The Computer Architecture Sequence at Michigan State University

Richard J. Enbody
Department of Computer Science and Engineering
Michigan State University, East Lansing, MI 48824-1226
enbody@cse.msu.edu

***Abstract*— This paper outlines the computer organization and architecture courses in the Department of Computer Science and Engineering at MSU.**

I. Introduction

Computer Architecture has been a key component of the MSU Computer Science degree since it was established in 1967. The four founding members of the department were Electrical Engineers, some of whom came to build MSU's first computer (MISTIC) in 1956. Two are still on the faculty and a third is a Professor Emeritus. The influence of those fledgling computer builders remains.

Currently the cornerstone of our undergraduate architecture sequence is a simulator developed by Dr. Reid, one of those who built the MISTIC computer almost forty years ago. The emphasis of our computer architecture course has been the construction of a simulated processor capable of running a simple operating system and executing programs. That is, the students build something, and it has to work.

Michigan State University uses the semester system with fifteen weeks of classes per semester.

II. Undergraduate Architecture Sequence

A. *CPS 320 Computer Organization and Assembly Language Programming*

This course is a required four-credit course with three hours of lecture and two hours of laboratory per week. Prerequisites are an Introduction to Computer Programming (CS1) course using C++ and a course in Discrete Mathematics. The current text is *Architecture, Organization and Programming* (Maccabe; Irwin, 1993).

This course develops an understanding of: the organization of digital computing systems, the representation of and operations on basic data types, the process of translating and executing a computer program.

The primary vehicles to achieve these objectives are the study of general concepts in this area, and the study of a specific computing system which illustrates these concepts. Students use a simulator to construct combinational and sequential circuits. They also write assembly language and C programs in a UNIX environment.

B. *CPS 420 Computer Architecture*

This course is a required four-credit course with three hours of lecture and two hours of laboratory per week. The prerequisites are Data Structures and Algorithms (CS2), Computer Organization, and Theory of Computing. The textbook is *Computer Organization and Design* by Patterson & Hennessy (Morgan Kaufmann).

In this course the student use simulation to design and implement a central processing unit for an up-to-date architecture. Simulation components ranging from gates to complex devices such as multiplexers, arithmetic-logic units, and read-only memories are explored. The design project includes the following:

development of the instruction set, implementing the datapath and datapath control, memory interface, input-output interface, development of modules representing subsystems, testing of individual modules.

The students learn to understand issues related to timing in digital systems such as pipelining, data hazards, and branch hazards; learn algorithms for doing arithmetic in digital processors; and learn to design and analyze the effects of various choices of number of levels of cache, main memory, and other secondary memory devices. The students are able to analyze basic performance characteristics and compare different processors based on clock speed, number of cycles per instruction, and instruction mix. Finally, students come to understand design issues related to input and output, including the use of memory-mapped input-output, buses, direct memory access, and programmed input-output.

C. *CPS 474 High-Performance Computing*

This course is an optional three-credit course offered once a year. It has two hours of lecture and two hours of laboratory per week. The prerequisites are Computer Architecture and Linear Algebra. The text is *High Performance Computing* by Severance and Dowd (O'Reilly and Associates).

This course focuses on how to program high performance computing systems. The course views computer architecture from a performance perspective. Students learn how to develop their own benchmarks to measure the performance of the various components of a computer. Students study how the central processor, memory hierarchy, floating point numbers, compilers and parallel processing architecture impact their programs. The course examines single processor and multiple processor systems from an architecture and programming perspective.

D. Experience With Our Undergraduate Courses

The Department has been quite satisfied with the architecture component of the curriculum. In particular, the large project to develop a simulated CPU in CPS 420 plays an important role in the curriculum well beyond its immediate value as a tool for understanding computer architecture. The project is developed in a series of modules over the course of the semester and, when taken as a whole, it is a very large project. For most students it is the first time they have worked on a computer project which spans fifteen weeks of work.

However, the architecture project provides another benefit students must use modules they construct. The stages of the project involve construction of the components of the CPU to be simulated. The components must interact with each other and some use components designed by the student earlier in the course. Eventually the components are combined into a working computer. The complexity of the components make them almost impossible to exhaustively test so students frequently find themselves having to fix modules which they thought worked correctly. The process of "eating one's mistakes" is a valuable learning experience. The concepts of correct design and thorough testing are reinforced. An interesting characteristic of the course is that the a clever student can be beaten by a meticulous student.

III. Graduate Architecture Courses

A. CPS 820 Advanced Computer Architecture

This course is a three-credit course with three hours of lecture per week, and Computer Architecture as a prerequisite. More than half the lectures are devoted to supplementary topics, but otherwise it covers the first six chapters of *Computer Architecture: A Quantitative Approach, Second Edition* by Hennessy & Patterson. Recent supplementary topics include benchmarking, SPEC95, performance monitoring (PERFMON), DRAM technology, disk technology, EPIC & IA-64, Alpha 21264, P6 "backside" bus, PCI "frontside" bus, and asynchronous processor design.

This semester's emphasis was on the analysis of computer architectures. We worked with software tools which allow access to the set of special on-chip counting registers on UltraSparc and PentiumPro microprocessors. The availability of this set of tools allowed us to investigate properties previously inaccessible outside the major chip producers. We also purchased a copy of the SPEC95 benchmark suite for our use.

B. CPS 822 Parallel Processing Computer Systems

This is a three-credit course with three hours of lecture per week, and Advanced Computer Architecture as a prerequisite.

The objective of this course is to consider a number of topics pertaining to the architecture of parallel computing systems. From the prerequisite CPS 820 course, students know that computer architecture is concerned with organization, software, and algorithms. This course covers just two sections of the Henessy and Patterson text in order to be certain that students have a common ground and understand the basic terminology. Students read a number of papers that cover a spectrum of topics. Several videos of presentations by industry leaders on certain aspects of parallel computing are also be studied. In addition, students are responsible for submitting short critical reviews of the assigned papers and being part of a project group.

C. CPS 920 Topics in Computer Architecture

A three-credit topics course is offered annually. The choice of topic and emphasis of the course is up to the instructor. The last offering covered the latest developments in microprocessor architecture, i.e. the significant developments since the latest texts and beyond what was covered in the Advanced Computer Architecture course.

D. Experience With Our Graduate Courses

Our CPS 820 Advanced Computer Architecture course is one of the courses in greatest demand in the graduate program. Most graduate students take the course, and students in Electrical Engineering who have an interest in Computer Engineering also take the course. It varies in its emphasis depending on who teaches it, but has tended to have some experimental component to it.

The Parallel Architecture course has often had programming as a component. The belief is that some amount of programming is needed to experience the unique characteristics of parallel architectures. Programming is not emphasized in this course since there is a sister course: CPS 838 Parallel Algorithms.

IV. Related Courses

In addition to these architecture courses there are a number of related systems courses. Examples include operating systems and networks course at both the undergraduate and graduate level. Courses such as hardware arithmetic, VLSI design, and implementation of neural networks are available in our Department of Electrical and Computer Engineering.

V. Conclusion

Computer Architecture has been an important component of the computer science program since its inception thirty years ago. We feel that it is a core component to our identity as a program.