Flip-Flops in Students' Conceptions of State

Geoffrey L. Herman, *Student Member, IEEE*, Craig Zilles, *Member, IEEE*, and Michael C. Loui, *Fellow, IEEE*

***Abstract*—The authors conducted a qualitative interview-based study to reveal students' misconceptions about state in sequential circuits. This paper documents 16 misconceptions of state, how students' conceptions of state shift and change, and students' methodological weaknesses. These misconceptions can be used to inform and direct instruction. This study revealed a need for the development and adoption of standard terminology and a need to focus digital logic instruction upon a central concept of information encoding. In addition, these misconceptions will serve as the basis for the creation of standard assessments called concept inventories. A concept inventory will provide rigorous and quantitative metrics to assess the effectiveness of new teaching methods.**

***Index Terms*—Combinational circuits, computer science education, logic circuits, sequential circuits, state machines.**

I. Introduction

ENGINEERING instructors are often baffled by students' mistakes. Students can solve complicated design problems one moment, and then do something that seems nonsensical the next moment. In response to these baffling mistakes, engineering education researchers are investigating why students struggle to solve problems, complete design projects, and even learn basic concepts. Researchers are documenting what misconceptions students have and why they develop misconceptions [1]–[3]. When the what and the why of misconceptions are known, instruction can be changed to remedy the misconceptions.

"State" is a pervasive topic in electrical and computer engineering (ECE) and computer science (CS). Students encounter the concept of state in courses on computer architecture, theory of computation, software development, digital signal processing, and stochastic processes (e.g., Markov models). The concept of state is so ubiquitous in computing that it could be claimed that a computer is capable of only two actions: 1) to analyze the state of the system, and 2) to change the state of the system.

Because state is central to computing, students need a firm, accurate grasp of this concept from the beginning of their studies, when state is first taught during digital logic courses. After documenting a framework of students' misconceptions of state, researchers and instructors can decide how to diagnose these misconceptions quickly and develop ways to remedy them.

The study in this paper (for which preliminary work appeared at the 39th ASEE/IEEE Frontiers in Education Conference, 2009 [4]) is part of the effort to develop a concept inventory (CI)—a multiple-choice test that reliably and validly measures student conceptual understanding—to quickly assess the misconceptions that students possess [5]. The state misconceptions presented in this paper, along with misconceptions documented in other studies, will be used to create distracters (wrong answers) for a digital logic concept inventory (DLCI) [4], [6]–[9].

The DLCI could reform digital logic instruction in the same way that the Force Concept Inventory (FCI) has revolutionized physics education [10], [11]. The FCI prompted the adoption of interactive engagement teaching methods and peer instruction [10], [12], [13]. A CI creates the impetus for change by providing a way to rigorously, objectively, and empirically compare conceptual learning. By comparing conceptual learning gains, researchers and instructors can compare the effectiveness of different teaching methods.

Student misconceptions about state are presented both to document the steps taken to create the DLCI and to provide insight into students' mental models of state. A detailed description of these misconceptions will have immediate implications for improving digital logic instruction.

II. Background

> Student 6: "*State is . . . how would I define that. Let's see, it's one of those things that you come to just like think of but not put in words*."

State may be hard for students to learn for two reasons: Instructors may not be always aware that they are teaching state, and the term "state" is both ambiguous and abstract.

A. Instructor's Tacit Knowledge

When a person has applied a concept or procedure until the application becomes cognitively automated, that person has "over-learned" that concept or procedure [14], [15]. A common over-learned activity is signing one's name. It is easy to write one's signature and hold a conversation at the same time. When a hand or finger is broken, though, writing a signature while holding a conversation becomes considerably harder or even impossible. This cognitive automization can be considered to be the compression of cognitive information so that the information takes up less space in working memory (much like data compression on a computer). Once cognitive information is compressed, the knowledge becomes the tacit knowledge of an expert—knowledge that is used, but not always demonstrated or expressed explicitly.

Manuscript received December 27, 2010; accepted March 03, 2011. Date of publication April 21, 2011; date of current version February 03, 2012. This work was supported by the National Science Foundation under Grant DUE-0618589.

G. L. Herman and M. C. Loui are with the Department of Electrical and Computer Engineering and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: glherman@illinois.edu; loui@illinois.edu).

C. Zilles is with the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: zilles@illinois.edu).

Digital Object Identifier 10.1109/TE.2011.2140372

Because instructors have so much experience with the concept of state, they commonly over-learn it [16]. They manipulate and apply the concept of state consistently, accurately, and without conscious effort. They tacitly use the concept of state in a number of diverse contexts and may not even realize that they have used it. This tacit use makes it harder for an instructor to unpack and explain the concept of state and its use. The level of familiarity with this concept also makes it easier for instructors to overlook the difficulty of the concept or understand why students struggle to apply it [17].

B. Contextual Definitions as a Source of Misconceptions

The term "state" is applied to many similar but different abstract concepts.

When a concept is abstract, students make two common mistakes: developing a definition of the concept that is grounded in the context of a problem, or equating the abstract conception with a concrete object. Research has shown that students manifest different misconceptions when asked to use a concept in differing contexts or problems [18], [19]. Often, these misconceptions develop from commonsense interpretations of immediately observable features of common problems [20]–[22]. When students focus on these surface features, they classify many concepts poorly. For example, students often think of electrical current as a fluid rather than a constraint-based process (i.e., concrete rather than abstract) [23]. These classification problems have even been observed in digital logic. Students commonly treat Boolean variables as physical objects rather than as representations of the truth of a condition [6].

Students use the term "state" in contexts such as chemistry and physics as well as everyday language. The state of matter (solid, liquid, or gas) in chemistry is a technical conception of state, but there are subtle differences between the meaning of state in this context and the digital logic context: A phase transition in chemistry produces a change in state, but a state transition in digital logic might not necessarily produce a change in state. Because students use the term "state" in multiple contexts, they may develop mental models of state that are inconsistent with the precise definition of state used in digital logic.

III. Methodology

The authors conducted a qualitative research study using standard methods to document students' misconceptions [24]. This section explains how students were selected for interviews and how they were interviewed.

A. Students

In Spring 2008, Fall 2008, and Spring 2009, the authors interviewed nine, six, and 11 undergraduate students, respectively, at the University of Illinois at Urbana–Champaign. Additional rounds of interviews were added until the analysis of the interviews failed to reveal new misconceptions. Students were recruited each semester from two large three-credit digital logic courses, one each in the Department of Computer Science and the Department of Electrical and Computer Engineering. Both courses were taught by instructors who had taught their respective courses for multiple semesters and been rated highly by their students, used the same textbook [25], used similar syllabi, and had about 200 students per semester. Both courses were lecture-based and administered online homework assignments, weekly paper-based homework assignments, simulation labs, two midterm exams, and a final exam. All interviewed students were traditional-age (18–22 years old) undergraduates majoring in computer science, electrical engineering, or computer engineering who had just completed one of the digital logic courses and had earned grades of B or C (from 1.7 to 3.3 on a 4.0 scale). These students were selected because their understanding was likely to be less complete than that of students with higher grades (i.e., more likely to have misconceptions), thus interviewing students with higher grades would not reveal misconceptions that were not present in the students with lower grades. Pilot interviews confirmed these expectations.

B. Interview Process

Students were interviewed for 1 h. Interviews were conducted in a modified "think-aloud" format: Students were instructed to vocalize their thoughts as they solved problems and responded to questions [26]. Prior to the interview, students were briefed on the study's goal of understanding how they think through various topics in digital logic design. They were told not to expect feedback during the interviews about whether their answers were correct, but to expect frequent requests to elaborate on what they were doing [26].

All interviews were recorded using a document camera (which recorded only what the student wrote) and microphone. The audio tracks of the interview recordings were transcribed verbatim, the students' gestures were included in the transcript, and every piece of paper the student wrote on was scanned electronically. Quotations presented in this manuscript have been "cleaned up" to remove excessive "likes," "ums," and repeated phrases. "Cleanup" was performed only when removing these artifacts did not change the content of the statements.

Students were paid for their participation, and all students gave written consent to be interviewed under IRB approval (University of Illinois at Urbana–Champaign number 07026).

C. Interview Questions

All students were asked questions spanning much of digital logic with several questions pertaining to state. Each semester, the authors interviewed students using a slightly different set of questions based on their analysis and findings from the previous round of interviews. For example, after students from the first round of interviews demonstrated a weak understanding of the different types of flip-flops, new questions were created to further investigate students' misconceptions about the different types of flip-flops.

All state machines were based on a Mealy machine design to match course presentation [25]. Students were also presented with equivalent versions of the various circuits shown in this paper. Students were allowed to choose whichever version of the circuit seemed most intuitive to them. Allowing students to choose their preferred circuit design was intended to reduce the possibility that a student's mistakes could be attributed to unfamiliarity with a specific circuit layout (to save space, only the circuit layouts that require the least space are shown here).

A finite state machine that has *n* states requires at least *m* flip-flops to implement as a sequential circuit. If a different finite state machine has 2*n* states, what is the minimum number of flip-flops needed to implement it?

Fig. 1. Example of a relationship question that tests a student's understanding of the relationship between flip-flops and state. Subjects should respond that $m + 1$ flip-flops would be needed for $2n$ states.

Students in both courses were primarily taught design methods to minimize the number of flip-flops in a circuit, given a fixed number of states. Interview questions focused on how well students understood the concepts that underline that design methodology.

Students interviewed during Spring 2008 were asked a series of open-ended *definition questions*, *relationship questions*, and *design problems*. *Definition questions* required students to explain a concept or term as if they were teaching a student in a first course in digital logic design. *Relationship questions*, such as that shown in Fig. 1, required students to determine how changing the quantity or values of circuit components would change the other components and concepts associated with them. The design problems were intended to simulate design problems the students may have encountered in their digital logic design course.

Students interviewed during Fall 2008 were asked a series of refined open-ended definition questions, relationship questions, design problems, and analysis problems. Some analysis problems asked students to perform routine analysis tasks, but hid some information from them (e.g., Question 6 in Fig. 3).

Students interviewed during Spring 2009 were asked a mixture of open-ended analysis problems and multiple-choice questions from the alpha version of the DLCI [5]. These multiple-choice questions were adapted from the questions and responses of the students interviewed earlier. The alpha version of the DLCI has been administered to 208 students at the University of Illinois at Urbana–Champaign.

Other interview questions and their acceptable answers will be introduced when needed.

D. Data Analysis

Interviews were analyzed using a grounded theory approach as described by Kvale [27], Strauss and Corbin [28], and Miles and Huberman [29]. The analysis protocol is described in depth in previous works [4], [6]. The three authors of this paper analyzed the data.

E. Terminology

This section defines terminology that is used for the remainder of the paper. The term *student* describes any person who has recently learned digital logic or is currently learning digital logic. The term *subject* describes any student who participated in the interview portion of the study. All subjects are given pseudonyms, such as "Subject 1," and are numbered according to a standard system employed by the authors across their publications in this research study.

A *finite state machine* consists of a finite set of *states* S, a finite set of *input symbols* I, a finite set of *output symbols* O, a transition function $\delta : S \times I \rightarrow S$, and an output function $\omega : S \times I \rightarrow O$. The transition function maps each pair of a state and an input symbol to a *next state* in S. The output function maps each pair of a state and an input symbol to an output symbol.

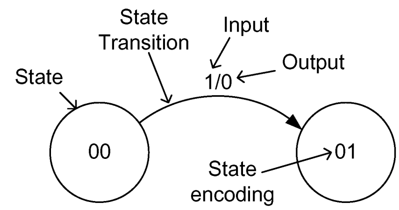


Fig. 2. Partial state diagram of a finite state machine.

A finite state machine can be implemented by a sequential circuit, which has flip-flops, inputs, and outputs. Unless otherwise specified, *input* and *output* refer to the inputs and outputs of the circuit rather than the inputs and outputs of individual gates and flip-flops. Each input sends a Boolean signal to the circuit, and each output sends a Boolean signal from the circuit. When a sequential circuit implements a finite state machine, each combination of values stored in the circuit's flip-flops encodes a particular state. Each combination of values of the inputs encodes a particular input symbol. Each combination of values of the outputs encodes a particular output symbol.

Fig. 2 shows part of a finite state machine with two states encoded 00 and 01, two input symbols encoded by 0 and 1, and two output symbols encoded by 0 and 1. Each circle represents a state. The arc represents a transition from state 00 to next state 01 under input value 1. When the state is 00 and the input value is 1, the output value is 0.

Eight digital logic textbooks offer the following definitions of state. One textbook did not offer a formal definition of state [30].

1) State "is all the bits stored in the circuit" [31].
2) State is represented by "the contents of the storage elements" [32].
3) "The *state* of a sequential circuit is a collection of *state variables* whose values at any one time contain all the information about the past necessary to account for the circuit's future behavior." This definition was originally crafted by Hellerman [33] and is later cited by Wakerly [34].
4) State "is the mechanism which is used to explain and represent the [past history of inputs]" [35].
5) "What is stored in memory is the *state* of the system" [36].
6) State is the "binary encoding [of] the state memory at one particular instance of time" [37].
7) State is "the binary information stored in [the circuit's storage elements] at any given time" [25].

For the rest of the manuscript, the authors use a definition that combines definitions 6 and 7: "State is the binary encoded information stored in the circuit's storage elements at any given time." While definition 3 may be the most precise, the definition chosen most directly relates to the definitions used by subjects during interviews.

The IEEE standard definition of state is currently withdrawn (see IEEE Standard 610.10-1994). The withdrawn definition was previously "the input to and information stored in a circuit

TABLE I
CONCEPTIONS AND MISCONCEPTIONS OF STATE LISTED IN ALPHABETICAL ORDER BY CATEGORY. $^*n$ = THE NUMBER OF STATES. m = THE NUMBER OF FLIP-FLOPS. i = THE NUMBER OF INPUTS. o = THE NUMBER OF OUTPUTS

Category	Name of conception	Explanation of conception
Correct	Correct	Subject correctly understands both that state is encoded using binary bits and that these bits that compose state can be found in the memory components of the circuit.
	Memory	Subject correctly associates the state of a circuit with its memory components (i.e., flip-flops, registers), but the subject does not directly explain how state is encoded in the memory values during this statement.
	Property	Subject defines state broadly as a property that characterizes the condition of an object, but fails to specifically relate this idea to the state of a circuit. (correct but incomplete)
	Location conceptions	
Signals	Save	Subject thinks the state of a circuit is only an instantaneous value that must be stored or kept. This subject believes that memory and state are two completely distinct concepts.
	Value	Subject thinks the state of a circuit is the value of a signal within the circuit (e.g., If a signal's value is 1 then it is in the 1 state).
Circuit components	All Values	Subject thinks the state of a circuit is the value of all signals/circuit components in that circuit.
	FFs Inputs	Subject thinks the state of the circuit is the values held by the FFs and any inputs. (e.g., $n = 2^{m+i}$, $n = m2^i$)
	FFs Outputs	Subject thinks the state of the circuit is the values held by the FFs and any outputs of combinational logic (e.g., $n = 2^{m+o}$ or $n = m2^o$)
	In Out	Subject thinks the state of a circuit is the combination of all inputs and outputs of the circuit.
	Inputs	Subject thinks the state of a circuit is the value of the circuit's inputs
	Outputs	Subject thinks state of a circuit is the output of the circuit.
Negative locations	No def	Student does not offer a formal definition of state when specifically asked to define the term.
	Not FFs	Subject explicitly ignores the contents of flip-flops when analyzing the state of a circuit.
	Not Inputs	Subject correctly says that the state of a circuit is not the inputs of the circuit, but does not offer a positive statement about what circuit components compose state.
	Not Outputs	Subject correctly says that the state of a circuit is not the output of the circuit, but does not offer a positive statement about what circuit components compose state.
	Encoding conceptions	
Encodings	1x1 FFs	Subject thinks that at an instant of time there is one state per FF in the circuit. Student fails to think of the circuit state as an abstract whole.
	Exponential	Subject can recall the exponential relationship between number of "stateful" circuit elements and the number of potential states in the circuit. It is not clear that the subject always understands that state is encoded using bits or that state is encoded in memory.
	Linear	Subject incorrectly thinks there is a linear relationship between number of "stateful" circuit elements and the number of states (e.g., $n \propto m$). Subject does not use a minimal encoding scheme.
	Quadratic	Subject incorrectly thinks there is a quadratic relationship between the number of "stateful" circuit elements and the number of states (e.g., $n = m^2$).
	Type	Subject thinks that different types of FFs can encode different numbers of states (e.g., JK FFs can encode four states and D FFs can encode two states)

or device," but this definition contradicts the definitions used by the textbooks.

IV. RESULTS AND THEMES

A complete list of the misconceptions discovered can be found in Table I. Misconceptions about state are classified into two types: location misconceptions (misconceptions about what components encode state) and encoding misconceptions (misconceptions about how state is encoded within those components). The more common misconceptions are presented.

A. Location Misconceptions and Indefinite Definitions

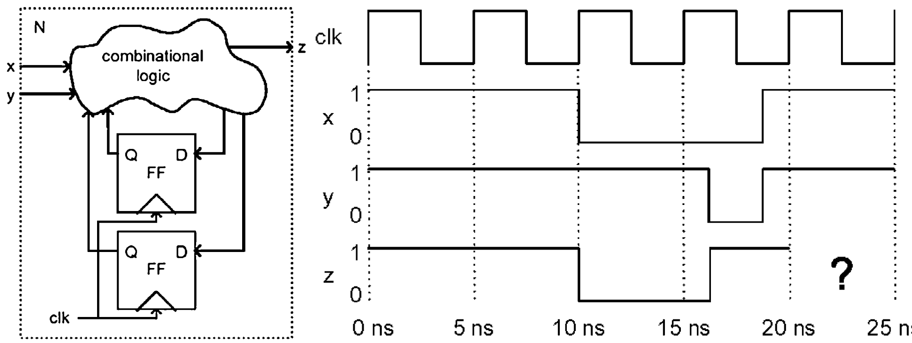
Subjects knew that state was associated with the information in a circuit, but they often did not know what information is relevant to the state of a circuit or where that state is encoded. Even when they knew that state was related to information, subjects struggled to describe state accurately in technical terms and had to resort to using specific examples (state diagrams) or physical location metaphors (Subject 5: "*State is where you are*."). Subjects used more than 10 invalid definitions of state. These definitions indicated that state was encoded in a variety of circuit components.

State as Output (Table I: Outputs): Eight subjects defined state as the the value of the circuit's outputs. This definition was the most common misconception.

> Subject 8: "For state, the inputs can, you never care about the inputs. I mean 'cause you can change the inputs and still have the same exact outputs and still be in the same state.... So I said *[state] was just the current output of the circuit*."

This subject claimed that state comprises the outputs because the outputs may not change even though the inputs change. This reasoning shows that state as output is attractive because output can be thought of as the measurement that summarizes everything of interest that is inside the circuit.

Question 6. The block diagram of a synchronous finite state machine – circuit N – is shown below. Circuit N is composed of combinational logic (of unknown design) and positive edge triggered D flip-flops. Circuit N also has two inputs *x* and y and one output *z*.



Suppose that the timing diagram is produced by circuit N for times 0-25 ns. Which of the following statements is true about the output z for times 20-25 ns?

1) The output z will be 1, because x and y are both 1 as they were for 0-10 ns.
2) The output z will be 1, because x and y are held constant for 19-25 ns.
3) The output z will be 1, because x and y are repeating the same input sequence starting at 20 ns that was started at 0 ns.
4) The value of the output z cannot be determined from the given information.

Fig. 3. Sequential circuit analysis interview and DLCI question. The correct answer is choice 4.

Subjects may also have developed this misconception because a flip-flop's output is its state. By overgeneralizing, a subject may believe that the output of any circuit is its state.

This conception can be paraphrased as "state is the information that a circuit provides to an external entity (circuit or user)."

State as the Inputs and Outputs (Table I: In Out): Seven subjects defined state as the values of a circuit's inputs and outputs. This definition was commonly revealed when subjects were solving Question 6 shown in Fig. 3. When solving this problem, Subject 1 said, "If x and y (*the inputs*) are both high already and z (*the output*) is already high, that's an identical state as [earlier]." In other words, states are identical when the inputs and outputs are identical.

When solving the same problem, another subject asserted that the values of the flip-flop outputs were irrelevant when he knew the value of the inputs and outputs.

> Subject 7: "Well I assumed that *the combinational logic just sort of canceled out the uncertainty in the flip-flops*."

This conception can be paraphrased as "state is the information in communications between a circuit and an external entity (circuit or user)."

State as the Values of All Signals (Table I: All Values): Four subjects defined state as the value of all signals in a circuit.

> Subject 1 explicitly defined state as the following: "The state in a circuit is where you're currently at. So if you have some sort of datapath or something you have certain values, like if you have [circuit inputs] or if you have [circuit outputs]. So if you have some sort of arithmetic unit you have what your result is currently, so I guess *state is just where you're at with all of your values*."

Similarly, Subject 15 said, "The state represents everything that is in [the circuit]."

This conception can be paraphrased as "state is all information in the circuit."

State as Inputs and Storage Elements (Table I: FFs Inputs): Five subjects defined state as the value of the inputs and storage elements of the circuit. Subject 9 stated this definition, and when asked why, replied as follows.

> "Because, I think the outputs are dependent on [the inputs and flip-flops], 'cause that's all you need to know about a circuit to determine the outputs. *Because if you know the circuit, you can find the outputs given [the inputs and flip-flops], so that's more information than you need, right?* If you don't know the circuit then you can't find the outputs. So I probably wouldn't put [outputs] just because, we're usually given the implementation of the circuit."

Subject 9 used the "inputs and storage elements" as the definition of state because the values (1 or 0) of the inputs and storage elements can be used to derive the values of all other signals. If the circuit implementation were absent, the subject said he would have included the outputs as part of the state. This definition of state is similar to the *state is all values* definition, but this definition finds the minimum number of signals needed to derive all values. Subjects who used this definition understood that there are different types of information in a circuit, but they failed to distinguish between circuit memory information and user information.

This conception can be paraphrased as "state is the minimum information needed to derive the values of all circuit components."

TABLE II
LIST OF STATE CONCEPTIONS THAT EACH SUBJECT USED, ORDERED FROM MOST DEFINITIONS USED TO LEAST. HIGHLIGHTED ENTRIES EMPHASIZE CONCEPTIONS THAT ARE CONTRADICTORY. SEE TABLE I FOR EXPLANATIONS OF THE ABBREVIATIONS

Subject	1	2	8	11	10	6	13	4
Concepts used by the subject	All Values In Out Linear Memory Not FFs Outputs Save	All Values Exponential Inputs Not Outputs Outputs Property Value	Exponential FFs Inputs FFs Outputs Memory Not Inputs Outputs	1x1 FFs FFs Inputs FFs Outputs In Out Linear Memory No Def	1x1 FFs Exponential FFs Inputs In Out Outputs Quadratic	Linear Memory No Def Outputs Save	Exponential FFs Inputs Inputs In Out Linear	Memory Not FFs Outputs Value
Subject	**5**	**7**	**16**	**15**	**3**	**9**	**14**	**12 & 17**
Concepts used by the subject	In out Memory Type Value	Exponential In Out Memory Outputs	All Values FFs In Out Outputs	All Values Memory Value	Linear Memory	FFs Inputs No def	Correct Outputs	Correct

State as the Value of a Signal (Table I: Value): The final location misconception reveals that the subjects did not know what circuit components have state, and that they also conceive of state as something that is a property of every circuit component individually. Subjects revealed that they did not naturally conceive of state as a single trait of the entire circuit as a whole. Subject 2 attributed state to every possible value of every signal in a timing diagram: "This is the one state, that's a zero state. One state, zero state . . . [subject points to all 1's and 0's by the signal names in the timing diagram]." Similarly, Subject 14 said, "[State] is probably on or off. I'm assuming." For these subjects, state was the value of an individual signal or descriptor.

This conception can be paraphrased as "state is the information revealed by a single signal."

Students' Shifting Definitions: Table II lists the various definitions that each subject used during the interviews. Table I explains the abbreviations used in Table II. The highlighted definitions are contradictory definitions.

Several subjects changed their definitions of state for nearly every problem they were asked to complete. As can be seen in Table II, only Subjects 12 and 17 used consistent definitions of state throughout the entire interview. Not surprisingly, these subjects held the correct definition of state. Notice also that no subject who used more than two definitions ever gave a correct and complete definition of state. On average, subjects used four distinct definitions of state. These definitions often included mutually exclusive portions of the circuit or were even contradictory (e.g., Subject 11 said that state is found in only memory and later said that state is found in only the inputs and outputs). While subjects' definitions tended to shift when changing contexts or problems, some subjects shifted definitions within a few sentences in the same context. Subject 16 used two definitions in close proximity; the correct definition is in bold face, and the incorrect definition is in italics.

> "**[the circuit] has, three flip-flops so, there's eight states. Because, you need two to the m flip-flops for n states. So two to the third, so eight potential states**. *At any given time it has, I guess one state. Because, it's one input and one output*."

Indefinite Instructor Definitions: Analysis of the interviews revealed that instructors also may be teaching indefinite definitions. The term *state transitions* was particularly ambiguous.

Eight textbooks were consulted for their definitions of state transitions [25], [30]–[32], [34]–[37]. Three of these explicitly define a state transition to be any arrow on a state diagram or how a sequential circuit responds to a relevant clock edge [31], [34], [35]. Five of them offer no formal definition of state transition. Of these five, two gave examples that implied that a state transition is any arrow on a state diagram. The other three gave examples that strongly implied that a state transition is a change of state, and these textbooks made a strong distinction between state transitions and the term for the arrows on a state diagram. The arrows on state diagrams were even identified by four different terms: *arcs*, *edges*, *branches*, and *transitions*.

Because this terminology is ambiguous, the authors decided against identifying misconceptions about the state transition concept [38]. This ambiguity is important, though, because it might be a source of confusion and conceptual difficulty for students.

B. Encoding Difficulties

Many subjects demonstrated that they did not know how state was stored in a circuit even if they knew which circuit components stored the circuit state. In a typical circuit, state information is encoded in the flip-flops as bits. Optimal bit encoding dictates that the addition of one flip-flop doubles the number of states that the circuit can potentially be in over time (i.e., the maximum number of states is 2^m when the circuit has m flip-flops). Some subjects demonstrated confusion about how to encode state using this exponential relationship (Table I: Exponential) and asserted that states and flip-flops are related by linear (i.e., the number of states is proportional to m, Table I: Linear) or quadratic relationships (i.e., the number of states is m^2, Table I: Quadratic). When answering the question in Fig. 1, Subject 6 expressed this linear relationship by saying, "The reasoning is you've got just double the number of states to save and if you already accounted for exactly this, if you had exactly enough room to save n states, then to save more you're going to need the same amount more." Subject 10 expressed the quadratic relationship with similar reasoning when answering the question in Fig. 1: "[The problem statement] says that n states requires m flip-flops, so $2n$ states is, you have a set of n states, and you have another set of n states, so [the first set] requires m, [the second set] requires m. So it's gonna be m times m. So it's m squared." When asked to describe the relationship between states and flip-flops on alpha versions of the DLCI, 25% of students chose the linear relationship, and 12% of students chose the quadratic relationship.

The linear encoding misconception is straightforward, but the quadratic encoding misconception is more troubling. When reading the quotations from Subjects 6 and 10, one could reasonably assume that both subjects would arrive at the linear encoding scheme. Both subjects used the same reasoning up until the end of their answers. They both described flip-flops as containers that hold a certain number of states, and that to hold an extra set of states, the circuit would need an extra set of containers. The two subjects diverged when they needed to explain this mathematical relationship using symbols. Subject 10 made a second conceptual mistake by claiming that he should multiply the two sets of containers rather than add them together.

A final encoding misconception was seen in comments where subjects asserted that a circuit (with memory components) would not have state until the user specified the state. Subject 2 made this claim saying, "Suppose you have a chunk of memory, and you allocate it, but you dunno what's there, you can't really talk about the state." This subject either failed to understand that the circuit will have state even before the user allocates chunks of memory to a program or circuit, or that the circuit has state even if the user does not know what is stored in the memory. These misconceptions reveal that students do not always associate state with the encoding of information in the memory—especially if that information is random. This misconception may be caused by interference from what subjects learned in programming classes.

Flip-Flop Types (Table I: Type): Five of six subjects who were asked about the differences between flip-flop types struggled to recall the nature of the different flip-flop types. Some of these subjects did not believe that all flip-flops store only one bit of information regardless of type. Instead, these subjects believed that JK and SR flip-flops (which have two inputs) store twice as many states as D and T flip-flops (which have one input).

Subjects also failed to understand why they would use different flip-flop types in different situations and that the different flip-flop types are arbitrary tools that were created to facilitate design. Subjects asserted that different flip-flop types *must* be used in specific situations. They failed to understand that flip-flop types are interchangeable parts given the proper conversion techniques.

Flip-Flop State Versus Circuit State (Table I: 1 × 1 FFs): Subjects struggled to understand how a circuit has a single state at a given moment of its operation, but can be in many states over the course of its operation. Subject 10 demonstrated this misconception when asked how many states a circuit would have when it contained three flip-flops. This subject initially answered the question correctly by answering that the circuit could potentially be in eight different states, but then revealed his misconception.

> Subject 10: "So out of those eight [potential states], how many are we actually using? So we could be using like four of the states, or five of the states. Not all eight.... So I think that [it] might have one state per each [flip-flop]. So it's three different states."

Subject 2 demonstrated a similar type of reasoning when asked a similar question.

> Interviewer: "With two flip-flops, how many states would you have?"

> Subject 2: "Two. Uhh, no no. Well, you could have four different states, potentially, because, you know, $\langle 0, 0\rangle$, $\langle 1, 1\rangle$, but, at any given time, only two."

Subjects struggled to think of the state of the circuit as a single descriptor of the circuit. Instead, they thought of the state of the circuit as the state of each individual flip-flop at a moment in time.

These subjects' reasoning can be explained by considering the different contexts that might have prompted their reasoning.

- *If the circuit has two flip-flops, then the circuit can exist in* $2^2 = 4$ *states over time*. The subject may be thinking of the number of "state bubbles" that would be drawn in a state diagram.
- *If the circuit has two flip-flops, then the circuit will have two states at a given moment*. The subject may be thinking of how many state variables (flip-flop outputs) would appear in a timing diagram or next-state table.

It can be seen that the subjects' reasoning is likely to be context-dependent and fluid because there are different prominent surface features in the different contexts. This shifting of context based on surface features may explain the lack of coherence between the various definitions of state.

C. Pattern Matching and Problem Categorization

Subjects frequently tried to recall problems from lectures or tried to use simple pattern-matching schemes. Subject 5 said, "It was like five [PowerPoint] slides in a row that I memorized for this test, which just give you step by step how to do this every time." These types of comments were most common when subjects were asked to solve common design problems such as designing a sequence recognizer or using one type of flip-flop to implement a different type of flip-flop. Subjects were able to describe and incorporate the nuances of design problems—when there was a standard method for completing the problem.

When exposed to nonstandard problems such as Question 6 in Fig. 3, subjects misused pattern matching. On Question 6, seven subjects solved the problem by using a simple pattern-matching strategy that ignored the synchronizing function of the clock signal and treated the sequential circuit as a combinational circuit. Answer choice 1) on the multiple-choice form was created after subjects used this strategy. Subject 11 demonstrated this pattern matching behavior.

> Subject 11: "I kind of looked at it as, when XNOR ... so when [x and y are] both 1, [z is] 1, or when they're both 0, it's 1, ... So, basically, right here, when they're both 1, I thought it should be 1."

When solving problems, subjects typically identified a key feature of the problem and then categorized the problem based on this feature. Three problem features served as key features for problem categorization: question/answer format, visibility, and change.

Question 15. The block diagram of a synchronous finite state machine – circuit G – is shown below. Circuit G is composed of combinational logic (of unknown design) and positive edge triggered D flip-flops. Circuit G also has one input *x* and one output *z*. Suppose the timing diagram below is produced by circuit G. **On the timing diagram, indicate the times when *x* influences the state of circuit G**.

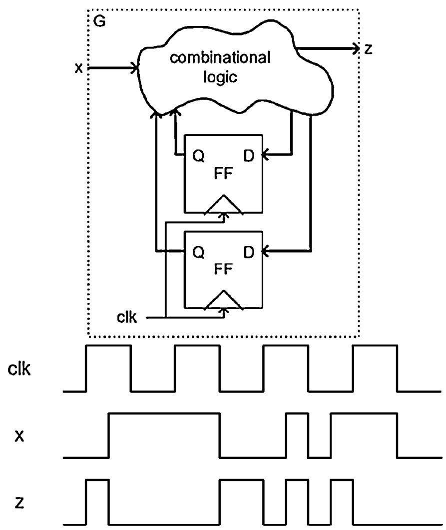


Fig. 4. Sequential circuit analysis interview question. Subjects should highlight the positive clock edges.

Subjects categorized problems by the way the question was posed or by the form of their answer. To solve Question 6 (Fig. 3) or Question 15 (Fig. 4), subjects frequently identified the timing diagram as the most important part of the problem. They ignored the circuit diagram even though the circuit diagram is the key to solving the problem. Several subjects identified these problems as "timing diagram problems" and complained about them.

> Subject 13: "[Timing diagrams] just confuse the hell out of me. I don't know what they mean, I don't get 'em at all."

These comments reveal that the subjects did not categorize problems according to a set of underlying concepts. The subjects did not recognize that they could fill in timing diagrams using the same concepts that they used to complete truth tables and next-state tables that they had easily solved earlier in the interviews. Subjects identified other problems as "state bubble" problems rather than identifying them by the function of the state machine.

Subjects also tended to focus on only the information that was clearly visible in the problem statement. Subjects frequently attempted to solve Questions 6 and 15 by only looking at the signals that were shown in the timing diagram. Few subjects noticed that the flip-flop inputs and flip-flop outputs were absent from the timing diagram. Consequently, subjects frequently associated the output z with the state. When solving Question 6, one subject even asserted that the certainty of the visible signals (x, y, and z) "*canceled out the uncertainty of the flip-flop*" signals that he could not see. When subjects focused on the visible elements of "timing diagram" problems, they used another problem feature to solve the problem—change.

People tend to focus on the changes in their environments rather than on the constants (for instance, it is easy to forget that a fan is blowing until it turns off). Subjects demonstrated an emphasis on change by only focusing on the times when signals changed values. When solving Question 15, subjects claimed that x influences z only when both signals changed at the same time. These subjects ignored the fact that x influences the value of z at all times, even when x and z are constant, because x is an input to a combinational circuit and z is an output of this circuit. This focus on change revealed another set of misconceptions about the nature of signals in a circuit.

When subjects encountered signal behaviors that did not match their mental expectations of what should happen, they either attempted to pattern-match or they claimed that the signals had will or volition. Subject 2 attributed volition to the output z in Question 15 after he failed to recognize that the clock can update the flip-flop outputs, which in turn influence the value of the circuit output z: "Now, z changes here [the second negative edge of z], kind of, out of its own volition." This misconception is reminiscent of problems that many people have when encountering computers that "do not do what I tell them to do."

V. Discussion

Subjects were able to solve standard classroom problems, and their instructors had even certified that they knew enough about digital logic to pass their introductory courses with grades of B and C. Nonetheless, when subjects were exposed to unfamiliar contexts, they manifested their misconceptions and uncertainties about the core concepts of sequential circuits. Subjects demonstrated inconsistent mental models (Section IV-A), they misunderstood how a computer stores and manipulates information (Section IV-B), and they struggled to apply appropriate tactics when solving problems (Section IV-C). In this section, the authors propose some connections between the various misconceptions in order to create a strategy for addressing these misconceptions.

A. *State is the Information Needed to Describe the Relevant Condition of a Circuit*

As discussed in Section IV-A, subjects used five different definitions for the state of a sequential circuit.

1) State is the information generated by a circuit for an external entity (circuit or user).
2) State is the information in communications between a circuit and an external entity.
3) State is all information in the circuit.
4) State is the minimum information needed to derive the values of all circuit components.
5) State is the information revealed by a single signal.

Subjects commonly shifted between these definitions as the problems and contexts changed. In some specific contexts, some of these definitions could be valid. For example, definition 1 might be perfect for counter circuits where the output of the circuit is equivalent to the state, but definition 1 is irrelevant if students are asked to relate the number of states to the number of flip-flops. Some student definitions of state, such as definition 4, are inherently adaptive and can easily bend to resemble the other definitions in the list.

Ultimately, the subjects' biggest problem is that they frequently adapted their definitions or changed their definitions to match the context of the problems they were solving. Therefore, the authors hypothesize that these five definitions can be combined into one new definition: "State is the information needed to describe the relevant condition of a circuit."

This hypothesized definition explains many problem-solving behaviors. A short case study can illustrate the utility of this definition. When subjects were asked to state the exponential relationship between numbers of flip-flops and state, they used the correct definition more than in any other problem. Subjects used the correct definition because the relevant information in the problem was the state-based components. Subjects were then presented with circuits that had varying numbers of inputs and outputs but always three flip-flops. Subjects' definitions shifted to include inputs only when the number of inputs was zero (i.e., subjects used the correct definition before they saw zero inputs, but used the *FFs Inputs* misconception after seeing zero inputs). The oddity of zero inputs drew subjects' attention away from the number of flip-flops and toward the number of inputs. Because the number of inputs seemed to be more important to the subjects' perception, their functional definition shifted.

B. Centrality of Information Encoding

Many concepts in digital logic rely upon the central concept that information in a computer is encoded into bits. Although subjects manipulate the information encoding concept in many contexts, they had significant difficulty understanding how the state of the circuit is optimally encoded in the flip-flops. Administrations of the DLCI have revealed potential connections between students' ability to encode information in three devices: state in flip-flops, addresses in random-access memory (RAM), and selection inputs in multiplexers [5].

Instructors should help students to grasp this information encoding conception and to see how it applies to these various contexts. If students learn this concept well, they may be better equipped to overcome one of their abstraction difficulties.

C. Abstraction Difficulties

When subjects were asked to solve problems like designing sequence recognizers (a circuit that tracks inputs and indicates when a desired sequence has been entered), they performed well, if not perfectly. Within these contexts, subjects were able to use a very concrete example of state to guide their selection of a state definition. State diagrams provide a physical mapping for the abstracted or encoded information contained within a circuit. Subjects seemed to find state to be an accessible concept within this physical context. In contrast, when subjects were asked to define state as a concept with only circuit diagrams as context, they demonstrated uncertainty. Subjects used physically oriented analogies, treated state as a physical object or location, and cited specific examples (such as state machines) to explain state.

Subjects demonstrated difficulties with abstraction when they were uncomfortable with, or oblivious to, hidden information. Subjects routinely skipped over the effect of the flip-flops in Questions 6 and 15 because the values of the flip-flops were not explicit in the timing diagrams. When subjects knew they did not have all the information they *wanted* to have (even if it was not needed to solve the problem), they expressed great discomfort.

D. Limitations

Although this study provides a rich description of students' misconceptions about state, the frequencies of the misconceptions in the interviews do not necessarily indicate the frequencies of these misconceptions in the general student population. For example, the identified misconceptions may not appear as frequently among, or be as detrimental to, students who have primarily been taught Moore machines (in which circuit outputs depend directly only on flip-flop states, not on circuit inputs as well) or one-hot encoding schemes (each flip-flop corresponds to one state in a state diagram). The misconception of state as circuit output is not as detrimental when working with Moore machines, and the misconception about the linear encoding of state is actually a correct conception in one-hot designs. Future studies could compare the effects of learning Mealy versus Moore machines or binary encoding versus one-hot encoding schemes.

VI. Implications and Conclusion

This paper revealed 16 misconceptions about what state is and how it is encoded. These misconceptions suggest several implications for instruction and research.

A. Implications for Instruction

The precision of the terminology in science and engineering can easily be taken for granted. Terms such as *heat* and *force* are currently precisely defined by both verbal descriptions and numerical formulas, but it took centuries for scientists to settle on these current definitions [39], [40]. Since digital logic is a relatively new discipline, some of its terminology and notation is understandably still in flux. Textbooks do not agree on the definitions of many terms. The IEEE standard definition for state is currently defunct and resembles the *FFs Inputs* misconception. If instructors and textbooks cannot agree on the precise definition of terms, it is not surprising that students possess imprecise conceptions of these terms. Instructors and researchers have the chance to learn from history and craft terminology with care over a few years rather than a few centuries. Instructors and textbook authors are encouraged to be more careful about the definitions of terms they use and to develop broadly accepted standards.

Information encoding provides one basis for constructing a unified conceptual framework for digital logic (a single concept that can be reapplied to numerous concepts in digital logic). Students should be taught information encoding and then be shown how this one concept is reapplied to the concept of state, RAM, multiplexers, decoders, and number representations.

The interviews revealed that students do not always think of flip-flops as storing bits of information and state as an abstracted/encoded piece of information. The authors suggest that only D-type flip-flops be used; other types of flip-flops should not be introduced. Since students struggle to understand the fundamental state-encoding purpose of flip-flops, introducing multiple types of flip-flops may only distract students from this fundamental purpose. Furthermore, the simplicity of the D flip-flop's functionality allows students to focus on the state-encoding purpose of one flip-flop rather than on recalling the complex functionality of the other types.

Instructors should help students switch between various levels of abstraction and representations of information. Instructors should show that state diagrams, next-state tables, timing diagrams, and circuit diagrams are convenient abstractions tools for visualizing physical circuits. Each of these abstractions has strengths and weaknesses. Students need to learn to think of these tools according to their purposes, strengths, and weaknesses.

Pedagogies that focus on reducing cognitive load have been shown to improve conceptual learning and problem solving [41], [42]. Instructors could help reduce students' cognitive loads by asking students to annotate worked examples and then explain how the instructor solved the problem. Documentation of problem-solving methods has previously been shown to increase learning and problem solving ability [43], [44]. Instructors could also reduce cognitive load by assigning design problems that constrain the methods that students are allowed to use when solving a problem. By eliminating the need to choose an appropriate methodology, students can focus on the mechanics of solving the problem. Once students have experience both in examining problem-solving strategies through worked examples and in solving highly constrained design problems, they will be better equipped to handle the difficulty of open-ended design problems [43], [44].

B. Implications for Research

This paper focused on listing the types of misconceptions that students possess. This list of misconceptions suggests three questions for additional research: 1) Why do students develop the different misconceptions? 2) How do contextual cues affect students' use of different misconceptions? 3) Do different instructional paradigms foster different misconceptions? Researchers can test the nature and cause of students' misconceptions by presenting students with similar problems whose context and presentation styles have been manipulated to test when students use their various misconceptions [45].

Education researchers should develop teaching methods that help students develop consistent mental models of state and information encoding. The effectiveness of these methods should then be assessed.

The misconceptions described in this paper are being used to construct the Digital Logic Concept Inventory. Since the DLCI is a short, multiple-choice assessment tool, it can assess the pervasiveness and robustness of these misconceptions. Future research studies with the DLCI may be able to answer the third research question and subsequently find best practices for teaching digital logic and state and sequential circuits.

Acknowledgment

J. Handzik transcribed interviews. L. Kaczmarczyk helped develop analysis protocols. The opinions, findings, and conclusions do not necessarily reflect the views of the National Science Foundation or the University of Illinois at Urbana–Champaign.

References

[1] P. S. Steif, "An articulation of the concepts and skills which underlie engineering statics," in *Proc. 34th ASEE/IEEE Frontiers Educ. Conf.*, Oct. 2004, vol. 2, pp. F1F-5–F1F-10.

[2] R. L. Miller, R. A. Streveler, M. A. Nelson, M. R. Geist, and B. M. Olds, "Concept inventories meet cognitive psychology: Using beta testing as a mechanism for identifying engineering student misconceptions," in *Proc. ASEE Annu. Conf. Expos.*, 2005, pp. 1–18.

[3] K. E. Wage, J. R. Buck, and C. H. G. Wright, "Obstacles in signals and systems conceptual learning," in *Proc. 3rd IEEE Signal Process. Educ. Workshop*, Aug. 2004, pp. 58–62.

[4] G. L. Herman, C. Zilles, and M. C. Loui, "Work in progress: Students' misconceptions about state in digital systems," in *Proc. 39th ASEE/IEEE Frontiers Educ. Conf.*, San Antonio, TX, Oct. 18–21, 2009, pp. T4D1–T4D2.

[5] G. L. Herman, M. C. Loui, and C. Zilles, "Creating the digital logic concept inventory," in *Proc. 41st Annu. ACM Tech. Symp. Comput. Sci. Educ.*, Mar. 2010, pp. 102–106.

[6] G. L. Herman, L. Kaczmarczyk, M. C. Loui, and C. Zilles, "Proof by incomplete enumeration and other logical misconceptions," in *Proc. 4th ICER*, Sydney, Australia, 2008, pp. 59–70.

[7] G. L. Herman, M. C. Loui, and C. Zilles, "Students' misconceptions about medium-scale integrated circuits," *IEEE Trans. Educ.*, 2011, to be published.

[8] G. L. Herman, M. C. Loui, and C. Zilles, "Work in progress: How do engineering students misunderstand number representations?," in *Proc. 40th ASEE/IEEE Frontiers Educ. Conf.*, Arlington, VA, Oct. 27–30, 2010, pp. T3G1–T3G2.

[9] G. L. Herman and J. Handzik, "A preliminary pedagogical comparison study using the Digital Logic Concept Inventory," in *Proc. 40th ASEE/IEEE Frontiers Educ. Conf.*, Arlington, VA, Oct. 27–30, 2010, pp. F1G1–F1G6.

[10] R. Hake, "Interactive-engagement vs traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses," *Amer. J. Phys.*, vol. 66, pp. 64–74, 1998.

[11] D. Hestenes, M. Wells, and G. Swackhamer, "Force concept inventory," *Phys. Teacher*, vol. 30, pp. 141–158, 1992.

[12] J. P. Mestre, "Facts and myths about pedagogies of engagement in science learning," *Peer Rev.*, vol. 7, no. 2, pp. 24–27, 2005.

[13] C. H. Crouch and E. Mazur, "Peer instruction: Ten years of experience and results," *Amer. J. Phys.*, vol. 69, no. 9, pp. 970–977, Sep. 2001.

[14] R. Clark, D. Feldon, J. V. Merrienboer, K. Yates, and S. Early, "Cognitive task analysis," in *Handbook of Research on Educational Communications and Technology*, J. Spector, M. Merrill, J. van Merrienboer, and M. Driscoll, Eds., 3rd ed. Mahwah, NJ: Erlbaum, 2008, pp. 1801–1856.

[15] M. D. Svinicki, *Learning and Motivation in the Postsecondary Classroom*. San Francisco, CA: Anker, 2004.

[16] D. Shinners-Kennedy, "The everydayness of threshold concepts: State as an example from computer science," in *Threshold Concepts Within the Disciplines*. Rotterdam, The Netherlands: Sense, 2008, pp. 119–128.

[17] D. Perkins, "The many faces of constructivism," *Educ. Leadership*, vol. 57, no. 3, pp. 6–11, 1999.

[18] A. A. diSessa, N. M. Gillespie, and J. B. Esterly, "Coherence vs. fragmentation in the development of the concept of force," *Cogn. Sci.*, vol. 28, no. 6, pp. 843–900, 2004.

[19] J. Clement, "Algebra word problem solutions: Thought processes underlying a common misconception," *J. Res. Math. Educ.*, vol. 33, no. 1, pp. 16–30, 1982.

[20] I. A. Halloun and D. Hestenes, "The initial knowledge state of college physics students," *Amer. J. Phys.*, vol. 53, pp. 1043–1055, 1985.

[21] I. A. Halloun and D. Hestenes, "Common sense concepts about motion," *Amer. J. Phys.*, vol. 53, pp. 1056–1065, 1985.

[22] M. T. H. Chi, P. J. Feltovich, and R. Glaser, "Categorization and representation of physics problems by experts and novices," *Cogn. Sci.*, vol. 5, pp. 121–152, 1981.

[23] M. T. H. Chi, J. D. Slotta, and N. deLeeuw, "From things to processes: A theory of conceptual change for learning science concepts," *Learn. Instruct.*, vol. 4, pp. 27–43, 1994.

[24] S. B. Merriam, *Qualitative Research: A Guide to Design and Implementation*. San Francisco, CA: Jossey-Bass, 2009.

[25] M. M. Mano and C. Kime, *Logic and Computer Design Fundamentals*, 4th ed. Upper Saddle River, NJ: Prentice-Hall, 2008.
[26] K. A. Ericsson and H. A. Simon, *Protocol Analysis: Verbal Reports as Data*. Cambridge, MA: MIT Press, 1984.
[27] S. Kvale, *Interviews: An Introduction to Qualitative Research Inquiry*. Thousand Oaks, CA: Sage, 1996.
[28] A. Strauss and J. Corbin, *Basics of Qualitative Research*. Thousand Oaks, CA: Sage, 1998.
[29] M. B. Miles and M. Huberman, *Qualitative Data Analysis: A Sourcebook of New Methods*. Thousand Oaks, CA: Sage, 1984.
[30] J. D. Irwin and D. V. Kerns, Jr., *Introduction to Electrical Engineering*. Upper Saddle River, NJ: Prentice-Hall, 1995.
[31] F. Vahid, *Digital Design*. Hoboken, NJ: Wiley, 2006.
[32] S. Brown and Z. Vranesic, *Fundamentals of Digital Logic With VHDL Design*. New York: McGraw-Hill, 2009.
[33] H. Hellerman, *Digital Computer System Principles*. New York: McGraw-Hill, 1967.
[34] J. F. Wakerly, *Digital Design: Principles and Practices*. Upper Saddle River, NJ: Prentice-Hall, 2006.
[35] D. D. Givone, *Digital Principles and Design*. New York: McGraw-Hill, 2003.
[36] A. B. Marcovitz, *Introduction to Logic and Computer Design*. New York: McGraw-Hill, 2008.
[37] E. O. Hwang, *Digital Logic and Microprocessor Design With VHDL*. Toronto, ON, Canada: Thomson, 2006.
[38] K. Goldman, P. Gross, C. Heeren, G. Herman, L. Kaczmarczyk, M. C. Loui, and C. Zilles, "Setting the scope of concept inventories for introductory computing subjects," *Trans. Comput. Educ.*, vol. 10, no. 2, pp. 5:1–5:29, 2010.
[39] S. Carey, "Reorganization of knowledge in the course of acquisition," in *Ontogeny, Phylogeny, and Historical Development*. Norwood, NJ: Albex, 1988, pp. 1–27.
[40] M. P. Crosland, *Historical Studies in the Language of Chemistry*. New York: Dover, 1962.
[41] J. Sweller and G. A. Cooper, "The use of worked examples as a substitute for problem solving in learning algebra," *Cogn. Instruct.*, vol. 2, pp. 59–89, 1985.
[42] P. A. Kirschner, J. Sweller, and R. Clark, "Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching," *Educ. Psychol.*, vol. 41, no. 2, pp. 75–86, 2006.
[43] R. Dufresne, W. Gerace, P. Hardiman, and J. Mestre, "Constraining novices to perform expert-like problem analyses: Effects on schema acquisition," *J. Learn. Sci.*, vol. 2, pp. 307–311, 1992.
[44] W. J. Leonard, R. J. Dufresne, and J. P. Mestre, "Using qualitative problem-solving strategies to highlight the role of conceptual knowledge in solving problems," *Amer. J. Phys.*, vol. 64, pp. 1495–1503, 1996.
[45] G. L. Herman, L. Kaczmarczyk, M. C. Loui, and C. Zilles, "Discovering students' misconceptions in boolean logic," *Trans. Comput. Educ.*, 2011, submitted for publication.

Geoffrey L. Herman (S'08) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign in 2011.

He is a graduate affiliate with the Center for Teaching Excellence, University of Illinois at Urbana–Champaign. His research interests include understanding how students develop conceptual understanding, audio signal processing, and the development of intelligent online tutoring and lecture systems.

Dr. Herman is a Mavis Future Faculty Fellow. He was awarded an ASEE Educational Research and Methods Division Apprentice Faculty Grant and has won the Olesen Award for Excellence in Undergraduate Teaching in the Department of Electrical and Computer Engineering, University of Illinois at Urbana–Champaign.

Craig Zilles (M'10) received the Ph.D. degree in computer science from the University of Wisconsin–Madison in 2002.

He is an Associate Professor with the Computer Science Department, University of Illinois at Urbana–Champaign. Prior to his work on computer architecture and compilers, he developed the first algorithm that allowed the rendering of arbitrary three-dimensional polygonal shapes for haptic interfaces (force-feedback human-computer interfaces). He holds seven patents. His current research focuses on computer science education and the interaction between compilers and computer architecture.

Dr. Zilles and his work have been recognized by a Best Paper Award at ASPLOS and an *IEEE Micro* Top Pick. For his teaching and scholarship of teaching and learning, he was the recipient of the IEEE's Mac Van Valkenburg Award and the University of Illinois Rose and Everitt awards.

Michael C. Loui (S'75–M'80–SM'87–F'06) received the Ph.D. degree in computer science from the Massachusetts Institute of Technology, Cambridge, in 1980.

He is a Professor of electrical and computer engineering and University Distinguished Teacher-Scholar with the University of Illinois at Urbana–Champaign. He was Associate Dean of the Graduate College at the University of Illinois from 1996 to 2000. He directed the Theory of Computing program at the National Science Foundation, Washington, DC, from 1990 to 1991. He serves as Executive Editor of *College Teaching* and a member of the Editorial Board of *Accountability in Research*. His interests include computational complexity theory, professional ethics, and the scholarship of teaching and learning.

Prof. Loui is a Carnegie Scholar.