Appendix A

PROJECTS FOR TEACHING COMPUTER ORGANIZATION AND ARCHITECTURE

A.1 INTERACTIVE SIMULATIONS....2
A.2 RESEARCH PROJECTS....3
A.3 SIMULATION PROJECTS....4
SimpleScalar....4
SMPCache....5
A.4 ASSEMBLY LANGUAGE PROJECTS....6
A.5 READING/REPORT ASSIGNMENTS....7
A.6 WRITING ASSIGNMENTS....7
A.7 TEST BANK....8

Many instructors believe that research or implementation projects are crucial to the clear understanding of the concepts of computer organization and architecture. Without projects, it may be difficult for students to grasp some of the basic concepts and interactions among components. Projects reinforce the concepts introduced in the book, give students a greater appreciation of the inner workings of processors and computer systems, and can motivate students and give them confidence that they have mastered the material.

In this text, I have tried to present the concepts of computer organization and architecture as clearly as possible and have provided numerous homework problems to reinforce those concepts. Many instructors will wish to supplement this material with projects. This appendix provides some guidance in that regard and describes support material available in the **Instructor's Resource Center (IRC)** for this book, accessible by instructors online from Pearson. The support material covers six types of projects and other student exercises:

- Interactive simulations
- Research projects
- Simulation projects
- Assembly language projects
- Reading/report assignments
- Writing assignments
- Test bank

A.1 INTERACTIVE SIMULATIONS

Interactive simulations provide a powerful tool for understanding the complex design features of a modern computer system. Today's students want to be able to visualize the various complex computer systems

mechanisms on their own computer screen. A total of 20 simulations are used to illustrate key functions and algorithms in computer organization and architecture design. Table A.1 lists the simulations by chapter. At the relevant point in the book, an icon indicates that a relevant interactive simulation is available online for student use

Because the simulations enable the user to set initial conditions, they can serve as the basis for student assignments. The IRC includes a set of assignments, one set for each of the interactive simulations. Each assignment includes a several specific problems that can be assigned to students.

The interactive simulations were developed under the direction of Professor Israel Koren, at the University of Massachusetts Department of Electrical and Computer Engineering. Aswin Sreedhar of the University of Massachusetts developed the interactive simulation assignments.

For access to the animations, click on the rotating globe at this book's web site at http://williamstallings.com/ComputerOrganization.

A.2 RESEARCH PROJECTS

An effective way of reinforcing basic concepts from the course and for teaching students research skills is to assign a research project. Such a project could involve a literature search as well as a Web search of vendor products, research lab activities, and standardization efforts. Projects could be assigned to teams or, for smaller projects, to individuals. In any case, it is best to require some sort of project proposal early in the term, giving the instructor time to evaluate the proposal for appropriate topic and appropriate level of effort. Student handouts for research projects should include:

- A format for the proposal
- A format for the final report
- A schedule with intermediate and final deadlines
- A list of possible project topics

The students can select one of the listed topics or devise their own comparable project. The IRC includes a suggested format for the proposal and final report as well as a list of possible research topics.

A.3 SIMULATION PROJECTS

An excellent way to obtain a grasp of the internal operation of a processor and to study and appreciate some of the design trade-offs and performance implications is by simulating key elements of the processor. Two useful tools that are useful for this purpose are SimpleScalar and SMPCache.

Compared with actual hardware implementation, simulation provides two advantages for both research and educational use:

- With simulation, it is easy to modify various elements of an organization, to vary the performance characteristics of various components, and then to analyze the effects of such modifications.
- Simulation provides for detailed performance statistics collection, which can be used to understand performance trade-offs.

SimpleScalar

SimpleScalar [BURG97, MANJ01a, MANJ01b] is a set of tools that can be used to simulate real programs on a range of modern processors and systems. The tool set includes compiler, assembler, linker, and simulation and visualization tools. SimpleScalar provides processor simulators that

range from an extremely fast functional simulator to a detailed out-of-order issue, superscalar processor simulator that supports nonblocking caches and speculative execution. The instruction set architecture and organizational parameters may be modified to create a variety of experiments.

The IRC for this book includes a concise introduction to SimpleScalar for students, with instructions on how to load and get started with SimpleScalar. The manual also includes some suggested project assignments.

SimpleScalar is a portable software package the runs on most UNIX platforms. The SimpleScalar software can be downloaded from the SimpleScalar Web site. It is available at no cost for noncommercial use.

SMPCache

SMPCache is a trace-driven simulator for the analysis and teaching of cache memory systems on symmetric multiprocessors [RODR01]. The simulation is based on a model built according to the architectural basic principles of these systems. The simulator has a full graphic and friendly interface. Some of the parameters that they can be studied with the simulator are: program locality; influence of the number of processors, cache coherence protocols, schemes for bus arbitration, mapping, replacement policies, cache size (blocks in cache), number of cache sets (for set associative caches), number of words by block (memory block size).

The IRC for this book includes a concise introduction to SMPCache for students, with instructions on how to load and get started with SMPCache. The manual also includes some suggested project assignments.

SMPCache is a portable software package the runs on PC systems with Windows. The SMPCache software can be downloaded from the SMPCache Web site. It is available at no cost for noncommercial use.

A.4 ASSEMBLY LANGUAGE PROJECTS

Assembly language programming is often used to teach students low-level hardware components and computer architecture basics. CodeBlue is a simplified assembly language program developed at the U.S. Air Force Academy. The goal of the work was to develop and teach assembly language concepts using a visual simulator that students can learn in a single class. The developers also wanted students to find the language motivational and fun to use. The CodeBlue language is much simpler than most simplified architecture instruction sets such as the SC123. Still it allows students to develop interesting assembly level programs that compete in tournaments, similar to the far more complex SPIMbot simulator. Most important, through CodeBlue programming, students learn fundamental computer architecture concepts such as instructions and data co-residence in memory, control structure implementation, and addressing modes.

To provide a basis for projects, the developers have built a visual development environment that allows students to create a program, see its representation in memory, step through the program's execution, and simulate a battle of competing programs in a visual memory environment.

Projects can be built around the concept of a Core War tournament. Core War is a programming game introduced to the public in the early 1980s, which was popular for a period of 15 years or so. Core War has four main components: a memory array of 8000 addresses, a simplified assembly language Redcode, an executive program called MARS (an acronym for Memory Array Redcode Simulator) and the set of contending battle programs. Two battle programs are entered into the memory array at randomly chosen positions; neither program knows where the other one is. MARS executes the programs in a simple version of time-sharing. The two programs take turns: a single instruction of the first program is executed,

then a single instruction of the second, and so on. What a battle program does during the execution cycles allotted to it is entirely up to the programmer. The aim is to destroy the other program by ruining its instructions. The CodeBlue environment substitutes CodeBlue for Redcode and provides its own interactive execution interface.

The IRC includes the CodeBlue environment, a user's manual for students, other supporting material, and suggested assignments.

A.5 READING/REPORT ASSIGNMENTS

Another excellent way to reinforce concepts from the course and to give students research experience is to assign papers from the literature to be read and analyzed. The IRC includes a suggested list of papers to be assigned, organized by chapter. A PDF copy of each of the papers is available at box.com/coa10e. The IRC also includes a suggested assignment wording.

A.6 WRITING ASSIGNMENTS

Writing assignments can have a powerful multiplier effect in the learning process in a technical discipline such as computer organization and architecture. Adherents of the Writing Across the Curriculum (WAC) movement (http://wac.colostate.edu/) report substantial benefits of writing assignments in facilitating learning. Writing assignments lead to more detailed and complete thinking about a particular topic. In addition, writing assignments help to overcome the tendency of students to pursue a subject with a minimum of personal engagement, just learning facts and problem-solving techniques without obtaining a deep understanding of the subject matter.

The IRC contains a number of suggested writing assignments, organized by chapter. Instructors may ultimately find that this is the most important part of their approach to teaching the material. I would greatly appreciate any feedback on this area and any suggestions for additional writing assignments.

A.7 TEST BANK

A test bank for the book is available at the IRC site for this book. For each chapter, the test bank includes true/false, multiple choice, and fill-in-the-blank questions. The test bank is an effective way to assess student comprehension of the material.