International Journal of Research (IJR) Vol-1, Issue-9, October 2014 **ISSN 2348-6848**

Integrating Research and E-Learning in Advanced Computer Architecture Courses

Rahul Rapariya[1] Surbhi Miglani[2], Sakshi Kohli[3]
rahulrapariya@gmail.com, miglani.surbhi0@gmail.com, kohlisakshi92@gmail.com

ABSTRACT

This paper presents novel methods in teaching advanced computer architecture courses. These methods include presenting fundamental computer architecture issues using e-learning; employing visual aids to teach fundamentals concepts like Caches, pipelining and scheduling. In addition, this paper discuss integrating research into the course is beneficial to the students pursing a PhD career.

1. INTRODUCTION

Advanced Computer Architecture courses are usually taught in the senior year of undergraduate curriculum or first year graduate curriculum. This is a fundamental course for microprocessor designers and computer architectures. Hence, establishing a good understanding for this course is a must for graduating engineer.

The concepts taught in this courses includes: measuring performance, Instruction Set Design, Memory Hierarchy and Caches, Pipelining and its Hazards, Instruction Level Parallelism, I/O storage, and latest contemporary computer architecture issues. The course usually combines the software and hardware approaches that increase performance of the microprocessor design. The course introduces these concepts and presents quantitative approaches to measure the feasibility of these approaches on performance emphasizing on the differences between hardware and software approaches.

There are several computer architecture books [1], [2], and [3] available to the Computer Architecture Instructor. However, only reference [1] gives a quantitative approach to computer architecture concepts. Also, Hennessy & Patterson's gives a comprehensive documentation on most of computer architecture topics.

This paper will first discuss some computer architecture concept set for e-learning like Cache Associativity, superscalar microprocessors, and dynamic scheduling algorithms. Then discuss integrating research topics in the course will be presented.

2. CACHE SET ASSOCIATIVTY

This section proposes changing the presentation of the set associativity concept. The concept of cache set associativity is presented in [1] could be presented in a better way. This paper simplifies the presentation of associativity concept to make it better for students to visualize. Figure 1 & 2 show the set associativity explained according to [1]. This approach presents the cache to be split into number of sets and each set has equal number of lines. For example, a 2 -way set associative cache having 8 lines will have 4 sets and each set has two lines.

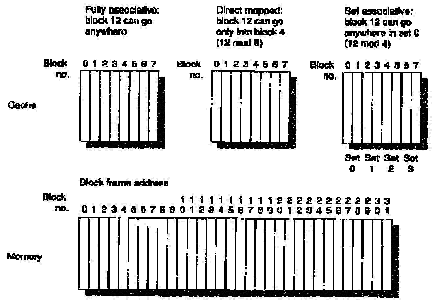


Fig. 1. Hennessy & Patterson's Memory Diagram [1].

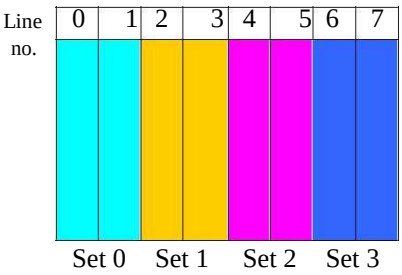


Fig. 2. Hennessy & Patterson's 2-way set associative cache.

This paper proposes set associativity to be dividing the cache into n sets; each set has number of lines. For example, the same 2-way set associative cache with 8 lines will be divided into two sets and each set has four lines. Figure 3 shows a 2-way associative

cache with four lines in each set. The hardware will be the same whether the concepts is explained according to [1] or this approach, but this novel way is better to visualize the concept of associaitivity.

3. COMPLICATED CONCEPTS MADE EASY USING VISUAL AIDS

Computers and microprocessors are rich with concepts that seem complicated for new students. These concepts are easily explained with visual aids. Concepts like Pipelining and its hazards, Superscalar design, Instruction Level Parallelism, and Dynamic Scheduling. This section shows these concepts explained using visual aids like PowerPoint animations.

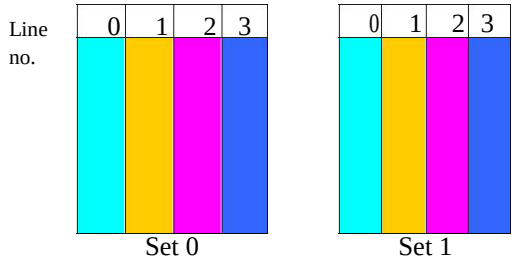
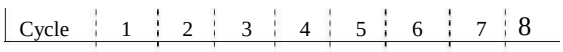


Fig. 3. Novel approach for explaining the concept of cache associativity.

3.1. Pipelining and Hazards

The concept of pipelining is simple to understand, but pipelining hazards can get complicated. Figure 4 shows a slide of a PowerPoint presentation showing each pipeline stage with respect to its instruction. This if for a DLX processor of five pipeline stages (Instruction Fetch, Instruction Decode, Instruction Execute, Instruction Memory write, and Instruction Write Back). In PowerPoint presentations, each instruction will appear at a given time such as shown in Figure 5. Figure 6 will be display the next instruction to be issued in the pipeline stage and so forth.

The pipeline could be also shown in terms of cycles, meaning display the events at each clock cycle as shown in Figure 5 for instruction issued.



For pipeline hazards, the visual aid could show bubbles inserted in the pipeline. Figure 7 shows bubbles and data forwarded using arrows.

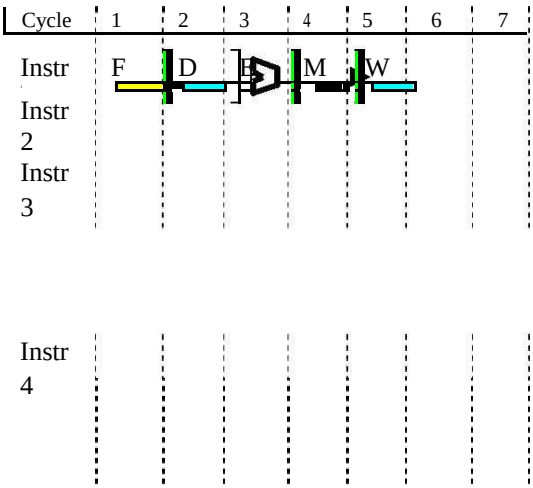


Fig. 5. DLX pipelining the first stage.

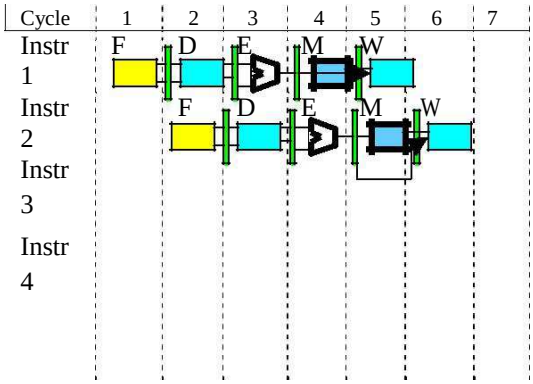
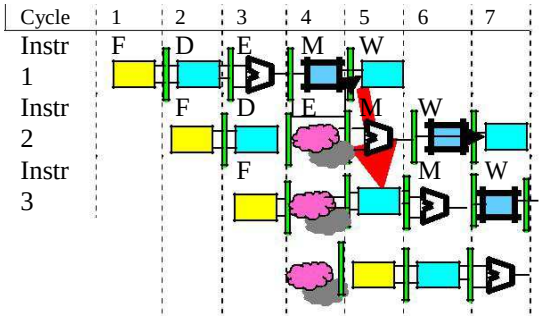
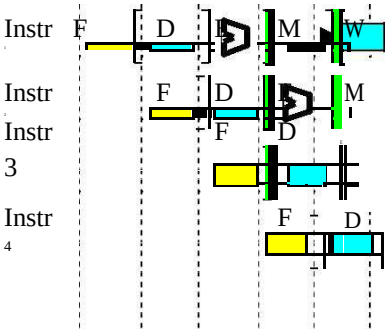
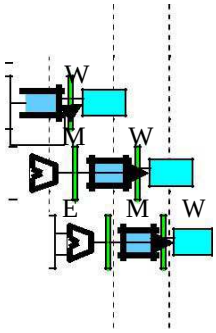


Fig. 6. DLX pipelining the second instruction.




Instr
F
D
E
M
W
Instr
F
D
E
M
Instr
3
F
D
Instr
4
F
D


W
M
W
E
M
W

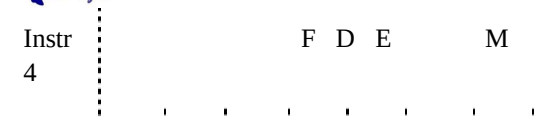


Fig. 7. DLX pipeline stage.

3.2. Superscalar and multi-issue machines

The concept of superscalars can also be explained with the visual aids. For example, Figure 8 shows a Fig. 4. DLX pipeline stage.
2-way issues for a DLX supersacalar machine where one pipeline is assigned for integer and the other for floating-point operations. Note that floating -point operation takes 3 cycles to execute. Again this could be presented to the students as a motion animation where two instructions are issued at a given time.

Instruction type	**Pipeline Stages**						
Integer	F	D	E	M			
FP	F	D	E	E	E	M	
Integer		F	D	E	M		
FP		F	D	E	E	E	M
Integer			F	D	E	M	
FP			F	D	E	E	E

Fig. 8. DLX Superscalar issue of Integer and FP pipeline.

3.2. Instruction Level Parallelism

Instruction Level Parallelism and Dynamic Scheduling is made easy with visual aids. For example, Tomasula's algorithm for dynamic scheduling can be easily understood using animations. Figure 9 shows the window of Dynamic scheduling at cycle =0 for five instructions scheduled on this window.

Instruction	Issue	Exec	Write Result
i			
i+1			
i+2			
i+3			
i+4			

Memory	Busy
Load1	
Load2	
Load3	

Reservation Station

Time	Name	Busy	Op	Vj	Vk	Qj	Qk
	Add1						
	Add2						
	Add3						
	Mul1						
	Mul2						

RF	F0	F2	F4	F6	…….		F30

Fig. 9. Tomasula's algorithm

The engineering student can then fill in the blank in each cycle and the given time for it. For example the student can be writing each cycle result as the lecture taking place. Note that these examples are obtained mostly form [1]. However, the idea to involve the student in the process of learning and solving the problem as it been presented for the very first time is novel.

4. INTEGRATING RESEARCH TOPICS

Advanced Computer Architecture is rich with new topics that are in the research stage. The student must be aware of these topics before completing any advanced computer architecture course. This could be

integrated in the course project where the students are asked to provide a quantitative measure for these new topics, or it could be in the form of exercise and small end of the course homework. The effect of this research on there understanding is tremendous.

The best types of research papers to provide for the students are papers that provide the original concept. Also, paper that compares different microprocessors and their implementations [4] [5] would be good challenge to the students. However, for advanced courses and computer architects the best approach is to challenge the students with advanced topics [6]. This will give more bases for creativity when pursing their engineering career.

5. CONCLUDING REMARKS

Advanced Computer Architecture is rich with advanced topics. Some of the universities nowadays offer two levels of Computer Architecture courses for graduate level engineers. This paper offers a better ways to present some of the typical concepts of computer architecture such as pipelining, pipelining hazards, cache associativity, and Instruction Level Parallelism, and Dynamic Scheduling. There are more creative ways to present computer architecture concepts that are not presented here.

The most advanced way of learning is through visual aids and e-learning. Future trends in teaching Computer Architecture may lead to e-learning at a distance. This could be explored in future papers.

6. REFERENCES

[1] J.L. Hennessy & D.A. Patterson, "Computer Architecture A Quantitative Approach 3rd edition," Morgan Kaufmann Publishers, USA, 2003.

[2] Barry Wilkinson, "Computer Architecture Design and Performance 2nd edition," Prentice Hall Europe, 1996.

[3] M. Morris Mano, "Computer System Architecture 3rd edition," Prentice Hall, Inc. USA, 1993.

[4] Bruce Jacob and Trevor Mudge, "Virtual Memory in Contemporary Microprocessors," IEEE Micro, vol. 18, no 4, pp. 60-75, July 1998.

[5] M. Mittal, A. Peleg and U. Weiser "MMX technology Architecture Overview," Intel Technology Journal, pp. 1-12, Q3, 1997.

[6] Heald et. all, "Implementation of the third generation SPARC V9 64-b microprocessor," ISSCC Digest of Technical Papers, pp 412-413, 2000.