A Survey of Hands-on Assignments and Projects in Undergraduate Computer Architecture Courses

Xuejun Liang
Department of Computer Science
Jackson State University
Jackson, Mississippi, USA 39217

Outlines

- Introduction
- Categorization of Assignments
- Survey Results and Analyses
- Conclusions and Future Works

Introduction

- According to the Computing Curricula 2001 Computer Science Final Report, core topics of Computer Architecture and Organization include
  1. Digital logic and digital systems,
  2. Machine level representation of data,
  3. Assembly level machine organization,
  4. Memory system organization and architecture,
  5. Interfacing and communication,
  6. Functional organization,
  7. Multiprocessing and alternative architectures,
  8. Performance enhancements, and
  9. Architecture for networks and distributed systems

Introduction (Cont.)

- Teaching the subjects in the computer architecture area can be difficult.
- Hands-on assignments and projects are helpful to students in learning the course subjects.
  - designing, programming, simulating, and implementing a computer processor or a computer system.
- There are too many choices in selecting hands-on assignments and projects for a computer architecture and organization course
- An overall picture of these possible hands-on assignments and projects is certainly desirable
  - Categorizing them and then getting their distribution over different categories

Introduction (Cont.)

- A survey was performed
  - Focused on hands-on assignments and projects given in undergraduate computer architecture and organization courses
  - 35 courses were selected from universities listed among the top 50 Engineering Ph.D. granting schools by the US News & World Report 2008 rankings
  - Teaching materials were publicly accessible via their course websites
- The results from the survey are presented in this paper

Categorization of Assignments

A. Digital Logic Design
   1. Basic Digital Logic Design
   2. Scalar Processor Design
   3. Cache Design
   4. Superscalar Processor Design

B. Assembly Language Programming
   5. Basic Assembly Programming
   6. Advanced Assembly Programming

C. High-Level Language Programming
   7. Basic High-Level Programming
   8. Processor Simulator
   9. Cache Simulator
   10. Advanced High-Level Programming

D. Exploiting Processor Simulators
   11. Using Simulators
   12. Modifying Simulators

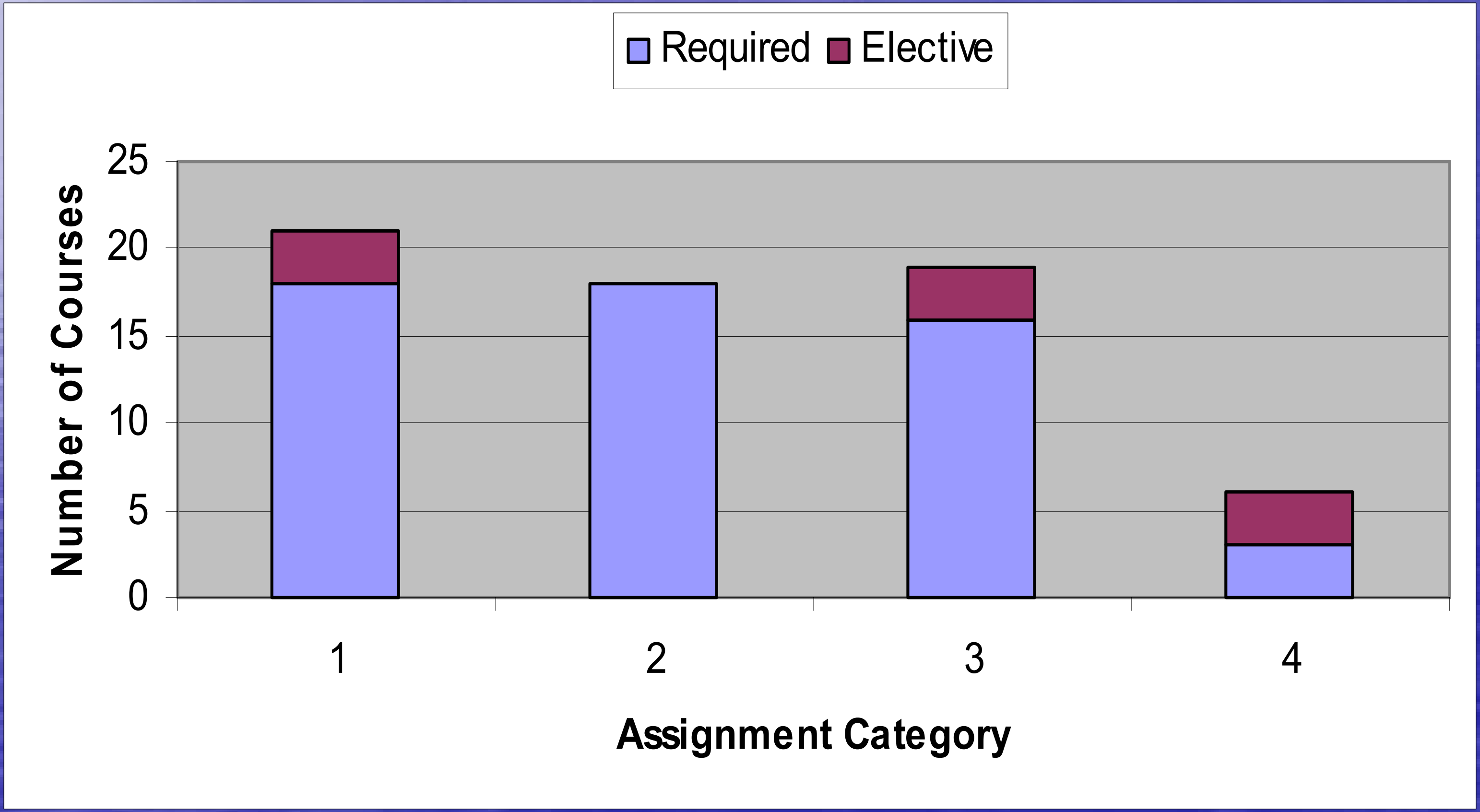
Survey Results and Analyses (1)

Assignment Categorization

#	WT (%)	Assignment and Project Categories											
		A				B		C				D	
		1	2	3	4	5	6	7	8	9	10	11	12
01	35		x			x	x						
02	25		x	x		x	x						
03*	35										x		
04	NA	x	x			x		x		x	x	x	
05*	50		x	x					x				
06	30					x	x	x		x			
07	40					x			x	x			
08*	30	x	x	x	x								

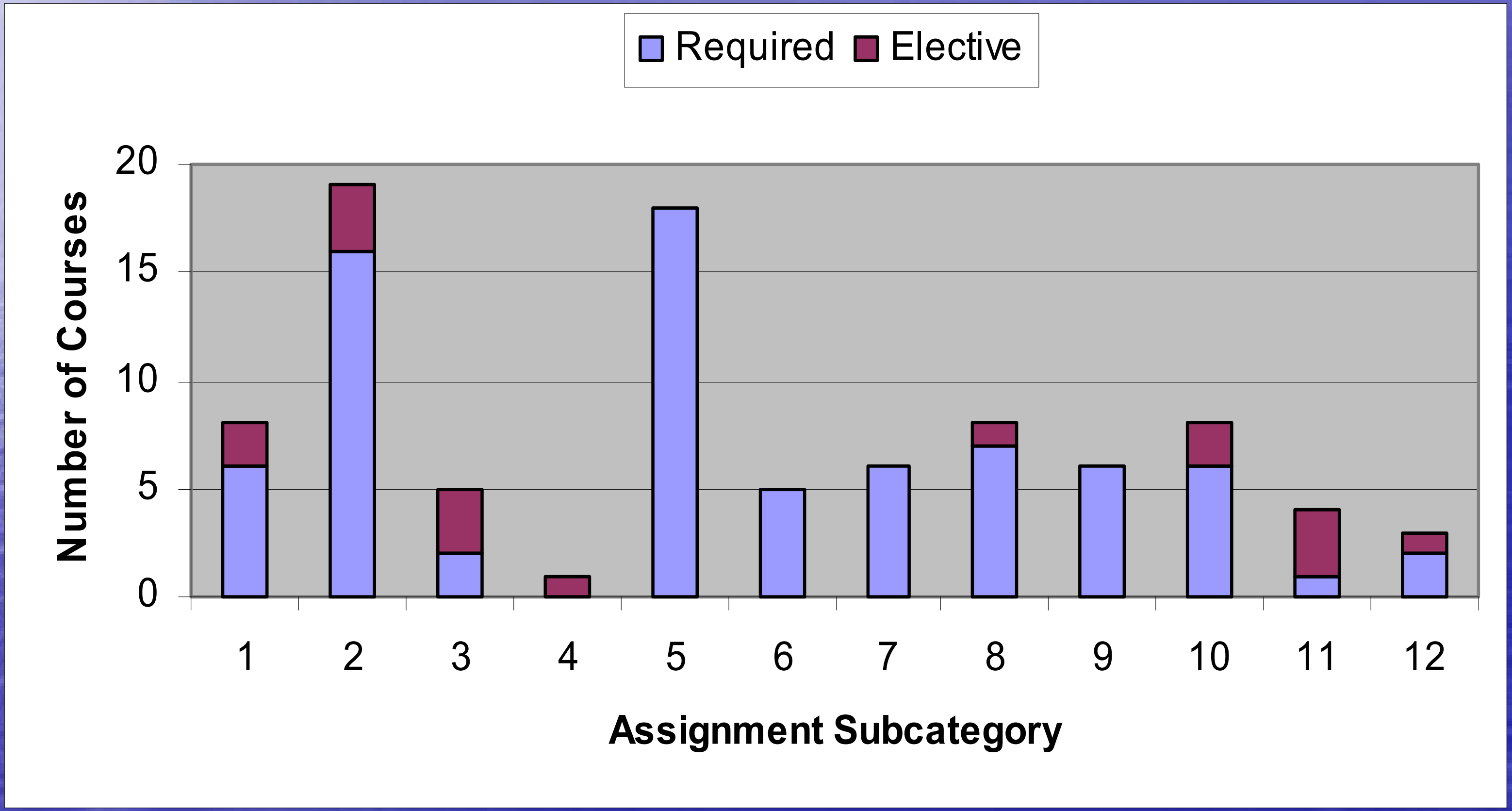
Survey Results and Analyses (2)

Course Distribution over the 4 Categories



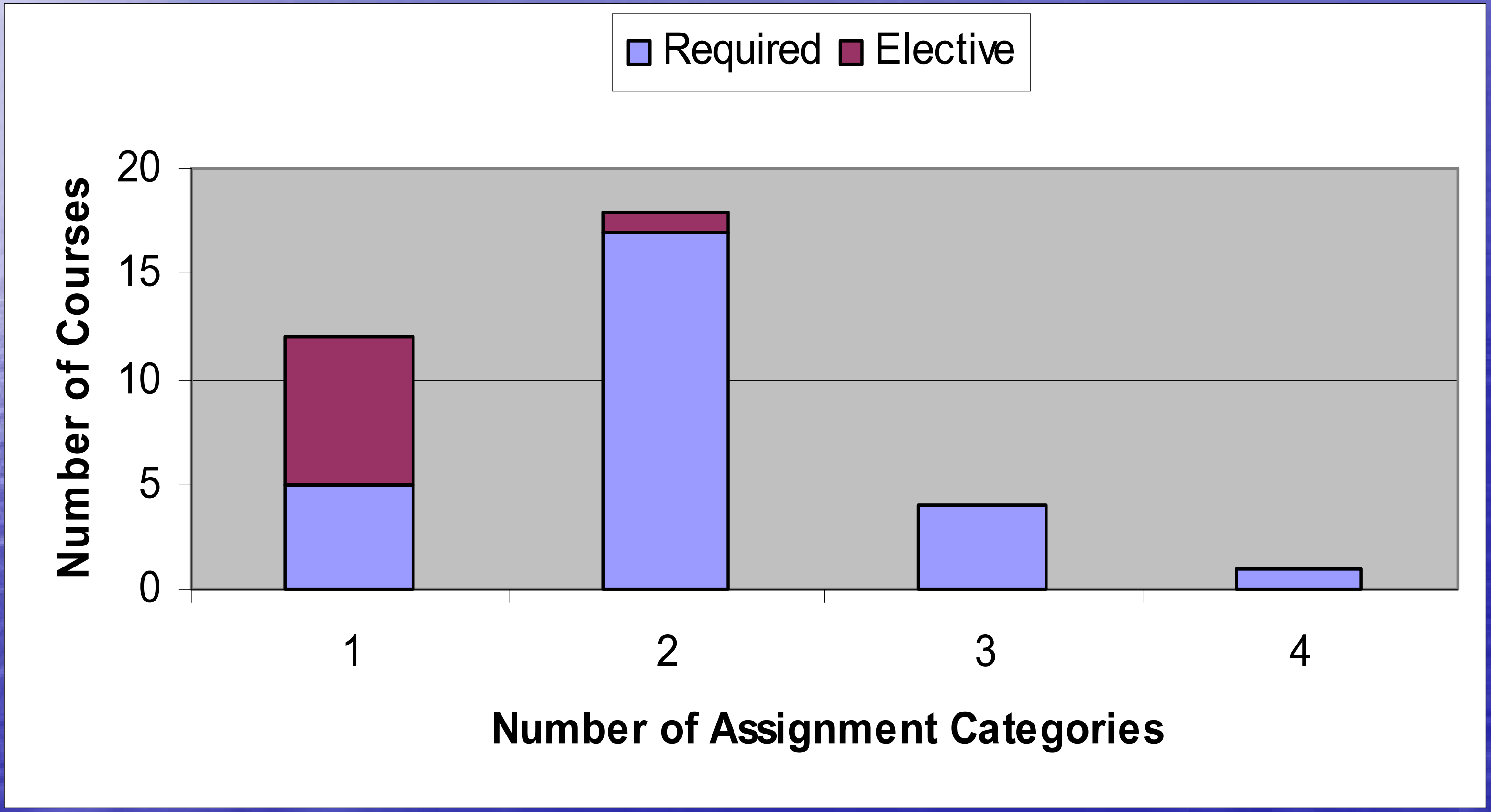
Survey Results and Analyses (3)

Course Distribution over the 12 Subcategories



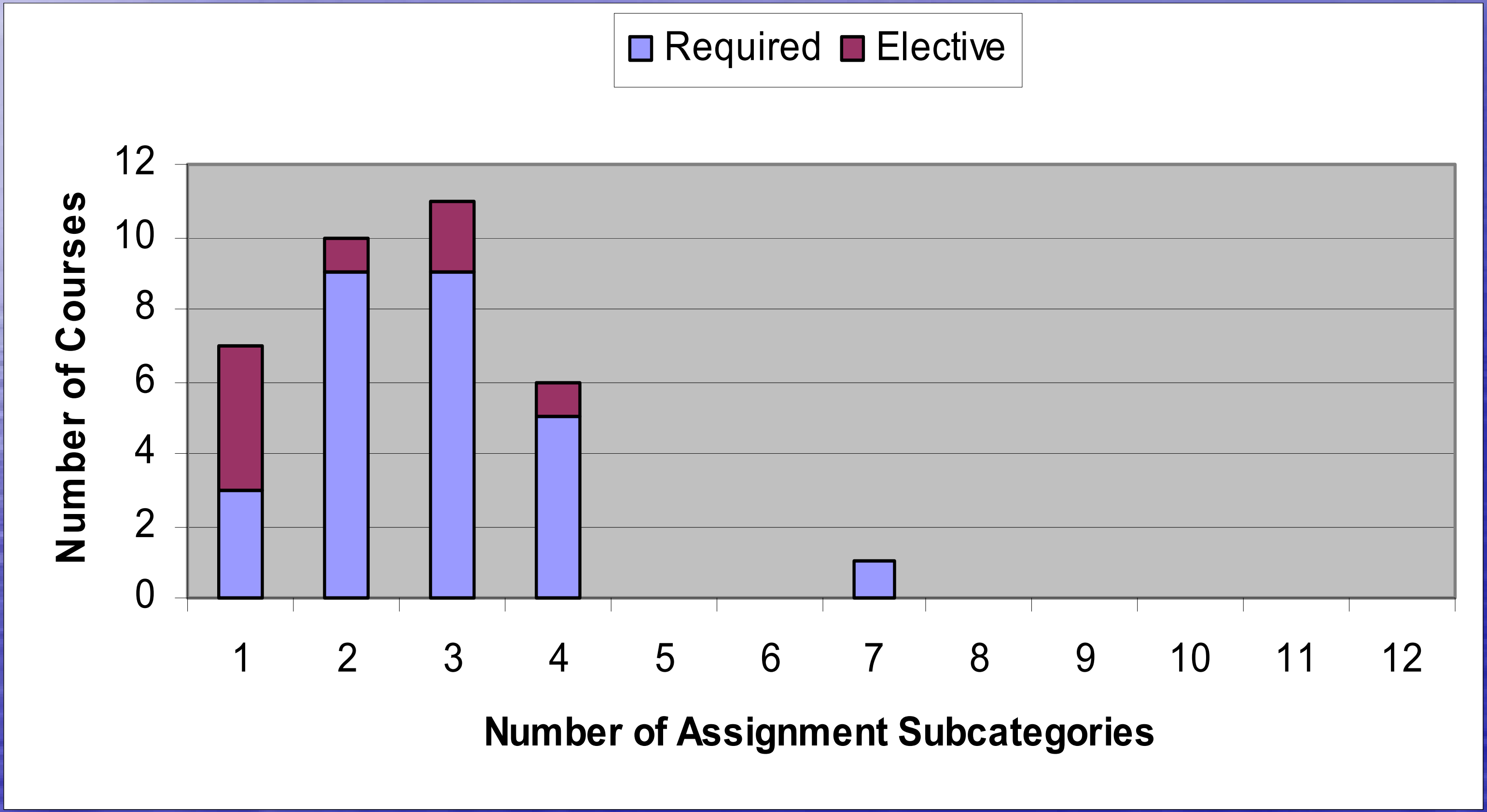
Survey Results and Analyses (4)

Course Distribution over Category Coverage



Survey Results and Analyses (5)

Course Distribution over Subcategory Coverage



Survey Results and Analyses (6)
Digital Logic Design

- Verilog and VHDL are popular
- FPGA is also used in six courses
  - Xilinx's XUP Virtex-II Pro Development System
  - Altera's Development and Education Board
  - XESS's XSA board
  - UC Berkeley's Calinx
- Commercial Logic and FPGA design tools used
  - ModelSim
  - Synopsys VCS
  - Xilinx ISE
  - Altera Quartus II
  - Aldec Active-HDL

Survey Results and Analyses (7)
Digital Logic Design (Cont.)

- Educational logic design and simulation tools used
  - JSIM is a CAD tool from MIT
  - VIRSIM is a graphical user interface to Synopsys VCS
  - Logisim is a graphical tool for designing and simulating logic circuits
  - Chipmunk system software tools from UC Berkeley
  - SMOK/CEBOLLITA from the University of Washington
  - Funsim/Timsim used at Cornell University
- The processor architectures implemented
  - A subset of a well-known processor such as MIPS and Alpha
  - An artificial processor architecture
    - Beta is an educational RISC processor used in MIT
    - PAW is a simple architecture used in the Princeton Univ.
    - Mic-1 is a microarchitecture used in the Tanenbaum's textbook
    - SRC (Simple RISC Computer) is used in the Heuring's textbook

Survey Results and Analyses (8)
Assembly Programming

- Basic assembly programming
  - System I/O, ALU operations, and control flows
  - Stacks, subroutines, and recursions
  - Programmed I/O, interrupts, and exceptions
- Advanced assembly programming
  - A simple timesharing OS kernel on the Beta processor at MIT
  - An interpreter that simulates a subset of the MIPS-I ISA at Stanford
  - SPIMbot contests at Univ. of Illinois–Urbana-Champaign
  - A dynamic memory allocator at Texas A&M Univ.
  - SnakeOS Operating System on LC-3* at Univ. of Pennsylvania

* LC-3 an ISA used in the Patt and Patel’s textbook.

Survey Results and Analyses (9)
Assembly Programming (Cont.)

- Targeted processor architectures
  - Beta, MIPS, LC-2K7 (designed and used at the Univ. of Michigan–Ann Arbor), PowerPC, IA-32, PAW, LC-3, SRC, and x86.
- Processor simulators used
  - BSIM for Beta
  - SPIM and GMIPC for MIPS
  - LC-3 Simulator and PennSim for LC-3
  - SRC Assembler and Simulator

Survey Results and Analyses (10)
High-Level Programming

- High-level programming languages used
  - C, C++, and Java
- Processors simulated
  - MIPS, LC-2K7, a student-designed ISA, PAW, and LC-3
- Advanced high-level language programming
  - Parallel programming on clusters with MPI at Stanford
  - An interpreter to simulate UNIX file system at Berkeley
  - MIPS Multicore Simulator, and Multiplayer Network Tetris Game at Cornell
  - Use shared memory (pthreads) and message passing (MPI) to compute the *N*th prime number at Duke
  - Write a multiprocessor program to do Quicksort running on the MulSim shared-memory multiprocessor simulator at UC-Davis

Survey Results and Analyses (11)
Exploiting Processor Simulators

- Using Simulators
  - Determine cache parameters using CAMERA and study virtual memory using CAMERA and VMSIM at Berkeley.
    - CAMERA is a simple cache simulator
    - VMSIM is a virtual memory management simulator of a computer system executing concurrent processes
  - Use the SimpleScalar to study benchmarking, branch prediction algorithms, cache memory systems, chip multiprocessors, and multithreaded processors
- Modifying Simulators
  - MIC-1 microcode modification
  - The code modification of sim-outorder in SimpleScalar to explore a micro-architectural issue
  - Extend the Mac-1 instruction set by adding a MDN instruction

Conclusions and Future Works

- Present an overall picture of major hands-on assignments and projects currently used in the undergraduate computer architecture education at the top universities in USA.
- Intended for helping educators to select and/or create right hands-on assignments and projects as well as tools.
- Future works
  - Evaluate and compare these hands-on assignments and projects as well as tools.
  - Adopt these hands-on assignments and projects as well as tools in an underrepresented institution