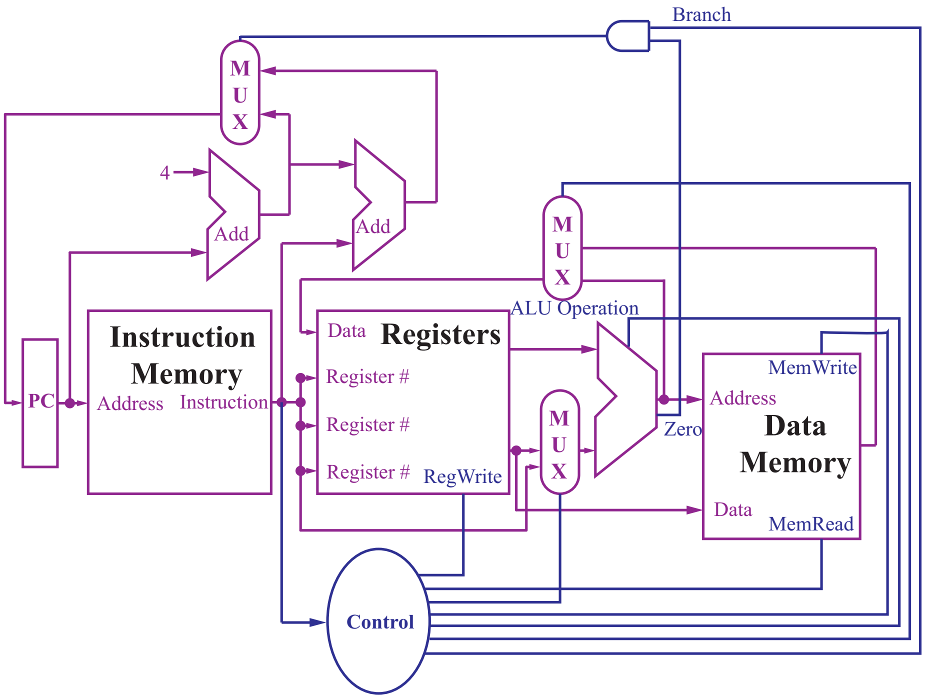

Figure 1: Homework assignment pertaining to the processor and datapath, specifically focusing on the datapath associated with memory instructions and R-type instructions.

units are critical for managing memory operations within a processor, enhancing the register file and ALU's capacity to execute a comprehensive range of instructions. This integrated approach facilitates the seamless execution of instructions, vital for the demands of contemporary computing tasks. In this TPK-focused teaching model, we meticulously select, organize, and examine the topics and objectives, taking into account their interconnectedness. This ensures a thorough preparation and review process for the course materials, enabling an in-depth understanding and application of computer architecture principles.

3.4 Addressing Data Hazards in MIPS Architecture

The instructional seen in Figure 2 provides a vital visual reference for understanding data hazards within the MIPS instruction pipeline, particularly highlighting the phenomenon of pipeline stalls or "bubbles" triggered by load-use hazards. These hazards pose a significant challenge as they interrupt the sequential flow of instruction execution, leading to a detrimental impact on the pipeline's efficiency. A load-use hazard occurs when an instruction eagerly awaits data from a prior load instruction that is still in the process of fetching the data, thereby necessitating the insertion of stalls in the pipeline until the required data is ready for use.

To mitigate such inefficiencies, Figure 2 illustrates two primary methods: instruction reordering and forwarding. Instruction reordering is a preemptive approach aimed at rearranging the code to prevent dependencies that cause hazards. Forwarding, on the other hand, is a dynamic solution that allows the immediate use of data as soon as it is available from a pipeline stage, effectively reducing the need for stalls. While forwarding is an advanced technique that preserves pipeline throughput, it's not always implementable due to specific instruction or hardware limitations.

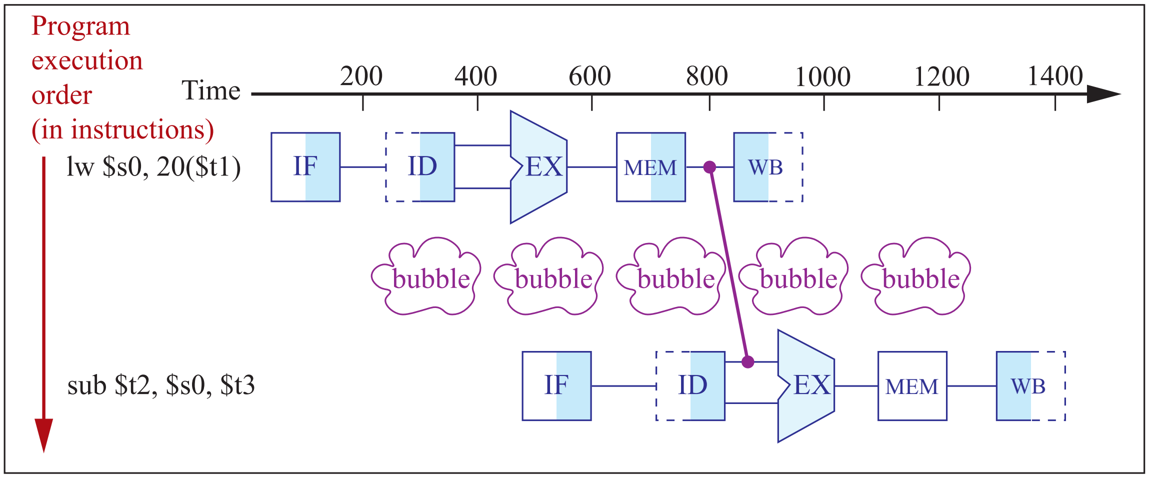

Figure 2: A MIPS pipeline stall due to a load-use hazard, with "bubbles" indicating delays as the CPU awaits data loading before proceeding with the `sub $t2, $s0, $t3` instruction, illustrating the impact of data dependencies on pipeline efficiency. This illustration originated in [26].

In instances where neither reordering nor forwarding is viable, inserting bubbles becomes the alternative. Although this method does resolve the immediate issue of data hazards by delaying subsequent instructions, it is not an ideal solution, as it introduces additional cycles of inactivity that degrade the processor's performance. The necessity to address these hazards is crucial for optimizing CPU pipeline performance, and this is emphasized through hands-on practice and simulation exercises. Such educational strategies not only illuminate the theoretical aspects highlighted in Figure 2 but also foster practical proficiency in resolving pipeline hazards.

4 Concluding Self-Assessment and Fundamental Evaluation of the TPK Method

The evaluation of the effectiveness of single cycle datapath and pipelined datapath in computer architecture curriculum comprised a range of milestone assignments, technical reports, and activities. To gauge students' familiarity with single cycle datapath and pipelined datapath, a comprehensive assessment and interview approach were employed both before and after the multiple practice exercises. Throughout the on-going assignments, feedback and suggestions from students were actively sought after each milestone to gain valuable insights into their experiences and lessons in designing, developing, and building single cycle datapath and pipelined datapath.

These self-assessments, intricately tailored to measure learning outcomes, are closely aligned with ABET standards, playing a crucial role in the overall ABET assessment process. Administered before the semester concludes, these assessments not only serve as a foundation for instructors to enhance teaching methods and fine-tune course designs but also play a pivotal role in validating the quality of learning and academic success within this course. In this particular course, students address six specific questions outlined in Table 2 and Figure 3, with corresponding ABET outcomes referenced in parentheses. This structured approach ensures a thorough examination of students' understanding of single cycle datapath and pipelined datapath.

Table 2: The Questionnaire of Students for Assessment of Education Quality

Questions and Outcome	Survey				
	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
Q1 - (*a*)	64.9%	27.0%	8.1%	0%	0%
Q2 - (*b*)	67.6%	24.3%	8.1%	0%	0%
Q3 - (*c*)	75.7%	18.9%	5.4%	0%	0%
Q4 - (*e*)	81.1%	16.2%	2.7%	0%	0%
Q5 - (*g*)	51.4%	35.1%	8.1%	5.4%	0%
Q6 - (*g*)	35.1%	56.8%	5.4%	2.7%	0%

- Question 1 - “I can understand and use knowledge of mathematics including advanced topics such as differential and integral calculus, linear algebra, discrete math, and differential equations in single cycle datapath and pipelined datapath.” (Outcome (*a*): An ability to apply knowledge of mathematics, science, and engineering principles to electrical engineering, *i.e.* Knowledge of mathematics encompasses advanced topics typically including differential and integral calculus, linear algebra, complex variables, discrete math, and differential equations.)
- Question 2 - “I can apply formal engineering design methodology to perform the design, experiments and construction of the single cycle datapath and pipelined datapath through multiple practice exercises based on experimental test data and interpretation, as well as to analyze and interpret data relating to single cycle datapath and pipelined datapath that resolve electrical system problems” (Outcome (*b*): An ability to design and conduct experiments, as well as to analyze and interpret data relating to electrical systems.)
- Question 3 - “I can understand and design basic single cycle datapath and pipelined datapath with assigned a sequence of in-class exercises and on-going practical questions and work to meet the final goals.” (Outcome (*c*): The capability to recognize, define, and effectively address challenges in the field of electrical engineering.)
- Question 4 - “I can understand structures and models of single cycle datapath and pipelined datapath and profoundly understand some important design methodologies of single cycle datapath and pipelined datapath models, and can also identify, formulate, and solve the issues raised in assigned in-class exercises and on-going practical questions of single cycle datapath and pipelined datapath” (Outcome (*e*): An ability to identify, formulate, and solve electrical engineering problems).
- Question 5 - “I have effective communication skills in the context of a collaborative, multi-disciplinary design activity in the study of single cycle datapath and pipelined datapath”. (Outcome (*g*): An ability to communicate effectively).
- Question 6 - “I can create professional documentation in connection with the in-class exercises and on-going design questions on single cycle datapath and pipelined datapath”. (Outcome (*g*): An ability to communicate effectively).

The findings from the assessment questionnaire are presented in Table 2 and visually illustrated in Figure 3. Upon reviewing the results in Table 2, it becomes apparent that a majority of students

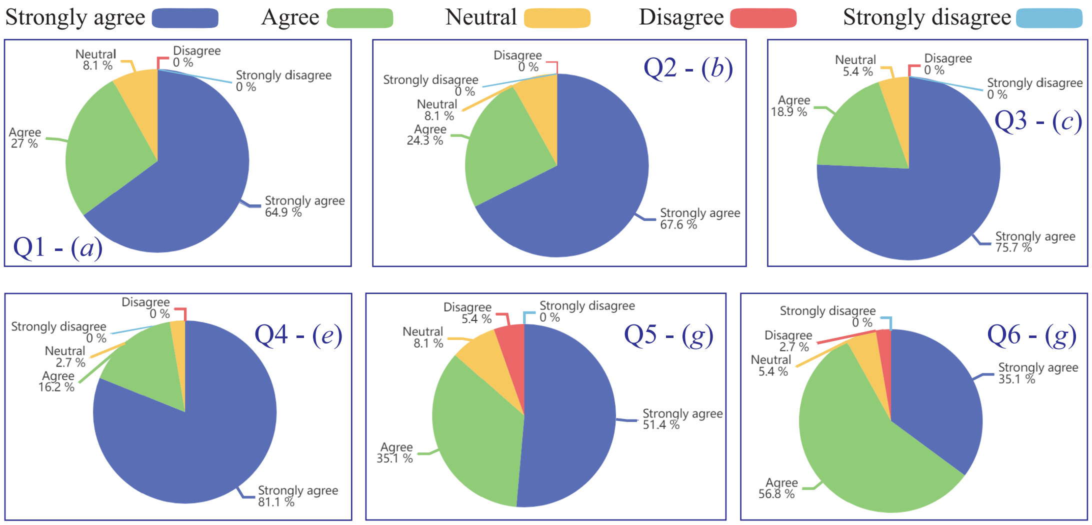

Figure 3: The illustration of the assessment of learning quality results.

overwhelmingly express “strong agreement” with statements corresponding to ABET outcomes (*a*), (*b*), (*c*), and (*e*). The percentages of “strong agreement” for outcomes (*b*), (*c*), and (*e*) in this course surpass those evidently observed in the instructor’s previous experiences teaching other courses. This suggests a potential beneficial impact of the applied teaching methods on these specific results. Regarding communication skills, particularly in writing and oral presentation aligned with ABET outcome (*g*), a notable finding is the presence of “disagreement”. Some graduate students express unease with their abilities in oral and written presentations, with 5.4% and 2.7% registering disagreement, respectively, in the survey. This occurs despite the training provided in this class and other relevant curricula.

There has been a remarkable increase in “strongly agree” responses, particularly for Question 4, reaching an impressive 83.3%, surpassing the usual rate. Question 4 is closely linked to the ability to identify, formulate, and solve electrical engineering problems, as illustrated in Figure 3. Furthermore, when compared to the instructor’s previous experience teaching the course using a traditional problem-based method and a Technological Pedagogical Knowledge (TPK) based method associated with a flipped learning (FL) approach (TPK-FL), along with in-class exercises and ongoing practical sessions for reflection and adjustments, the current course exhibits higher percentages for both “strongly agree” (81.1%) and “agree” (16.2%). This suggests that the integrated approach of MPEM pedagogy and the TPK-FL mechanism in this course is significantly more effective than previous instructional models.

Concerning Questions 2 and 3, notable percentages of “*strongly agree*” responses stand out at 67.6% and 75.7%, aligning with Outcomes (*b*) and (*c*), respectively. These outcomes emphasize the ability to design and conduct experiments, as well as to analyze and interpret data related to electrical systems (Outcome (*b*)). Additionally, Outcome (*c*), which pertains to the ability to design electrical systems, components, or processes to meet desired needs, similarly receives high “*strongly agree*” responses. There are “*neutral*” percentages of 8.1% and 5.4% in Questions 2 and 3, corresponding to Outcomes (*b*) and (*c*), respectively. Further analysis of interview and survey comments reveals that some students attribute this to a perceived lack of sufficient background in programming and hardware design.

In contrast to Questions 2 and 3, the percentages of "strongly agree" and "agree" for Question 1 stand at 64.9% and 27.0%, respectively, aligning with ABET Outcome *a*. These responses are attributed to the heightened effectiveness of datapath design methodologies. This methodology involves the application of mathematical knowledge, including advanced topics such as differential and integral calculus, linear algebra, discrete math, and differential equations in single-cycle datapath and pipelined datapath. Not only does this contribute to a more resilient learning experience, but it also underscores the nuanced effectiveness of these methodologies in the given context. Additionally, there is a "neutral" percentage of 8.1% in Question 1, corresponding to Outcome (*a*).

Evaluated through Question 5, assesses communication capabilities during in-class discussions and teamwork targeted to Outcome *g*. In this context, 5.4% of students express "disagreement", while 8.1% express "neutral", aligning with Outcome (*g*). The necessity for high-quality written documentation is a crucial aspect of this ongoing practical design class. However, students often encounter challenges in adapting to technical writing, resulting in a 2.7% "disagreement" rate and a 5.4% "neutral" rate in Question 6, linked to Outcome (*g*). This assessment is crucial for the development of professional documentation related to ongoing assignments and the design assignments focusing on datapath modules. Considering the inherent challenges in oral and technical writing communication skills within in-class discussions and ongoing practical assignments, 51.4% and 35.1% of students express "strong agreement" for Questions 5 and 6, respectively, both tied to Outcome (*g*).

Analyzing students' performance across multiple dimensions, including homework assignments, Q&A sessions, exams, self-assessment surveys, and feedback from the official university course evaluation, alongside a comparative assessment against the instructor's previous teaching experiences, highlights the clear effectiveness of the hybrid TPK-FL and MPEM pedagogy. This approach has unequivocally played a role in cultivating a deep understanding of computer architecture design.

5 Conclusion

This paper introduces and implements a hybrid pedagogy, combining Technological Pedagogical Knowledge (TPK) with a flipped learning (FL) method and incorporating multiple practice exercises (MPEM) to enhance learning opportunities. The TPK-based approach explores the intricate correlations between technological tools and specific pedagogical practices, while the MPEM method focuses on constructing TPK for senior students in the context of designing single-cycle datapath and pipelined datapath in a Computer Architecture course. The carefully planned series of multiple practice exercises covers interconnected topics in datapath design, facilitating timely and comprehensive learning. The flipped learning method disrupts the traditional teaching model, shifting the delivery of instructional content and utilizing class time for interactive discussions. By assigning course materials on single-cycle datapath and pipelined datapath beforehand, students engage in active learning during classroom sessions.

This innovative pedagogical approach leverages technology and active learning strategies to empower students to construct their knowledge of technological tools autonomously. It not only promotes a deeper understanding of the course materials but also encourages students to be more self-directed in their learning journey.

Results from students' self-assessment questionnaires compellingly demonstrate that the hybrid TPK-FL and MPEM pedagogy effectively accomplishes the intended learning outcomes for this advanced robotics course. The approach significantly enhances students' self-motivation, reflection, performance, exploration, and understanding of artificial intelligence and robotics techniques. As a result, the hybrid TPK-FL and MPEM pedagogy prove to be effective tools in fostering a profound comprehension of computer architecture design among students.

References

[1] M. Menekse, G. S. Stump, S. Krause, and M. T. Chi, "Differentiated overt learning activities for effective instruction in engineering classrooms," *Journal of Engineering Education*, vol. 102, no. 3, pp. 346–374, 2013.

[2] B. Kollöffel and T. De Jong, "Conceptual understanding of electrical circuits in secondary vocational engineering education: Combining traditional instruction with inquiry learning in a virtual lab," *Journal of Engineering Education*, vol. 102, no. 3, pp. 375–393, 2013.

[3] L. Cheng, A. D. Ritzhaupt, and P. Antonenko, "Effects of the flipped classroom instructional strategy on students' learning outcomes: A meta-analysis," *Educational Technology Research and Development*, vol. 67, pp. 793–824, 2019.

[4] W. Zhao, X. Luo, C. Luo, and Y. Peng, "Design and implementation of project-based courses on cutting-edge computer technologies," in *2017 ASEE Annual Conference & Exposition*, 2017.

[5] K. Zervoudakis, K. Mastrothanasis, and S. Tsafarakis, "Forming automatic groups of learners using particle swarm optimization for applications of differentiated instruction," *Computer Applications in Engineering Education*, vol. 28, no. 2, pp. 282–292, 2020.

[6] C. Luo, Z. Bi, and W. Zhao, "Function approximation through an efficient neural networks method," in *2019 ASEE Annual Conference & Exposition*, 2019.

[7] V. Leite, "Innovative learning in engineering education: Experimenting with short-term project-oriented research and project-based learning," in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, 2017, pp. 1555–1560.

[8] N. Hosseinzadeh and M. R. Hesamzadeh, "Application of project-based learning (PBL) to the teaching of electrical power systems engineering," *IEEE Transactions on Education*, vol. 55, no. 4, pp. 495–501, 2012.

[9] J. Rodríguez, A. Laveron-Simavilla, J. M. del Cura, J. M. Ezquerro, V. Lapuerta, and M. Cordero-Gracia, "Project based learning experiences in the space engineering education at Technical University of Madrid," *Advances in Space Research*, vol. 56, no. 7, pp. 1319–1330, 2015.

[10] E. Marasco and L. Behjat, "Integrating creativity into elementary electrical engineering education using CDIO and project-based learning," in *2013 IEEE International Conference on Microelectronic Systems Education (MSE)*, 2013, pp. 44–47.

[11] L. Huijuan, Z. Zhenyang, and S. Tengfei, "Case study of enquiry-based learning designed for rotating magnetic fields in electric machinery course," *International Journal of Electrical Engineering Education*, vol. 54, no. 4, pp. 341–353, 2017.

[12] L. Yuliati, C. Riantoni, and N. Mufti, "Problem solving skills on direct current electricity through inquiry-based learning with phet simulations." *International Journal of Instruction*, vol. 11, no. 4, pp. 123–138, 2018.

[13] S. I. Haryudo, A. I. Agung, R. Firmansyah, M. Widyartono, and W. Aribowo, “Development of learning tool control with inquiry based learning to improve learning motivation of students of electrical engineering,” *in International Conference on Indonesian Technical Vocational Education and Association (APTEKINDO)*, pp. 110–113, 2018.

[14] T. Eppes, I. Milanovic, and K. Wright, “Improving student readiness for inquiry-based learning: An engineering case study,” *International Association of Online Engineering*, 2020.

[15] J. Martín-Gutiérrez, P. Fabiani, W. Benesova, M. D. Meneses, and C. E. Mora, “Augmented reality to promote collaborative and autonomous learning in higher education,” *Computers in Human Behavior*, vol. 51, pp. 752–761, 2015.

[16] C. Luo, “An integrated learning approach used in computer architecture,” in *2019 ASEE Annual Conference & Exposition*, 2019.

[17] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan, “Cooperative inverse reinforcement learning,” *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[18] C. Luo, J. Wang, L. Wang, and W. Zhao, “Multi-lab-driven learning method used for robotics ROS system development,” in *2017 ASEE Annual Conference & Exposition*, 2017.

[19] E. A. Van Vliet, J. C. Winnips, and N. Brouwer, “Flipped-class pedagogy enhances student metacognition and collaborative-learning strategies in higher education but effect does not persist,” *CBE—Life Sciences Education*, vol. 14, no. 3, p. ar26, 2015.

[20] S. J. Dickerson and R. M. Clark, “Implementation of lessons learned to simulation-based reflection in a digital circuits course.” *Computers in Education Journal*, vol. 13, 2022.

[21] J. Wang, C. Luo, W. Zhao, and X. Li, “Empowering students with self-regulation in a project-based embedded systems course,” in *2017 ASEE Annual Conference & Exposition*, 2017.

[22] M. A. Burns, V. N. Johnson, K. Grasman, S. Habibi, K. A. Smith, A. Kaehr, M. F. Lacar, and B. Yam, “Pedagogically grounded techniques and technologies for enhancing student learning.” *Advances in Engineering Education*, vol. 11, no. 3, 2023.

[23] W. Zhao, X. Liu, C. Luo, and X. Luo, “Enhancing communication with students with a teaching method based on topical guide objectives,” in *2017 ASEE Annual Conference & Exposition*, 2017.

[24] C. Luo, X. Li, J. Wang, and W. Zhao, “Enhancement of electrical engineering education by a mentoring scheme,” in *2015 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 2015, pp. 72–76.

[25] C. Luo, “Enhancing the quality of student research by an on-going multiple-project-based course,” in *2015 ASEE Annual Conference & Exposition*, 2015, pp. 26–669.

[26] D. A. Patterson and J. L. Hennessy, *Computer Organization and Design ARM Edition: The Hardware Software Interface*, 2016.