Using an e-Learning Environment to Create a Baseline of Understanding of Digital Logic

Dr. Carolyn Plumb & Dr. Brock LaMeres

Overview

The purpose of this project is to **develop and test a set of adaptive learning course materials to improve mastery of computer engineering.** The interventions proposed will target a sequence of introductory digital logic courses that are found in every ABET accredited computer engineering program.

The first step in the project was to measure the baseline of student understanding prior to deploying the adaptive learning system. A total of 55 specific learning outcomes were created. Course material was developed in the form of a new textbook, instructional videos, and worked examples to teach the content associated with the outcomes. Over 600 assessment tools were created to measure student performance. Data was collected across five sections of the courses at MSU over a calendar year. This poster presents the results of the baseline.

Motivation

Can an adaptive, e-learning environment that provides personalized instruction improve student understanding of computer engineering?

Student Interest – Students lose interest when course material is either too hard or too easy.

Background Deficiencies - Students often lack the necessary prerequisite knowledge in introductory engineering courses due to their varied backgrounds and different high school curricula.

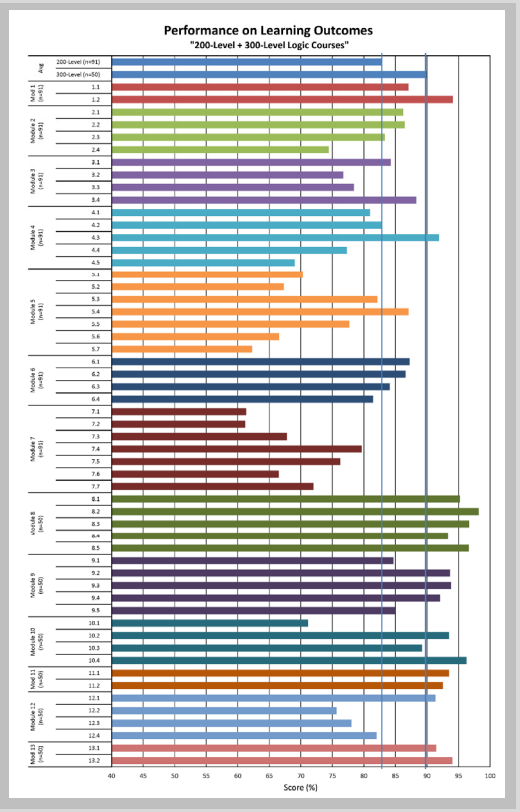
Large Entry-Level Courses – The sheer number of students in introductory courses prevents teachers from providing personalized instruction.

Baseline Data

Learning Outcomes

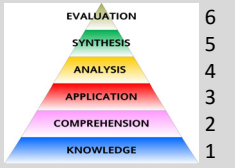
Learning Objective	Learning Outcome	Learning Category 1 2 3 4 5 6
	The overall learning goal of this module is to:	*After completing this module, a student will be able to:*
Module 1: To understand the basic principles of analog and digital systems.	1.1: Describe the fundamental differences between analog and digital systems.	x
	1.2: Describe advantages of digital systems compared to analog systems.	x
Module 2: To understand the basic principles of binary number systems.	2.1: Describe the formation and use of positional number systems.	x
	2.2: Convert numbers between different bases.	x
	2.3: Perform binary addition and subtraction by hand.	x
	2.4: Use two's complement numbers to represent negative numbers.	x
Module 3: To understand the basic electrical operation of digital circuits.	3.1: Describe the functional operation of a basic logic gate using truth tables, logic expressions, and logic waveforms.	x
	3.2: Analyze the DC and AC behavior of a digital circuit to verify it is operating within specification.	x
	3.3: Describe the meaning of a logic family and the operation of the most common technologies used today.	x
	3.4: Determine the operating conditions of a logic circuit when driving various types of loads.	x
Module 4: To understand the basic principles of combinational logic design.	4.1: Describe the fundamental principles and theorems of Boolean algebra.	x
	4.2: Analyze a combinational logic circuit to determine its logic expression, truth table, and timing information.	x
	4.3: Synthesis a logic circuit in canonical form (Sum of Products or Product of Sums) from a functional description.	x
	4.4: Synthesize a logic circuit in minimized form (Sum of Products or Product of Sums).	x
	4.5: Describe the causes of timing hazards in digital logic circuits and the approaches to mitigate them.	x
Module 5: To understand the basic principles of hardware description languages.	5.1: Describe the role of hardware description languages in modern digital design.	x
	5.2: Describe the fundamentals of design abstraction in modern digital design.	x
	5.3: Describe the modern digital design flow based on hardware description languages.	x
	5.4: Describe the fundamental constructs of VHDL.	x
	5.5: Design a VHDL model for a combinational logic circuit using concurrent modeling techniques.	x
	5.6: Design a VHDL model for a combinational logic circuit using a structural design approach.	x
	5.7: Describe the role of a VHDL test bench.	x
Module 6: To understand the basic principles of medium scale integrated circuit logic.	6.1: Design a decoder circuit using both the classical digital design approach and the modern HDL-based approach.	x
	6.2: Design an encoder circuit using both the classical digital design approach and the modern HDL-based approach.	x
	6.3: Design a multiplexer circuit using both the classical digital design approach and the modern HDL-based approach.	x
	6.4: Design a demultiplexer circuit using both the classical digital design approach and the modern HDL-based approach.	x
Module 7: To understand the basic operation of sequential logic circuits.	7.1: Describe the operation of a sequential logic storage device.	x
	7.2: Describe sequential logic timing considerations.	x
	7.3: Design a variety of common circuits based on sequential storage devices.	x
	7.4: Design a finite state machine using the classical digital design approach.	x
	7.5: Design a counter using the classical digital design approach and using an HDL-based, structural approach.	x
	7.6: Describe the finite state machine reset condition.	x
	7.7: Analyze a finite state machine to determine its functional operation and maximum clock frequency.	x
Module 8: To understand the full capability of hardware description languages.	8.1: Describe the behavior of a VHDL process and how it is used to model sequential logic circuits.	x
	8.2: Model combinational logic circuits using a process and conditional programming constructs.	x
	8.3: Describe how and why signal attributes are used in VHDL models.	x
	8.4: Design a finite state machine using the classical digital design approach.	x
	8.5: Describe the capabilities provided by the most common VHDL packages.	x
Module 9: To understand how hardware description languages can be used to create behavioral models of synchronous digital systems.	9.1: Design a VHDL behavioral model for a sequential logic storage device.	x
	9.2: Describe the process for creating a VHDL behavioral model for a finite state machine.	x
	9.3: Design a VHDL behavioral model for a finite state machine.	x
	9.4: Design a VHDL behavioral model for a counter.	x
	9.5: Design a VHDL register transfer level (RTL) model of a synchronous digital system.	x
Module 10: To understand the basic principles of semiconductor-based memory systems.	10.1: Describe the basic architecture and terminology for semiconductor-based memory systems.	x
	10.2: Describe the basic architecture of non-volatile memory systems.	x
	10.3: Describe the basic architecture of volatile memory systems.	x
	10.4: Design a VHDL behavioral model of a memory system.	x
Module 11: To understand the basic principles of programmable logic devices.	11.1: Describe the basic architecture and evolution of programmable logic devices.	x
	11.2: Describe the basic architecture of Field Programmable Gate Arrays (FPGAs).	x
Module 12: To understand the basic principles of binary arithmetic circuits.	12.1: Design a binary adder using both the classical digital design approach and the modern HDL-based approach.	x
	12.2: Design a binary subtractor using both the classical digital design approach and the modern HDL-based approach.	x
	12.3: Design a binary multiplier using both the classical digital design approach and the modern HDL-based approach.	x
	12.4: Design a binary divider using both the classical digital design approach and the modern HDL-based approach.	x
Module 13: To understand the basic principles of a computer system.	13.1: Describe the basic components and operation of computer hardware.	x
	13.2: Describe the basic components and operation of computer software.	x

Student Performance



Each outcome is associated with a learning "category" corresponding to Bloom's Taxonomy. This ensured that the assessment tools were developed to measure the correct ability that the outcome targeted (i.e., knowledge vs. synthesis, etc.).

This also allows the "cognitive difficulty" of the particular outcome to be tracked as another means of selecting where to deploy the adaptive learning modules.

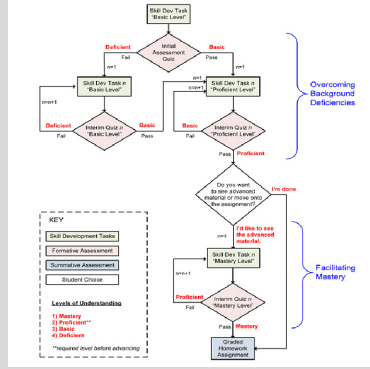


Current Status

The course material has being deployed at four diverse institutions (MSU-Bozeman, MSU-Billings, Flathead Valley Community College, and Salish Kootenai Tribal College). Baseline data has been collected for 3 terms across 5 sections. Student performance on the outcomes, combined with the Bloom's Taxonomy classification, point to which outcomes to use for the adaptive learning modules.

Next Steps

The adaptive learning modules are being finalized for the outcomes that have the lowest student performance. These will be deployed in the Fal-16 semester.



Adaptive Learning Flowchart

2016 ASEE Annual Conference and Exposition
New Orleans, LA June 26- 29, 2016.

This work is supported by the NSF Program *Improving Undergraduate STEM Education (IUSE)* through award number 1432373.