# Students' Misconceptions About Medium-Scale Integrated Circuits

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

***Abstract*—To improve instruction in computer engineering and computer science, instructors must better understand how their students learn. Unfortunately, little is known about how students learn the fundamental concepts in computing. To investigate student conceptions and misconceptions about digital logic concepts, the authors conducted a qualitative interview-based study. In the interviews, students verbalized their thinking while they solved digital logic problems. The interviews were analyzed to identify students' misconceptions. This paper presents five classes of students' misconceptions and methodological weaknesses concerning medium-scale integrated (MSI) circuits. These misconceptions were used to create a multiple-choice conceptual assessment, called a concept inventory, that is being used to test for the prevalence of the misconceptions discovered through the interviews. Although the misconceptions documented in the paper are primarily intended for the construction of the concept inventory, they can also point to potential ways to improve instruction.**

***Index Terms*—Boolean algebra, combinational circuits, computer science education, integrated circuit, logic circuits.**

## I. Introduction

DURING the design process, engineers use systems and components whose inner designs are unknown and must construct new systems from familiar systems and components. The ability to use and construct abstracted components is essential, but often difficult for students to learn. In digital logic design, students first encounter abstracted circuits when they are taught medium-scale integrated (MSI) circuits. MSI circuits with various functions and purposes can be constructed from the basic components (logic gates) and by using tools (truth tables and Boolean algebra) that students have previously learned. By understanding how students learn about MSI circuits, researchers have a new perspective on understanding how students learn more complicated design techniques and how students learn to think on multiple levels of abstraction. If students do not have the right conceptions about MSI circuits, they cannot be expected to use their knowledge reliably when they analyze or create designs.

This paper presents a study that elicits student misconceptions about MSI circuits through one-on-one interviews. This study is also part of a larger corpus of research to develop a concept inventory (CI)—a multiple-choice test that provides a reliable and validated measure of student conceptual understanding—to assess quickly what misconceptions are present in a student population [1]. Other papers document the common student misconceptions about number representations, Boolean variables and expressions, and sequential circuits and states [2]–[6]. This study will determine the misconceptions that will then be used to create multiple-choice questions about MSI components for a digital logic concept inventory.

Documentation of students' misconceptions can aid an instructor when their knowledge of students' misconceptions could be inaccurate or incomplete. New instructors, in particular, can benefit from knowing, before they begin teaching, what difficulties their students might find with the subject material. This study, therefore, aimed to explore students' misconceptions systematically, using standard qualitative research methods.

With the knowledge of common student misconceptions and a CI to assess these quickly, instructors can intelligently adjust their teaching methods to maximize students' learning. Electrical and computer engineering education could be revolutionized in the same way that physics education was by the Force Concept Inventory (FCI) [7], [8]. The FCI prompted the adoption of interactive engagement teaching methods and peer instruction by showing that these methods measurably reduced student misconceptions [7], [9], [10]. A CI creates the impetus for change by providing a way to rigorously, objectively, and empirically compare conceptual learning. When instructors and researchers can compare conceptual learning, they can compare the effectiveness of different teaching methods.

Students' misconceptions of, and methodological weaknesses in dealing with, MSI circuits are presented both to document the steps taken to create the digital logic concept inventory (DLCI) and to provide a unique insight into students' mental models of these circuits. Because the primary goal of the interviews was to elicit student misconceptions, the results focus primarily on misconceptions and secondarily on methodological weaknesses as they became apparent. Knowledge of these misconceptions and methodological weaknesses may have immediate implications for improving digital logic instruction. This paper does not present or evaluate a teaching method, but it provides a theoretical basis for developing future methods.

## II. Background

Misconceptions research has shown that students manifest different misconceptions when asked to use a concept in dif-

Manuscript received September 07, 2010; revised November 11, 2010; accepted December 21, 2010. Date of publication January 20, 2011; date of current version November 02, 2011. This work was supported by the National Science Foundation under Grant DUE-0618589. The opinions, findings, and conclusions do not necessarily reflect the views of the National Science Foundation or the University of Illinois at Urbana–Champaign.

G. L. Herman and M. C. Loui are with the Department of Electrical and Computer Engineering, 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.2011.2104361

fering contexts or problems [11], [12]. These research findings have shown that a student's understanding of a concept is often situationally based and can easily shift. Student misconceptions are not entirely unpredictable, though. Misconceptions are often rooted in the immediately observable features of common problems [13]. When students focus on these surface features, they classify concepts poorly. For example, students often think of electrical current as an object, rather than as a constraint-based process [14].

Research on student misconceptions has traditionally focused on misconceptions about physical phenomena that result from learning prior to formal instruction. These types of misconceptions can be referred to as *preconceptions*. Addressing and correcting preconceptions is difficult because these preconceptions are deeply rooted in experience and common-sense intuition [15], [16]. For example, a small child may conceive of the earth as a flat rectangular object. When the child is told that the earth is round, the child will often believe that the earth is round like a pancake rather than a sphere [17].

Students have preconceptions upon entering their first digital logic course. They treat Boolean variables as physical objects rather than as representations of the truth of a condition [2], and they treat state as a physical location or quantity rather than an abstract encoding [6].

MSI circuits are a special concept in digital logic and misconceptions research because students are unlikely to have preconceptions about them. Because students have no preconceptions, it is likely that most misconceptions about MSI circuits result from instruction. Consequently, students' misconceptions about MSI circuits might not be as deeply ingrained as physical preconceptions, and informed changes in instruction may have a greater impact on student learning.

## III. Methodology

This section describes the selection of interview subjects, the interview protocol, and the terminology used for the remainder of the paper. This section is adapted from previous publications [2], [6].

### A. Subjects

In Spring 2008, Fall 2008, and Spring 2009, the authors interviewed nine undergraduate students, six undergraduate students, and 11 undergraduate students, respectively, at the University of Illinois at Urbana–Champaign. More interview subjects were added until interviews ceased to reveal more misconceptions. All students were recruited from two large three-credit digital logic courses with simulation labs, one in the Department of Computer Science and one in the Department of Electrical and Computer Engineering. Both courses used the same textbook [18], and each had about 200 students per semester. All interviewed students were traditional age (18–22) undergraduates majoring in computer science, electrical engineering, or computer engineering who had just completed one of the digital logic courses and had earned grades between 1.7 and 3.3 on a 4.0 maximum scale. These students were selected because their understanding was likely to be less complete (and thus more likely to have misconceptions) than students with grades greater than 3.3. Pilot interviews confirmed these expectations.

### B. Interview Process

Each student was interviewed individually 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 [19]. Prior to the interview, students were briefed on the study's goal of understanding how students reason about 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 [19].

Each interview was recorded using a document camera and microphone. The audio tracks of the interview recordings were transcribed verbatim. Quotations from the transcripts 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 statement. For example, the quotation "Subject 2: Um, a multiplexer is an electronic component that [pause] let's see [pause] it basically takes [pause] how would I describe this? [pause] it takes a couple wires in, it takes like $n$ wires in and spits out 2 to the $n$ wires?" will be presented as "Subject 2: A multiplexer is an electronic component [pause] how would I describe this? [pause] it takes a couple wires in, it takes $n$ wires in and spits out 2 to the $n$ wires?"

Students were paid for their participation, and all students gave written consent to be interviewed under IRB approval (UIUC no. 07026).

### C. Interview Questions

Students in this study were interviewed for 1 h about many concepts in digital logic design. Due to time constraints, each student was interviewed on only a portion of all interview questions. Results from administrations of the DLCI are also presented [1]. The DLCI is a multiple-choice assessment tool based on the misconceptions documented in the misconceptions research [2]–[4], [6]. The items from the DLCI were previously administered and analyzed via statistical tests. These tests demonstrated that the DLCI items reliably and validly measured the presence of student misconceptions [1].

Students interviewed were asked a slightly different set of questions in each semester of the study (or the questions were simply formatted differently) based on the analysis and findings from the previous round of interviews. Spring and Fall 2008 students were asked, "What is a multiplexer (without using a truth table)?," "What is a decoder (without using a truth table)?," "Why/where would you use a multiplexer/decoder in a circuit?" They were asked to solve the problems in Figs. 1 and 2. They were asked to build an MSI circuit that could implement the function $f(x,y,z)=\bar{x}y+xz$. Finally, they were asked to calculate the number of address lines, the number of data lines, and the storage of a random-access memory (RAM)/read-only memory (ROM).

Students interviewed during Spring 2009 were asked to solve the multiple-choice MSI questions from the alpha version of the DLCI [1]. These multiple-choice questions were adapted from the questions and responses of the students during the earlier interviews. The remaining interview questions and acceptable answers to these questions will be introduced when needed.

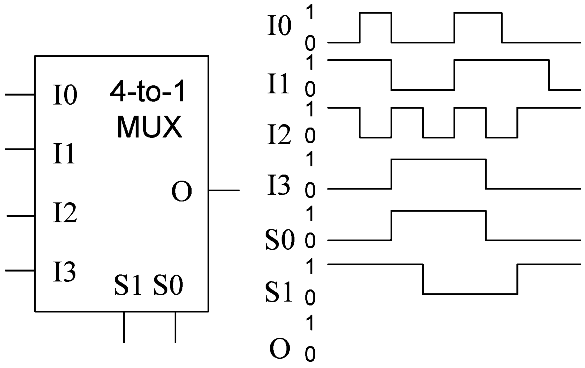


Fig. 1. Multiplexer analysis question that uses a timing diagram.

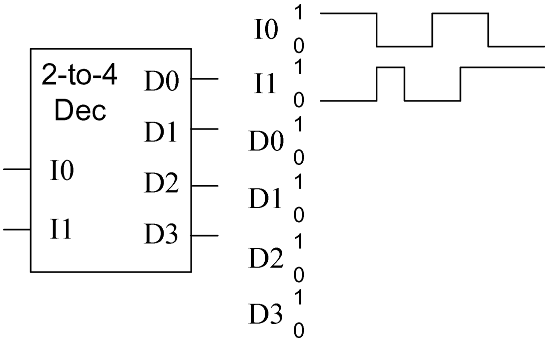


Fig. 2. Decoder analysis question that uses a timing diagram.

### D. Data Analysis

Interviews were analyzed using a grounded theory approach as described by Kvale [20], Strauss and Corbin [21], and Miles and Huberman [22]. The analysis protocol is described in depth in previous works [2], [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, 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."

The following abbreviations are used: multiplexer as MUX, binary decoder as decoder or DEC, read-only memory as ROM, and random-access memory as RAM. For the sake of this paper, each of these components is classified as an MSI circuit [18], [23], [24]. While a RAM is not an MSI circuit, RAM is a concept of interest because of its similarities to an MSI component—ROMs (addressing, data words, etc.).

Multiplexer selection inputs, decoder inputs, and ROM/RAM address lines are classified together as *encoded bits* because they represent design information in a binary code. Multiplexer data inputs, multiplexer data outputs, decoder outputs, and ROM/RAM data lines are classified as *unencoded bits* because the bits are not interpreted by the designer as a binary code.

## IV. Results and Themes

This section presents the misconceptions and themes revealed by the interviews and DLCI. The data from the interviews were used to create a rich description of students' conceptions and misconceptions, and the numerical data from the DLCI were used to provide an initial estimate of the pervasiveness of these conceptions and misconceptions.

### A. Recall and Routine Analysis

The majority of subjects could recall the general functionality of a multiplexer and a decoder and all subjects, except Subject 6, correctly completed the timing diagrams in Figs. 1 and 2.

When asked to describe the purpose or functionality of a multiplexer or a decoder, subjects were able to recall the functionality of a multiplexer much better than that of a decoder. All but one subject could describe the basic functionality of a multiplexer, and only two subjects could not say when to use a multiplexer. Two subjects could not remember the term "decoder," and three subjects could neither articulate the basic functionality of a decoder nor explain when to use a decoder.

### B. MSI Concept Aliasing

Excluding the subject who could not remember the terms "multiplexer" and "decoders," all but one subject stated that multiplexers and decoders are opposites. These responses were incorrect because the functional and structural opposite of a multiplexer is a demultiplexer, and the functional and structural opposite of a decoder is an encoder. Subjects described multiplexers and decoders as opposites based on structure or functionality. One subject thought that all multiplexers had $2^n$ data inputs and $n$ outputs with no selection inputs. A different subject described a decoder as a circuit with one data input, many outputs, and selection inputs. Other subjects simply described a multiplexer when asked to describe a decoder, or vice versa. The following quotations demonstrate these misconceptions.

Subject 2 initially described a decoder when asked about a multiplexer:

> "A multiplexer is an electronic component [pause] how would I describe this? [pause] it takes a couple wires in, *it takes $n$ wires in and spits out $2^n$ wires*?"

Subject 4 described a decoder as the structural opposite of a multiplexer:

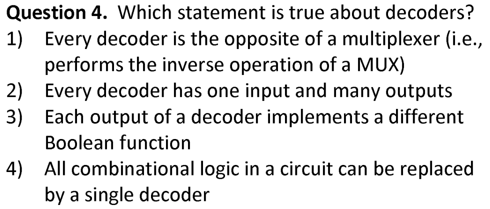
> "I'm thinking opposite of a MUX, so that makes me think *one to many*."

Subject 8 completely confused himself about multiplexers and decoders:

> Subject 8: "A multiplexer was the opposite of [a decoder] where *you take $2^n$ inputs and you get $n$ outputs*. Hold on. I got them both mixed up, didn't I?"
>
> Interviewer: "You tell me."
>
> Subject 8: "Decoder MUX. Oh man. Umm, decoder [pause] multiplexer [long pause]."

Subject 1 described a ROM ("a truth table in circuit form") when asked to describe a decoder.

Fig. 3 shows the DLCI item that tests for these types of misconceptions. When answering this item, 40% of students chose answers that describe a decoder as the functional (option 1) or physical opposite (option 2) of a multiplexer, and 11% of



**Question 4.** Which statement is true about decoders?

1) Every decoder is the opposite of a multiplexer (i.e., performs the inverse operation of a MUX)
2) Every decoder has one input and many outputs
3) Each output of a decoder implements a different Boolean function
4) All combinational logic in a circuit can be replaced by a single decoder

Fig. 3. DLCI item that probes students' understanding about the relationship between multiplexers and decoders.

students chose the answer that describes a decoder as a ROM (option 4).

### *C. Synchronous Combinational Circuits*

Three subjects thought that MSI components have synchronous behaviors like flip-flops. For instance, when analyzing Fig. 2, Subject 8 initially claimed that the outputs of the decoder could change only when the clock transitioned from 0 to 1. This subject also subtly claims that the outputs of a decoder can be initialized to zero:

> "Since you're starting with $\langle 0, 1\rangle$, D1 would be [1]. But if we assume that the values are all 0 starting out, you just change the signal at the beginning of this clock cycle."

### *D. Encoding Difficulties*

Subjects had several misconceptions about signals used to encode design information in MSI circuits (for example, address inputs and selection inputs). Encoded information signals proved especially difficult for subjects when they had to manipulate these signals apart from basic circuit analysis problems, such as Figs. 1 and 2.

*1) Convention Breaking and Bit Swapping:* When solving the circuit analysis problems in Figs. 1 and 2, subjects would break from typical conventions and rename variables $S_0$ and $S_1$ to be $x$ and $y$ because they did not like subscripts. After renaming the variables, some subjects swapped the order of the multiplexer selection inputs or the inputs of the decoder.

*2) Inputting All Literals Into a Decoder:* During the interviews, subjects were asked to use a multiplexer or decoder and up to one logic gate of their choice to implement the function $f(x, y, z) = \bar{x}y + xz$ (the Boolean expression that defines a 2-to-1 multiplexer). When solving this question, Subject 1 attempted to assign all literals ($x$, $\bar{x}$, $y$, and $z$) as inputs into the decoder rather than just the variables ($x$, $y$, and $z$):

> "I would have to use a fairly large decoder. I would have to use $X\bar{X}YZ$, which would give me 16 outputs [draws]."

*3) Forgotten Selection Inputs:* Several subjects drew multiplexers without, or with too few, selection inputs. This misconception was accompanied by the misconception that decoders and multiplexers are opposites of each other.

> Subject 7: "If we have a multiplexer, either 8 to 1 or 1 to 8, [this circuit is] an 8 to 1 [pause] If you send in eight things then it spits out one. Probably based on [pause] I can't remember what it's based on. Do you send in a second code?"

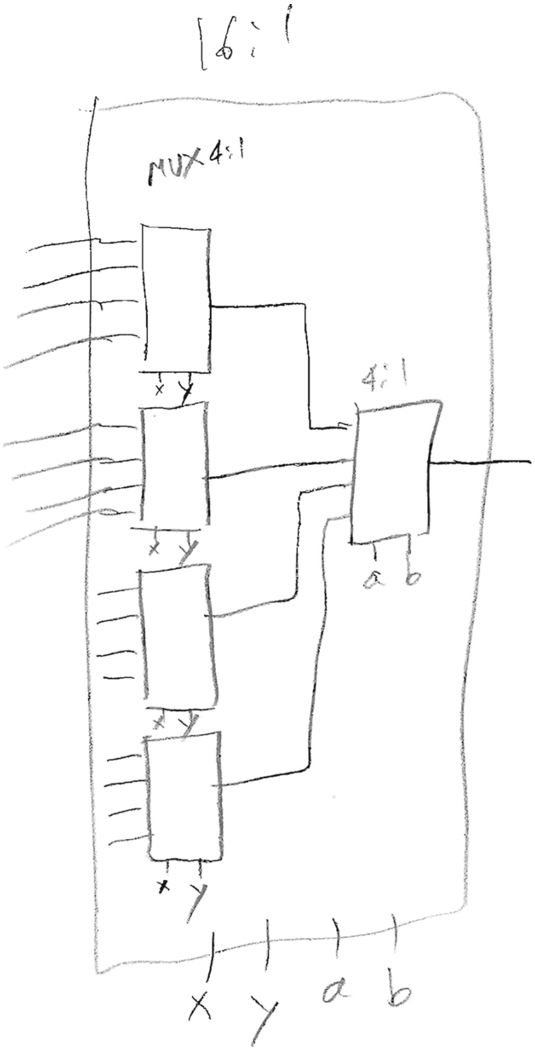


Fig. 4. Subject matches the selection inputs to the multiplexer structure and ignores the significance of the encoding scheme of the bits.

*4) Difficulties With Constructing Large Multiplexers:* When creating a 16-to-1 multiplexer from 4-to-1 multiplexers, most subjects arranged the 4-to-1 multiplexers correctly, but many did not correctly assign the selection inputs.

*Assignment by Proximity:* In Fig. 4, the subject arranged the 4-to-1 multiplexers correctly relative to each other, but then assigned the selection inputs based on physical proximity/structure rather than on the selection input encoding scheme. The subject ordered the selection inputs as $\langle x, y, a, b\rangle$ and indicated that the lower order data input bits were at the top of the multiplexer structure. Actually, for the ordering $\langle x, y, a, b\rangle$ of the selection inputs, the $x$ and $y$ selection inputs should be assigned to the right-most multiplexer, and the $a$ and $b$ selection inputs should be assigned to the four multiplexers on the left.

*Assignment by Counting Multiplexers:* In Fig. 5, the subject observed the *four* selection inputs of the 16-to-1 multiplexer and the *four* 4-to-1 multiplexers on the left, so he assigned one selection input to each 4-to-1 multiplexer. Consequently, the subject also maintained that 4-to-1 multiplexers have only one selection input for much of the interview. The subject was troubled when he realized that he did not have a fifth selection input for the fifth 4-to-1 multiplexer, but he never relinquished the

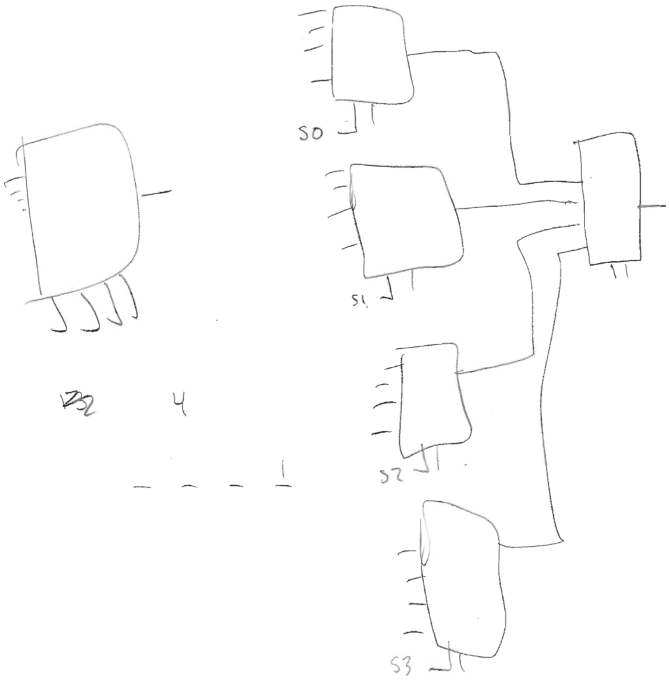


Fig. 5. Subject shows that he knows that a 16-to-1 multiplexer has four selection inputs, but assigns the selection inputs based on physical structure rather than an encoding scheme.

belief that each 4-to-1 multiplexer should receive a different selection input.

*Assignment by Counting Selection Inputs:* Subject 8 never considered how many selection inputs a 16-to-1 multiplexer should have as an independent circuit element. The subject determined that the 16-to-1 multiplexer would have 10 selection inputs based solely on the structure of the 4-to-1 multiplexers.

> Subject 8: "Well for [the rightmost multiplexer], I have four outputs [from the other multiplexers] and I have to select one from four for my final output. I'm gonna need one two three four five six seven eight nine ten, ten input bits for the chip selects. The two most significant ones go [to the right multiplexer's selection inputs]."

Results from the DLCI have shown that the assignment of selection inputs to a set of multiplexers that compose a larger multiplexer is extremely difficult. Two DLCI items test this concept. The first item asks students to assign selection inputs in a 4-to-1 multiplexer composed of 2-to-1 multiplexers (Fig. 6). For this item, 9% of students chose answers based on *assignment by proximity* (option 1), 9% chose option 5 (a variation on *assignment by proximity*), and 5% chose options 3, 4, and 6, which are based on *assignment by counting*. Follow-up interviews revealed that most subjects answered this question by assigning values for the selection and data inputs (say, by setting $\langle x, y\rangle = \langle 0, 1\rangle$ and checking if the output was what they expected), but could not explain why certain selection input assignment arrangements were invalid. To minimize assign and evaluate tactics, a similar multiplexer DLCI item was created (Fig. 7). This item is the most difficult item on the DLCI. About 20% of students chose *encode-by-proximity*-based answers (options 3, 4, and 6), and 40% of students have chosen *encode-by-counting*-based answers (options 5 and 7).

*5) ROM/RAM Addresses Versus Data:* Subjects were also asked several questions about a 32 M × 32 RAM chip.

1) How many data lines does the RAM chip have?
2) How many address lines does the RAM chip have?
3) How would you calculate the total number of bits the RAM can hold?
4) What is the size of a word in this RAM?
5) How many words does this RAM store?

Though subjects correctly answered questions 3–5, they struggled with questions 1 and 2. Subject 5 thought that there needs to be one address line per address, and he did not know that addresses are indexed by a binary code.

> Subject 5: "There's 32 times $2^{20}$ addresses. Each with 32 bits [pause] I think."
> Interviewer: "OK, what was your answer to (1)?"
> Subject 5: "How many data lines? I think you need one line for each address. So, I think it's $2^{25}$."
> Interviewer: "How about for the next question?"
> Subject 5: "[subject reads (2)] I think it's the same [$2^{25}$]."
> Interviewer: "Can you tell me what's the difference between an address line and a data line?"
> Subject 5: "The address line selects like the block of RAM. And the data line would be for writing to the RAM. Or for an output from it I think."

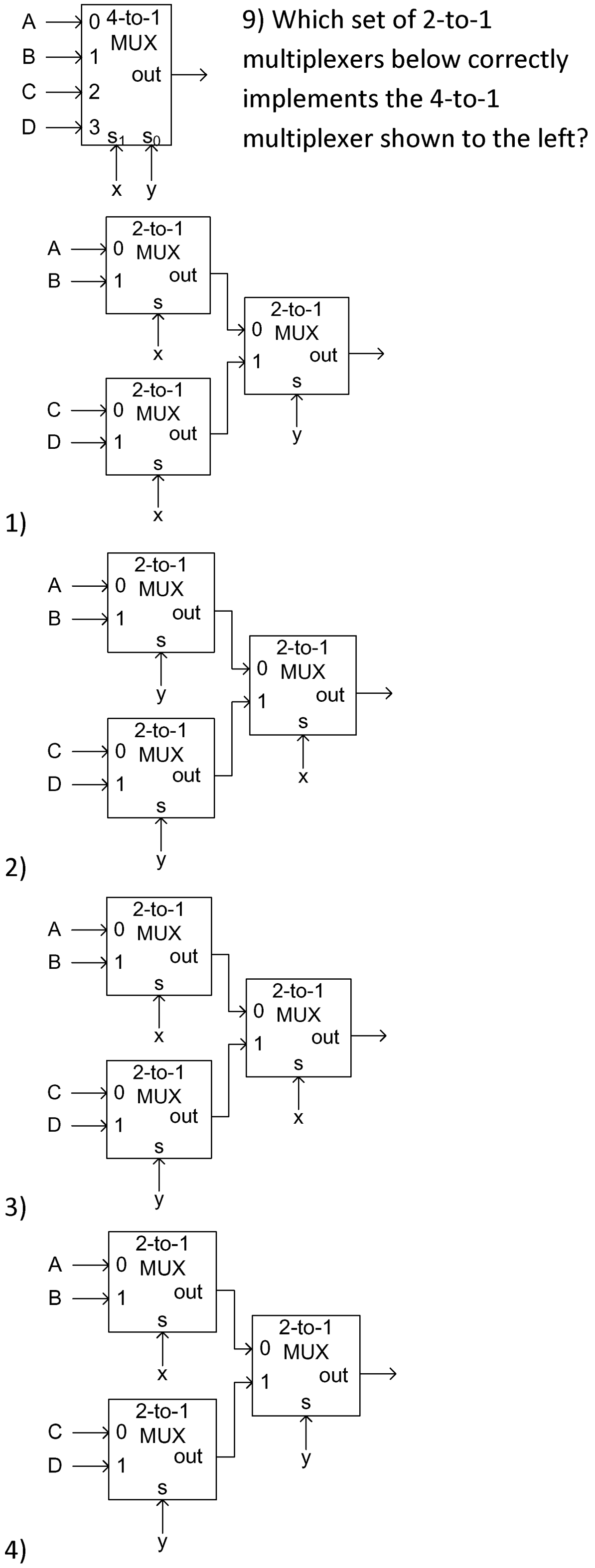
Results from the DLCI have confirmed that students struggle with the difference between address and data lines. The most common errors reveal that many students believe that the number of address lines is the same as the number of addresses (6% of students) or that there are $\log_2 d$ data lines, where $d$ is the number of bits per word (28% of students).

### E. Methodological Weaknesses

Subjects used processes that either hindered their ability to solve problems or allowed them to correctly solve problems whose underlying concepts they did not fully understand. Subjects relied on problem recall and explicitly referred to problems that they had solved in homework. The heavy reliance on problem recall showed that students favored matching their solution strategies to previous solutions rather than working from foundational conceptual knowledge.

Subjects frequently used the "assign and evaluate" method of analyzing circuits by assigning a series of test cases to a circuit and checking if the output of the circuit was what they expected. This method is often appropriate and powerful. Many subjects used assign and evaluate to solve the problem in Fig. 6 despite their incomplete knowledge. Subjects often misapplied this method by using "*Proof by Incomplete Enumeration* [2]." In *Proof by Incomplete Enumeration*, subjects check one or two cases to validate a hunch. If the one or two cases do not contradict their hunch, they decide that their initial hunch was correct. For example, when solving Fig. 6 a subject might initially believe that options (1) and (2) are correct. He would then check the $\langle 0, 0\rangle$ case and find that both circuits work for this case and select option (5) because both circuits worked for this one case.

As described earlier in Section IV-D, subjects relied heavily on the physical arrangement of circuit components to determine how to assign selection inputs in constructed multiplexers. Subjects assigned selection inputs based on the number of visible



9) Which set of 2-to-1 multiplexers below correctly implements the 4-to-1 multiplexer shown to the left?

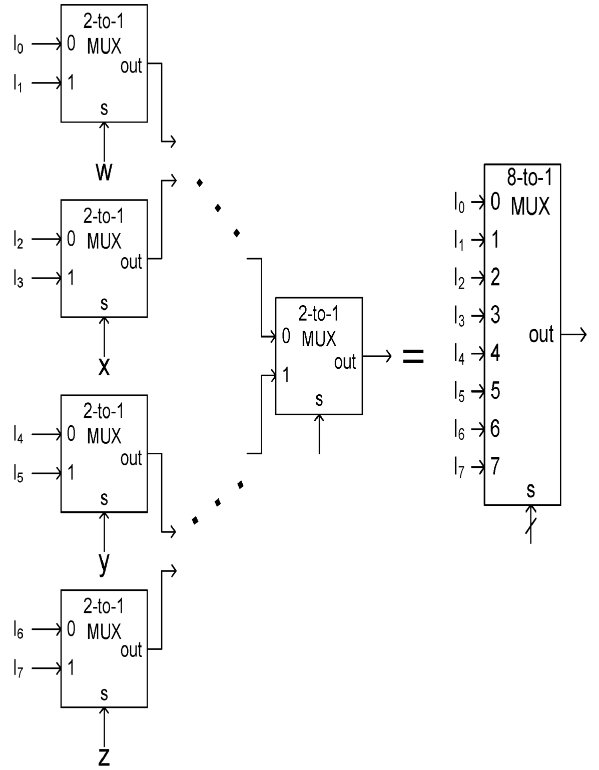
5) either (1) or (2) is valid

6) either (3) or (4) is valid

Fig. 6. Initial multiplexer selection inputs MSI item from the DLCI. Correct answer is (2).

selection inputs, on the proximity of outputs to inputs, and on the physical layout of the multiplexers.

**Question 11.** An 8-to-1 multiplexer (on the right) is implemented using 2-to-1 multiplexers. A portion of the implementation is shown below (on the left). The eight data inputs are ordered from $I_0$ to $I_7$. The selection input bits are ordered from least to most significant beginning with $s_0$. Assuming that all other select bits are chosen correctly with respect to your choice, which assignment of select bits for ⟨w,x,y,z⟩ will allow the implementation to function correctly?



1) $\langle w,x,y,z\rangle = \langle s_0,s_0,s_0,s_0\rangle$
2) $\langle w,x,y,z\rangle = \langle s_1,s_1,s_1,s_1\rangle$
3) $\langle w,x,y,z\rangle = \langle s_2,s_2,s_2,s_2\rangle$
4) $\langle w,x,y,z\rangle = \langle s_3,s_3,s_3,s_3\rangle$
5) $\langle w,x,y,z\rangle = \langle s_0,s_1,s_2,s_3\rangle$
6) As long as $w, x, y$, and $z$ receive **the same** select bit, the implementation can function correctly
7) As long as $w, x, y$, and $z$ receive **different** select bits, the implementation can function correctly

Fig. 7. Revised version of the previous DLCI item to eliminate the ability to "assign and evaluate." Correct answer is (1).

## V. Discussion

This study revealed that most students can correctly determine the outputs of an MSI component when given its inputs. Students could determine the outputs of these components even while possessing misconceptions about them. These misconceptions appear when students either designed MSI components or designed circuits with MSI components. These misconceptions likely form because students do not properly associate the functionality of MSI circuits with their purpose and because students struggle with the concept of information encoding.

### A. Information Encoding

Previous research has shown that students struggle to understand how information is encoded in bits. Students struggle to understand that flip-flops can each store one bit and that every unique configuration of bits encodes a unique state [6]. Administrations of the DLCI have also revealed that there are statistically significant correlations between students' understanding of how to encode state in bits, address locations in RAM, and assign bits in a multiplexer [1].

All MSI components discussed in this paper use encoded bits to determine their operation. If a student understands selection inputs, the ability to understand address lines and decoder inputs should be an easy next step. However, the subjects could demonstrate mastery of one concept, but not necessarily the others.

Instructors need to help students grasp this encoding conception and help them distinguish between components whose functions depend on interpreting some sets of inputs as the binary encoding of a number and other inputs that do not rely on such an interpretation. If instructors can help students learn information encoding, they might better apply the concept to new situations.

### B. Emphasis on Visible Features Over Purpose to Reduce Cognitive Load

Given that students do not have preconceptions about MSI circuits, the authors propose that cognitive load theory (CLT) offers insights into how students' misconceptions form.

When learning a new topic, students must be able to recall and manipulate a large amount of new information. Information in the brain is stored in either the nimble, but limited, working memory (similar to RAM in a computer system) or the large, but difficult to access, storage memory (similar to a hard drive) [25]. The information that a person manipulates in their working memory is called their *cognitive load* [26]–[28].

Different tasks have different levels of cognitive load. Adding two numbers presents a low cognitive load, while constructing a large circuit presents a high cognitive load. As a person gains more experience with a task, s/he lowers his/her cognitive load by "chunking" information together into meaningful, larger units of information [29] (not unlike how gates and wires are "chunked" together to create a multiplexer). This chunking process can occur through repetition, developing schema (strategies), and instruction.

The chunking of information into strategic units allows experts to access their knowledge more effectively than novices. Novices have much of the same knowledge that experts have, but this information has not yet been properly organized. As an example, when shown a chess board arranged "mid-game," chess experts can memorize the location of the pieces faster than novices. When the pieces are randomly placed, chess experts cannot memorize the location of the pieces faster than novices. Researchers discovered that chess experts memorized the strategies in play rather than the pieces [29].

Engineers perhaps created MSI circuits to ease the design process, and as a by-product they reduced their cognitive load. MSI circuits ease cognitive load by providing a toolbox of commonly used components, reducing visual clutter, and allowing the designer to focus on purpose, function, and strategy (like chess experts) rather than on the implementation of these circuits.

Perhaps because MSI circuits were all created to ease design, many textbooks present MSI circuits as a single unit [18], [23], [24]. As CLT explains, students naturally try to chunk these circuits together to make them easier to remember. Since MSI circuits all share similar visible features (boxy circuits with many inputs and outputs) but have different purposes, the similarity of visible features may induce students to chunk the circuits together based on the visible features rather than by purpose. Thus, a student might categorize a multiplexer as a box with many inputs and one output rather than as a circuit that selects one data input from many to send to the output. This type of chunking explains why students make the faulty associations seen in Section IV-B and why subjects thought that MSI components were synchronous components like flip-flops (boxy circuits with many inputs and outputs).

Because students chunk by visible features, they fill their working memory with lists of visible features (such as the number of inputs or number of outputs) rather than a single function (for instance, a multiplexer is a selector) and thus carry an unnecessarily high cognitive load. As the complexity—the cognitive load—of the problem increased, subjects forgot their list of features—thus perhaps failing to include selection bits on a multiplexer—due to a lack of space in their working memory. When subjects could not recall the features of MSI components, they had to rely on the visible structure of the problem to determine a solution and resorted to methods such as *assignment by proximity* and *assignment by counting*.

## VI. Conclusion

This study revealed that students' knowledge of MSI components is unreliable apart from recalling the basic function of MSI components. Students exhibit misconceptions about MSI components once they try to apply their basic knowledge to novel or complicated problems. These misconceptions result from the need to manage a lot of information that has not been properly organized in the students' minds.

### A. Implications

Engineering instructors commonly emphasize teaching design; instructors assign complicated design problems to give students exposure to the design process. When asked to solve novel, complicated problems, students suffer under a high cognitive load. In these situations, students resort to coping mechanisms such as relying on the physical arrangement of circuit components. These coping mechanisms cause them to make faulty associations that can develop into misconceptions. To teach design, instructors should create design problems that reduce cognitive load and focus on teaching the purpose of, and strategies associated with, MSI circuits.

Chess instruction may offer insight into how to create these types of assignments. Chess can be taught by playing an endless number of games (similar to using many nontargeted design problems), but chess is more commonly taught by showing chess novices in-game scenarios [30]. For example, a chess exercise may show a board with only a few pieces remaining, and the novice is instructed to find a way to checkmate the opponent in three moves. This type of exercise has a low cognitive load (a low number of pieces and a definitive objective), and it allows the novice to focus on learning the strategies associated with the chess pieces.

Some suggestions follow for problems that can reduce cognitive load and focus on teaching purpose and strategy.

1) Ask students to build an 8-to-1 multiplexer out of 2-to-1 multiplexers, but provide the correct wiring and circuit layout (e.g., Fig. 7 with the dots replaced by the multiplexers they conceal). Then, ask students to write an explanation for why one assignment of select bits works and why others do not. The cognitive load for the problem is lowered by removing the need to create the circuit and the need to remember the physical structure of a multiplexer.
2) Provide a completed, meaningful circuit (with MSI components) and ask students to write a plausible problem statement that led to the creation of the circuit. Students could then be asked to explain how the given circuit could be designed from the problem statement.

After assigning these problems, instructors can introduce more complicated design problems.

To reinforce this emphasis on purpose and strategy, instructors should introduce MSI components when the components are needed; examples of this would be to introduce multiplexers just before discussing selecting the outputs of an arithmetic logic unit, and decoders just before introducing RAM decoding schemes. More importantly, instruction should not teach the various MSI circuits as a single instructional unit, so multiplexers, decoders, and similar devices, say, should not be introduced in the same lecture. This type of instruction may reinforce students' faulty associations between multiplexers and decoders.

Digital logic instruction should also help students develop "unifying theories" of digital logic. In introductory physics, biology, or chemistry, a few vital theories—Newton's laws in mechanics, cell theory in biology—serve to unify the concepts of the discipline. The pedagogical value of these theories is that they give instruction a central focal point and help students organize their knowledge into small, manageable chunks of information. Many concepts in digital logic, such as number representations, MSI components, and circuit state, are united by the concept of encoding information into bits. By teaching these topics with a focus on information encoding, instructors can help students organize their knowledge and give them a concept that they can retain after they leave the course.

### B. Future Work and Limitations

Data for the study were collected only at the University of Illinois at Urbana–Champaign. To ensure that the documented misconceptions are not just a by-product of instruction at that institution, this study should be extended to other universities. Expansion studies should also document the prevalence of these misconceptions.

Future research should implement teaching interventions based on reducing cognitive load, emphasizing purpose and strategies for manipulating MSI circuits, and underscoring information encoding. Statistical studies on student scores on the DLCI after these interventions can measure students' gains in conceptual knowledge.

Finally, future research should investigate new ways to uncover how well students have learned the different strategies associated with circuit components. These studies should develop quantitative metrics to measure students' mastery of these strategies.

## ACKNOWLEDGMENT

J. Handzik transcribed the interviews. L. Kaczmarczyk helped develop the analysis protocols. B. Bish provided insightful commentary on chess and chess education.

## REFERENCES

[1] 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.*, Milwaukee, WI, Mar. 2010, pp. 102–106.

[2] 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.

[3] 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. T4D-1–T4D-2.

[4] 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. T3G-1–T3G-2.

[5] 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. F1G-1–F1G-6.

[6] G. L. Herman, M. C. Loui, and C. Zilles, "Flip-flops in students' conceptions of state," *IEEE Trans. Educ.*, 2010, submitted for publication.

[7] 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.

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

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

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

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

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

[17] S. Vosniadou and W. F. Brewer, "Mental models of the earth: A study of conceptual change in childhood," *Cogn. Psychol.*, vol. 24, no. 4, pp. 535–585, 1992.

[18] M. M. Mano and C. Kime, *Logic and Computer Design Fundamentals*, 4th ed. Upper Saddle River, NJ: Prentice-Hall, 2008.

[19] K. A. Ericsson and H. A. Simon, *Protocol Analysis: Verbal Reports as Data*. Cambridge, MA: MIT Press, 1984.

[20] S. Kvale, *Interviews: An Introduction to Qualitative Research Inquiry*. Thousand Oaks, CA: Sage, 1996.

[21] A. Strauss and J. Corbin, *Basics of Qualitative Research*. Thousand Oaks, CA: Sage, 1998.

[22] M. B. Miles and M. Huberman, *Qualitative Data Analysis: A Sourcebook of New Methods*. Thousand Oaks, CA: Sage, 1984.

[23] F. Vahid, *Digital Design*. Hoboken, NJ: Wiley, 2006.

[24] J. F. Wakerly, *Digital Design: Principles and Practices*. Upper Saddle River, NJ: Prentice-Hall, 2006.

[25] G. A. Miller, "The magic number seven plus or minus two: Some limits on our capacity to process information," *Psychol. Rev.*, vol. 63, pp. 81–97, 1956.

[26] P. Chandler and J. Sweller, "Cognitive load theory and the format of instruction," *Cogn. Instruct.*, vol. 8, no. 4, pp. 292–332, 1991.

[27] 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.

[28] 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.

[29] W. C. Chase and H. A. Simon, "Perception in chess," *Cogn. Psychol.*, vol. 4, pp. 55–81, 1973.

[30] G. Burgess, *The Mammoth Book of Chess*. New York: Caroll & Graf, 1997.

**Geoffrey L. Herman** (S'08) is a Ph.D. candidate with the Department of Electrical and Computer Engineering, University of Illinois at Urbana–Champaign.

He is a graduate affiliate of 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.

Mr. Herman is a Mavis Future Faculty Fellow, has won multiple fellowships and grants for improving engineering education, 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.

**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 (MIT), 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.

**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. He holds seven patents. Prior to his work on computer architecture and compilers, he developed the first algorithm that allowed rendering arbitrary three-dimensional polygonal shapes for haptic interfaces (force-feedback human–computer interfaces). 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 is the recipient of the IEEE's Mac Van Valkenburg Award and the University of Illinois Rose and Everitt awards.