THE MATHEMATICS OF INFORMATION SCIENCE

J. ZIMMERMAN and J.W. SMITH
Towson University, Towson, MD 21252
jzimmerman@towson.edu; jsmith@towson.edu

This paper describes a course, The Mathematics of Information Science, which was taught at Towson University in Spring 1998, 1999, and 2000. This course is the junior level interdisciplinary course of the Maryland Collaborative for Teacher Preparation program. The effectiveness of the course in teaching problem solving techniques and abstract mathematical ideas is documented. The students constructed their own knowledge from laboratory experiences involving digital logic circuits. They were subsequently challenged to abstract this knowledge and to find ways to solve progressively more difficult problems using these digital logic circuits. The mathematics of encoding and decoding information constituted the major mathematical content of the course. This course is shown to be effective in introducing prospective elementary and middle school teachers to abstract mathematical ideas and problem solving techniques.

Introduction and Description of the Course

The purpose of this paper is to describe a course called The Mathematics of Information Science and to document its effectiveness. The Mathematics of Information Science is a junior level interdisciplinary course in the Maryland Collaborative for Teacher Preparation (MCTP) program. The MCTP is a program for prospective elementary and middle school teachers who want to be better versed in mathematics and science than the average elementary or middle school teacher.

The educational objectives for this course are:

• To introduce the students to some high level mathematical ideas and notations
• To teach the students to generalize from an example to an abstract concept
• To help the students become more effective problem solvers
• To teach the students to break complex problems down into simpler pieces
• To help the students become more comfortable with technology

These objectives are consistent with recent national standards [1,2].

The central theme of this course is the representation of information using strings of bits. The course also focuses on correcting errors that can occur when information is encoded in this way. This material allows some high level mathematics and computer science to be taught using hands-on discovery based techniques. Although the content of the course is important since
computers represent information in this way, the pedagogical approach is even more important since the students who take the course are prospective elementary and middle school teachers. Our teaching method is characterized by a hands-on constructivist approach to education as espoused by the MCTP [3]. This guided inquiry-based hands-on technique leads to a deeper understanding and retention of the course material. Before taking this course, MCTP students have to take three mathematics courses, two life science courses, and two physical science courses. Thus, the students were familiar with the MCTP approach.

The course began with a survey to assess the prior knowledge of mathematics, science, and computer science possessed by the students. Specifically, we asked them about their familiarity with various software packages, Internet experience, programming experience, and most of the mathematical topics covered in this course. All of the students had some familiarity with e-mail and the internet and usually some other software. None of them had any knowledge of computer programming. All of the students had the enhanced MCTP version of the two mathematics courses required for elementary education majors. The other mathematics courses most commonly taken were basic statistics, discrete mathematics, and calculus (many in high school). Finally, many students asserted that they remembered none of their previous mathematics.

**Pedagogical Approach**

Typically, we presented each topic in five phases:

1. **Background Phase** — We introduced the students to each topic and gave them any necessary background information.

2. **Laboratory Phase** — The students explored the topic in the laboratory phase using either pencil and paper or circuit boards. The students worked on the lab assignments in groups.

3. **Summary Phase** — We discussed the mathematical concepts underlying their laboratory work. This was often a discussion with students sharing what they discovered, but on occasion it would be a lecture. It must be emphasized that this lecture would always follow the laboratory work so that the students had concrete experiences on which to build.

4. **Journal Writing Phase** — The students were required to write about their experiences. This allowed the students to reflect on their new knowledge. We encouraged them to share their thoughts and feelings about what they were learning.

5. **Journal Reaction Phase** — Finally, we answered questions and cleared up any misconceptions that the students might have.
We found this five-phase approach to be extremely successful. In particular, it was encouraging to see the students make discoveries on their own.

Journals

The students were required to keep a journal and to write a journal entry for each class. Each journal entry was graded "acceptable" or "not acceptable." If a journal entry was not acceptable, the student was allowed to rewrite and resubmit it. This policy was instituted to discourage hastily written and poorly thought out journal entries. We found that the journals helped the students reflect on what they learned during the class period. Some students even anticipated in their journal what was coming next in the course. Our instructions for journal entries are given in Figure 1 below.

Figure 1

Journals for Math/Cosc 326

Your grade in Math/Cosc 326 is determined in part by a journal that you keep. Your journal will record your thoughts and impressions about the course and about its content. You are to write in complete sentences with good grammar. We prefer that you type it although a neatly handwritten journal is acceptable. You should keep it in a ring binder and each written page is to be encased in clear plastic.

Each week in the Thursday class, you will turn in the entries that you have made for that week. You are required to have an entry for each class after the first week. You should also have entries about homework, projects, papers, etc. Your entry for each class should include, but not be limited to, the following information.

1. What did you do in class today?
2. Summarize what you learned from today's class. Be specific and give details about the content. Include diagrams, tables, formulas, etc. where appropriate.
3. Explain how this material is related to previous lessons. Speculate on what we will do next in class.
4. Comment about how you feel about this lesson.
5. Miscellaneous Comments.
In addition to the journals, the students were asked to write a small paper and to wire simple projects on the circuit boards (see Appendix A for a sample lab project). In 1998, we required the students to write a paper on the concept of a metric. In 1999 and 2000, the topic of the paper was equivalence relations. This change was instituted because a superficial Internet search on the word metric invariably came up with non-mathematical uses of the word metric, which confused the students. The course culminated with a final project. Each student group was required to design, build, and demonstrate a more complex circuit designed to do a specific task. See Appendix C for the list of suggested final projects given to the students. To test the students’ knowledge of the content of the course, we gave a midterm and a final exam.

Overview of the Course

The course began with a discussion of bits as abstract entities and how they can be realized by 0 : 1, off : on, or open : closed. After some preliminaries, the breadboards were introduced and the students were shown how Light Emitting Diodes (LEDs) placed on the breadboards can represent bits (as off : on). The breadboards used to wire the laboratory experiments have the configuration given in Figure 2. Components can be inserted in a breadboard without any need to solder connections. These components can be purchased from an electronic supply catalog [4].

![Figure 2](image)
Each of the small circles in the diagram represents a connector into which components, such as resistors, LEDs, and integrated circuit (IC) chips, can be inserted. The pins are connected in the following way to allow convenient connection of the components:

- Power is supplied to the components by the columns labeled 0 and +5
- In each of the 13 rows, pins a, b, c, d, and e are connected and pins f, g, h, i, and j are connected, but none of the pins a-e are connected to the pins f-j

Once the students developed a circuit design, they could proceed to use a breadboard to conveniently implement and test their design.

One of the main components of the course is learning how to wire circuits on the breadboards. We gradually guide the students into designing circuits to solve increasingly complex problems. They also learn that there are different ways of viewing a circuit. For example, they could describe a circuit by writing an algebraic expression for the circuit, developing a table that describes the function of the circuit, or realizing the circuit visually by giving a circuit diagram. Knowledge of these different ways of viewing circuits can enhance the student's understanding of circuits and the student's ability to design circuits to solve problems. See Figure 3 for an illustration of this.

**Fig. 3**

**Algebraic Expression**

\[ a \land \neg b \]

**Table**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>(a \land \neg b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**Circuit Diagram**

\[ \begin{array}{c}
    a \\
    b \\
\end{array} \]
The mathematical topics of logic and Boolean algebra [5] are discovered (with some help from the instructors) in a very natural way. It is refreshing to have the students actually begging the instructors to teach them the techniques of the propositional calculus and Boolean algebra so that they can use these principles in their own circuit designs. Another component of the course, which overlaps circuit building, is the study of simple error correcting codes. This topic provides an opportunity to introduce various mathematical ideas, such as equivalence relations, base 2 arithmetic, isometries, logarithms, matrices, metrics, modular arithmetic, permutations and combinations, probability, and Venn Diagrams. Details are provided in the subsequent text showing how these topics are developed. To conclude the study of error correcting codes, bar codes are discussed and the students figure out what errors will and will not be detected in a bar code.

**Specific Activities**

We began the course with some simple activities designed to get the students thinking about representing information using bit strings. For example, we asked the students how many bits are needed to represent each of \( k \) objects by a unique bit string. This leads naturally to logarithms base 2, since the number of bits required to represent \( k \) objects is the integer above \( \log_2(k) \). The base 2 representation of positive integers and base 2 arithmetic was introduced by Unifix cubes [6] in a hands-on activity, and then was done arithmetically using the division algorithm. While doing each step arithmetically, we referenced the corresponding step in the activity with the Unifix cubes. This shows the correspondence between the division algorithm and the grouping of the Unifix cubes into blocks. We only showed how to represent positive integers using bits in 1998. In 1999 and 2000, we also showed the representation of negative integers using the twos complement system. It is worth noting that representations of rational numbers and scientific notation could be inserted at this point.

After these basics, we began with Digital Logic Labs 1 through 6. Lab 1 dealt purely with the proper way to wire LEDs and how the breadboard was used. In Digital Logic Lab 2, the students investigated the six types of logic gates contained on the ICs and constructed input/output tables for each. After this assignment, we gave them the names and commonly used symbols for the logic gates. Labs 3 and 4 and the subsequent homework were designed to help the students understand the proper way to connect gates. The students also began wiring and testing circuits involving multiple logic gates. We stressed the relationship between the actual circuits, the circuit
diagrams, verbal descriptions of the circuits, and the input/output tables. We repeatedly asked the students to describe circuits in all four ways. One example, which we frequently used was the "majority vote circuit" which had an output of 1 if and only if a majority of the inputs were 1. At this point, the students were ready for more complicated projects.

Before Lab 5, Boolean expressions representing circuits [7] were introduced. This gave the students an algebraic language to describe and manipulate circuits. In Digital Logic Lab 5, the students discovered that two different circuits can have the same input/output table. We defined the corresponding Boolean expressions as logically equivalent. This notion of logical equivalence gave the students a mathematical way to show that two circuits do the same thing. Lab 5 is reproduced in Appendix A to give you an idea of the structure of these labs. We stressed this concept as a way to reduce the number of gates used in a circuit.

The students had been asking how to design a circuit to do a particular task. Previously, we had refrained from telling them much about this, but now we spent some time on this topic. Digital Logic Lab 6 led the students to discover the Disjunctive Normal Form (DNF) [8], and we then talked at length about this and some other ways of designing circuits. We also talked about simplifying circuits using Boolean algebra [9]. We found that working with the circuit boards was very helpful to the students’ learning. This helped make the techniques and concepts much more "real" to the students. Some students found that, while at first they did not understand the circuit’s function, hands-on experience wiring and testing the circuit helped them understand the circuit. As one of the students commented: “We were able to watch as our ideas were put into motion.”

We introduced the theory of error correcting codes using Coding Lab 1 which is reproduced in Appendix B. The students constructed by trial and error a 5-bit code with four valid code words, which would correct one error. The idea we wanted them to discover was that any two code words must differ in three positions if the code is to be able to correct one error [7]. As the students worked on Coding Lab 1, we went around the room and looked at their attempts, pointing out where each of their attempts failed. If any two code words differed in only one position, then we can convert the code word to another valid code word by changing one bit. Since both words are valid, the students can’t detect the change. Thus, their code would not detect or correct this one error. If any two code words differed in two positions, we would change the bit in one of those positions. The students found that they could not determine the proper message and therefore could not correct this error. Thus, they were led to the conclusion that in a code, which corrects one error, each code word must differ in three bits from any other code word. Most of the teams got an answer on their own and with a little help by the end of the period, they all did. This lab formed the
basis of most of our discussions about coding.

The ball of radius one around a fixed code word (called the center of the ball) is defined as all words that differ in only one bit from that code word. From Coding Lab 1, the students discovered the fact that a code, which can correct one error, must have disjoint balls of radius one. Therefore, the centers of the balls must be sufficiently far apart. The Hamming distance between two words is defined as the number of bits where the two words differ [7]. The students were asked to fill in a table of the distances between code words in each of the codes that they developed in Coding Lab 1. The students observed that the minimum distance between valid code words was three and that this was necessary in a code which would correct one error.

The students learned that the Hamming distance function \(d\) satisfies three important properties. These properties make error detection and correction possible. These properties are:

1. \(d(x, y) \geq 0 \text{ and } d(x, y) = 0 \text{ if and only if } x = y\)
2. \(d(x, y) = d(y, x)\)
3. \(d(x, y) \leq d(x, z) + d(z, y)\)

for all bit strings \(x, y,\) and \(z\). We proceeded to define a metric [10] as any function that satisfies these three properties. We showed how these abstract properties allowed errors to be detected and corrected. The distance function between points in a Euclidean plane is another example of a metric. The students were asked to find other examples of metrics and write a small paper on them.

The discussion ended with a result called the Sphere Packing Limit, which relates the minimum Hamming distance between code words to the power of a code to detect and correct errors [11]. It states that if \(n\) is the number of bits in each code word, \(M\) is the number of valid code words and \(t\) is the number of errors we want to correct: \(M \cdot \left[ \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{t} \right] \leq 2^n\). We mentioned that a code is perfect if equality holds above.

The students had come up with a number of 5-digit codes from Coding Lab 1 and we wanted to know how many such codes there were. First, we wanted to see how many new codes we could construct from the codes that they found by trial and error. The students were shown the idea of a "FLIP," where the bits in a particular position on all of the valid code words are flipped from 0 to 1 or vice versa, and of a "SWITCH," where the bits in two different positions are interchanged in all code words. We defined a specific five-digit code, called the standard code, which could correct one error. Its code words are 00000, 11100, 00111, 11011. We had the students convert the
standard code to one with a specific word in it by using FLIPS and SWITCHES, recording their steps.

Next, they were asked to complete a distance table for their new code. They observed that this table was the same as the one for the standard code and concluded that FLIPS and SWITCHES preserve distances. We defined an isometry [10] as any operation that preserves distances. They were assigned some homework on this topic to make sure that they understood the ideas discussed. Every team converted their code to the standard code by FLIPS and SWITCHES and presented their results on the board.

Finally, we proved that every five-digit code with four code words, which can correct one error, is equivalent under an isometry to the standard code. We were extremely interested in the journal entries for this class. All of the students claimed to understand the procedure for converting to the standard code and most actually demonstrated their knowledge. Some of the students were clearly impatient with the detailed argument and did not understand why we needed such an argument. However, they were all able to see why the argument implied that there was no five-digit code with five code words that can correct one error.

Next, we mentioned that isometry of codes is an example of the more general concept of an equivalence relation. In 1999 and 2000, the instructors spent more time on equivalence relations and had the students write a paper on other examples of equivalence relations. Most students noticed that equivalence of circuits was another example of an equivalence relation.

At this point, we introduced the 7-bit Hamming code (which is a perfect code). The position of errors in this code can be found easily by using Venn Diagrams [7] or by using modular arithmetic. In the Spring 2000 class, we discussed an efficient method for finding errors in this code using matrix multiplication. In the last part of the coding theory section, we explored bar codes [12]. There are, of course, a number of good references concerning coding theory [13-16].

The hands-on circuit designing experience and the mathematics were interwoven throughout the course, especially in the final project. The purpose of the final project was to design and implement a large circuit that accomplishes some task involving coding. The students worked in groups of two or three. Before starting on the actual wiring, they designed their circuit on paper. Then, they checked their design carefully to see that it accomplished the required task with a reasonably small number of chips. Each group implemented a different project from a list
Summary of Educational Objectives

The course had the following major educational objectives:

• To introduce the students to some high-level mathematical ideas and notations

  We helped the students to understand how to approach complex mathematical problems by using abstract mathematical ideas. For example, logarithms were used to relate the number of possible bit strings of fixed length to the number of bits in one of them. Given a set of objects, the number of bits required to associate a distinct bit string to each object could be calculated. Another example is the use of the concept of divisibility to analyze bar codes. The students saw how mathematical notations were developed and observed that many of the decisions that went into their development are arbitrary [17]. In several places, the students were asked to develop their own notation, and after they had done so they were given the standard notation. We then spent some time looking at what made a notation good. One student cited the importance of “the relationships between notations and concepts.”

• To teach the students to generalize from an example to an abstract concept

  In problem solving, abstraction often helps to get at the heart of the problem. For example, the students found that it was much easier to use circuit diagrams with symbols for logic gates rather than wiring diagrams because the former were not cluttered with the resistors and LEDs, necessary to the wiring, but not to the design of the circuit. Boolean expressions represented an additional level of abstraction. They helped greatly when designing a more efficient circuit, one that used a smaller number of logic gates.

• To help the students become more effective problem solvers

  Some student comments illustrate this point effectively.

  “...to attack a problem from different angles..”

  “The kind of problem solving we were doing was very different from any other I have done. I think my ability as a problem solver was expanded due to this.”

  “… going from a global point of view and moving down the ladder to small steps.”
"… first understand what the problem is or what exactly it is asking … examining [sic] relationships between things and look for patterns. I feel more confident about solving problems, because I have solved harder problems than I have ever had to solve before."

- To teach the students to break complex problems down into simpler pieces
  Once the problem was decomposed into simpler pieces, the students found it easier to solve these individual pieces and could then integrate their solution of them into a solution of the original problem. One student told us that, "this makes solving any problem, no matter the level of difficulty, a possibility. I never looked at problem solving like this before." The final project, because of its complexity, required the students to apply this approach extensively.

- To help the students become more comfortable with technology
  We found that most of these students were either poorly informed about technological issues or were afraid of dealing with them. For example, they found wiring the circuit boards intimidating at first. Once they developed good techniques, they became much more comfortable and even enjoyed wiring circuit boards.

**Student Reflections as Given in the Journals**

The journal entries submitted gave the students an opportunity to reflect on what they learned. Since this course is part of a teacher preparation program, it was interesting to see that several of the students found that their own learning process was helpful in this context.

"In order to truly identify with our students, we have to remember how it feels to deal with abstract concepts and search for understanding."

"This course is designed for us to think in the way we will one day want our students to think and be curious about mathematics."

"If we do this kind of mental role-playing, then we will better understand our students and become better teachers for it."

Many students felt that the amount of time spent doing hands-on activities was a valuable part of their experience.

"It is amazing what a little hands-on learning can do to the understanding of a topic. If we had just talked about circuits and discussed what happened when we used resistors, LEDs, and wires, I think I would have been totally confused. But when I was able to
play around with the switches and circuits, I gained a better and more complete understanding . . ."

"Sandi and I had a little 'accident' when we wired the cathode of the LED directly to the ground with a 270 ohm resistor. This is the best way for me to understand exactly what a resistor does."

"...I can't tell you how happy I am that we're finally going back to working with the circuits. ...I missed the hands-on aspect of working on the digital logic labs."

Several students observed that the process of explaining ideas to others was very helpful to their own learning.

"...I was eager to share my understanding of that material with others. I hope that when I am a teacher I can inspire this kind of enthusiasm in my students, and I think one of the best ways to do this is by displaying enthusiasm for the subject matter myself."

"...I explained how to do some of the problems to another person. This cemented in my mind exactly what I would have to do to solve a problem, and it allowed me to make sure that the way I was explaining the information was clear and concise."

"I am anxious to explain our findings to Lisa what she returns on Tuesday, seeing if I actually can explain what I have done, proving that I understand what I'm doing."

The journals showed that during the semester, the students improved significantly in their ability to deal with the course material.

**Analysis of Student Responses on the Assessment Form**

The students were assessed on their knowledge of the course content on the exams and projects. The effectiveness of the course in achieving its educational goals was evaluated at the end of the course using a course assessment form. They were given 10% of their grade on the final exam for completing this assessment. The results were analyzed only after all final grades had been turned in. The assessment tool we used is reproduced as Figure 4.

**Figure 4**

**MATH/COSC 326 Assessment**

Directions: Please circle the number which best describes your perception of this course and then write a paragraph about your experiences. This assessment is to be turned in to Dr. Lorie Molitor and you will get 10 points on the Final Exam for turning it in.
1. How effective has this course been in giving you experience in breaking a problem down into simpler pieces?
   Very effective Not effective at all
   5 4 3 2 1

   Paragraph: Give an example from this course where you broke a problem into simpler pieces and solved each piece.

2. How effective has this course been in giving you experience in generalizing from an example to an abstract concept?
   Very effective Not effective at all
   5 4 3 2 1

   Paragraph: Give an example from this course where you developed a general concept from looking at concrete examples.

3. How well did the topics in this course fit together? Was there a central theme or themes?
   Very well Not well at all
   5 4 3 2 1

   Paragraph: What was the main theme or themes of this course?

4. How well has this course given you experience in being a flexible problem solver?
   Very well Not well at all
   5 4 3 2 1

   Paragraph: Give an example of a problem that you solved that you are particularly proud of and explain why you are proud of your solution.

5. Has this course given you any new understanding of abstract mathematical concepts that you did not have before the course?
   Yes, a lot. Yes, some. None
   5 3 1

   Paragraph: Give an example of a mathematical concept that you feel you understand well that you did not understand before this course. It does not have to be a major concept.

6. Has this course given you any new understanding of how notations are developed to represent concepts and objects?
   Yes, a lot. Yes, some. None
   5 3 1
Paragraph: Give an example of a notation that you feel you understand well and discuss its strengths as a notation.

7. How effective has this course been in helping you to feel more comfortable with technology?
   Very effective 5 4 3
   Not effective at all 2 1

8. How effective has this course been in helping you to feel more comfortable with abstract thinking?
   Very effective 5 4 3
   Not effective at all 2 1

9. How effective has this course been in helping you to feel more comfortable with mathematics?
   Very effective 5 4 3
   Not effective at all 2 1

10. How well did this course embody the MCTP philosophy?
    Very well 5 4 3
    Not well at all 2 1

11. How well did this course fit in with the other MCTP courses that you have taken?
    Very well 5 4 3
    Not well at all 2 1

12. How has this course impacted your approach to problem solving?

   The responses of the three classes on the assessment questions were surprisingly uniform. A hypothesis test using an alternative hypothesis that the means were different in pairs for the three classes for each question yielded little support for this hypothesis, except for question #3. In question #3, it was observed that the 1999 class had a significantly lower mean than either the 1998 or the 2000 class. A comparison of the 1999 and the 2000 class with the null hypothesis that the means were equal yields a p-value of \( p = .0056 \) and the same comparison between the 1998 class and the 1999 class gives a p value of \( p = .055 \). Since this question asked whether the course had a central theme, we can conclude that the 1999 class did not see the organizing theme as well as the other classes. Nevertheless, we note that the 1999 class had a mean response to this question of 3.47 on a 5-point scale and the authors consider this a good response. The means for the 1998 and 2000 classes were 4.21 and 4.57, respectively. Therefore, we feel comfortable in pooling the data from the three classes for all questions. A bar graph illustrating the numerical results of the assessment on all questions for each of our three Math 326 classes is included as Appendix D.
The results of this pooled analysis are given in Table 1. There were 34 students in the three classes who filled out assessments and $N$ is the number of these students that answered any particular question. We would also like to point out that the 1999 class contained an unhappy student who did not see why this material was needed for teaching in elementary school. This student gave scores of 1 (with 5 being the highest) on 9 out of the 11 questions. Statistically, these responses can be considered outliers on 5 of the 11 questions. After careful deliberation, we decided to include this student's responses on all questions. However, we would like to note that this depresses the mean by .08 on average and significantly increases the standard deviation.

The results of the assessment may be summarized as follows. On question #1, the students felt strongly that the course was effective in giving them experience in breaking a problem down into simpler pieces. As an example of this, many students cited their final project and how wiring circuits forced them to break problems into simpler pieces. Two of the more interesting student responses follow.

"There are so many examples. One is the majority vote circuit and how we went from the concept to the actual gates and wires. The whole class was breaking down concepts into simplified pieces. Isn't that a large part of 'math' also? We did a lot of 'math.' The class built the idea of a circuit, broke it down into pieces, and then went as far as explaining how to fix the pieces when a mistake is made."

"In March we learned to use DNF [Disjunctive Normal Form] to create a circuit diagram for a table that has more than one 1 on it. We used DNF to create a table for each of the 1's, then we created a circuit for each of these tables. Then to join each of the circuits, we added an OR gate between them."

On question #2, the students felt that the course was effective in giving them experience in generalizing from an example to an abstract concept. Here the students' examples covered virtually every topic in the class. Here are several representative student comments.

"We used abstract concepts when we discussed the triangle inequality."

"The Hamming Distance. By looking at a few tables and example codes I was better able to understand the concept."

"We began learned [sic] about logic gates by using a breadboard to see the inputs and outputs of the actual chips. Then we continued to move towards the abstract concepts and ideas of representing these chips with Boolean algebra and
diagrams."

"For example, in Coding Lab 1 we had to find a 5-bit code where an error could be detected. We were later told a way to make finding one easier: a code can correct up to \( r \) errors .... \((2r+1)\) units apart."

"When we discussed a formula for combinations, it was after we did examples and realized that it was a lot of work to do by working out each combination."

"When we learned the twos complement, we used examples. We then made it abstract by using variables to represent a positive number and the negative of that (in binary). We showed how they when added together will come out to zero."

(This student included an extensive diagram of the whole process.)

Question #3 asked the students to identify the central theme of the course. Most of the students saw the dual themes of "wiring" and "coding" and usually considered one of them to be subordinate to the other one. Some student comments illustrating this viewpoint follow.

"Everything in the class eventually focused on two major topics: coding & circuit diagrams (wiring). At one point, we started working with logarithms and all of us thought that this was useless, but it turned out that this helped us to figure out shortcuts when we were working with binary numbers."

"I felt like the main theme of the course was built around computer systems (bit strings, logic gates, and circuit boards). However, with all this, we had to implement many mathematical concepts. We used permutations, combinations, logs, probability, Boolean expressions and many other math concepts."

"The main themes of this course were (1) Exploration of higher level mathematical ideas (2) Intro into circuits and circuit design/execution. These main themes did go together very well though and in fact were interwoven throughout the course."

Other students answered this question by commenting on the educational objectives of this course and how they felt about them.

"I think that one of the main themes of this course focused on changing the way that we think. We were often influenced to think about things abstractly or mathematically. A lot of this course had to do with mathematics."

"I felt too much of an emphasis was put on topics such as logs and metrics. It seems like the course tried too hard to incorporate higher level math."
Question #4 asked if the course encouraged the students to be flexible problem solvers. Most students thought the course succeeded with this task. The students were asked to give an example of a problem that they solved that they were particularly proud of. With a few exceptions, the final project was the example mentioned. The difficulty of the project and the level of teamwork needed to solve it were repeatedly cited. Some of the student comments were:

"The final project gave me the perfect opportunity to tie all of my understandings together. Constructing that circuit was something I never would have comprehended before, but now I could do it, understand it, and explain it pretty well. It served nicely as a culminating activity."

"The final project of solving a circuit setup that has specific inputs, the output would indicate the number of one's that were in the input. I am proud of the solution to this, because I wouldn't have even understood what this problem was asking at the beginning of the class. Now I could figure out what I needed to hook the circuit up. Finally, when we went to wire the circuit, we were able to do so on the 1st try."

"We had to simplify the majority vote circuit down to five gates! This required logical understanding of the problem, not just how gates work."

"... I had to work harder on not getting discouraged and frustrated than on being flexible in solving a problem. ..."

Question #5 asked the students if they achieved any new understanding of abstract mathematical concepts. The results reflected some new understanding, but the standard deviation was quite high. Most students cited an improved understanding of number systems, logarithms, combinations or modular arithmetic although other topics such as metrics were also cited. Our hope was to introduce some sophisticated mathematical concepts to the students. Some students achieved a deeper understanding of these ideas than others, but the journal entries showed that all of the students had some ability to abstract ideas from the labs and projects.

"If someone had said 'parity' to me three months ago, I would have had no idea what they were talking about. I would not have understood check bits or bit sequences. While these are more computer concepts that are related to mathematical concepts, I believe my understanding of base 2 counting in general has developed too. There are just so many things that were new to me before this course, that now I understand (or at least can recognize).

"I could never grasp the concept of logarithms before this class. The way it was
explained was very helpful."

"Once again, Permutations and Combinations. Not one math teacher that I have ever had explained the difference. There was Permut. And Combin., with no comparison of the two. Even though our professors touched briefly on this topic, I gained a lot by asking the question, "What is the difference ...?"

Question #6 concerned the student’s understanding of how notations are developed to talk about abstract objects. We spent more time on notations as abstract entities in the 1998 class than in the 1999 and 2000 classes where we merely pointed out the advantages of particular notations. Since four students in the 1998 class said that they did not understand the question, we chose to de-emphasize this part of the course. Most student responses in the 1999 and 2000 classes were to give a specific example of where a notation (usually Boolean Algebra) made life simpler. Here is one of our better students from the 1998 class:

"Symbols that represent gates. I didn't see this as a strong point of the course. I already understood that notations can be very arbitrary and often has no real logic for designating "D" as an and gate versus "?" or "?". It was mentioned throughout the course how arbitrary notations can be, and I think more emphasis should be placed on this idea. This idea can extend to all mathematical symbols as 'man-made.'"

Here is a representative student comment from the 1999 and 2000 classes:

"Boolean expressions are a notation form that I understood well. By going from a table or circuit diagram to a Boolean expression, you can determine how to verbally describe your circuit. The strength of the notation was that it made sense, and paralleled the diagram and the circuit table that was constructed. Overall, it proved an easy way of describing the circuit."

On the next three questions, the students responded that the course was moderately effective in making them feel more comfortable with technology, abstract thinking, and mathematics. The students felt strongly that this course embodied the MCTP philosophy, but that it was only somewhat related to the other MCTP courses that are taught at Towson University. The instructors believe that the uniqueness of the course content obscured the relationship with the other MCTP courses at Towson.

Finally, we asked how this course impacted the student's approach to problem solving. The
overwhelming response was that this course taught them to break a problem down into smaller, more manageable pieces, solve those pieces, and then assemble the solutions into a solution of the whole. The process of designing a circuit with a particular output encourages this problem solving strategy.

"I look for the big picture first and then break the problem down into pieces. Looking for the overall goal or outcome and sketching a draft plan to reach that goal and then break it down further into steps. I liked going from a global point of view and moving down the ladder to small steps."

"I thought I had a good grasp at breaking down problems, until this class! The relationships between notations and concepts is clearer. I have become more confident in being able to see them. Therefore I guess the class was a good thing..ha ha..the class was great. I am glad I took it."

"This course has taught me that I must first understand what the problem is or what exactly it is asking. This course has also really showed me how to examine relationships between things and how to look for patterns. ... I feel more confident about solving problems, because I have solved harder problems than I have ever had to solve before."

"If there is one thing I have learned from this class, it's that there is always more than one way to approach and solve a problem."

"The nature of the subject forces me to examine each step of the process. My difficulties with problem solving resided in trying to reach a conclusion without thorough understanding of the process. In analyzing circuit diagrams and input/output tables, as well as Boolean expressions, I learned how to more effectively work through the processes of examining each step. This course essentially exercised my ability to solve problems carefully and logically."

Conclusion

The instructors view the specific content chosen as an excellent vehicle for achieving the educational objectives of the course. The richness of this content allows many mathematical ideas to be explored. In addition, the content was accessible to the students. They were able to learn the course material and performed well on the examinations. The students’ final projects were exceptionally well done and demonstrated mastery of the course content. As with any inquiry-based method of learning, the students experienced fairly high levels of frustration. This frustration was kept manageable by having the students work in teams and by the instructors periodically checking
the progress of each team. Thus the content and pedagogical approach were well suited to achieve the educational objectives.

We achieved more success with some of our educational objectives than with others. We felt that this course was extremely successful in helping students to become more effective problem solvers and in encouraging the students to break problems into smaller and more manageable pieces. Wiring actual circuits helped the students become less afraid of technology. After their experiences in this course, they felt more confident that they could handle unfamiliar technology.

We were moderately successful in helping the students become more comfortable with high level mathematics and in improving their ability to generalize from examples. One very positive benefit that occurred repeatedly was the "now I understand that" experience. Much of the students' previous mathematical knowledge was applied to new and "real-life" situations. This required them to understand the mathematics that they were using, instead of merely being able to compute the answer to a standard test question. One student even mentioned that this was true for the topic of "subsets." This is particularly amazing since we spent at most ten minutes on a whim asking the students how a computer might represent a subset of a set.

Although this class was designed for the MCTP program, the subject is accessible to the average college student. With minor modifications, this course would make a good general education course. The subject matter is rich enough to use diverse mathematical ideas and to provide many opportunities for hands-on learning. Introducing elements of computer hardware made the students more comfortable with computer technology. We are quite excited about The Mathematics of Information Science and look forward to teaching it again.

Acknowledgments

The authors gratefully acknowledge the assistance of Richard M. Krach in reviewing the assessment instrument and Tadanobu Watanabe and Rebecca A. Zimmerman for comments on a draft of this article.
References


Table 1 - Assessment Results

<table>
<thead>
<tr>
<th>Question</th>
<th>N</th>
<th>Mean</th>
<th>$\sigma$</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>34</td>
<td>4.12</td>
<td>.80</td>
<td>±.27</td>
</tr>
<tr>
<td>2</td>
<td>33</td>
<td>3.77</td>
<td>.79</td>
<td>±.27</td>
</tr>
<tr>
<td>3</td>
<td>34</td>
<td>3.96</td>
<td>.99</td>
<td>±.34</td>
</tr>
<tr>
<td>4</td>
<td>34</td>
<td>4.09</td>
<td>.74</td>
<td>±.25</td>
</tr>
<tr>
<td>5</td>
<td>34</td>
<td>3.88</td>
<td>1.21</td>
<td>±.41</td>
</tr>
<tr>
<td>6</td>
<td>29</td>
<td>3.55</td>
<td>.89</td>
<td>±.33</td>
</tr>
<tr>
<td>7</td>
<td>34</td>
<td>3.91</td>
<td>.92</td>
<td>±.31</td>
</tr>
<tr>
<td>8</td>
<td>34</td>
<td>4.00</td>
<td>.91</td>
<td>±.31</td>
</tr>
<tr>
<td>9</td>
<td>34</td>
<td>3.65</td>
<td>.97</td>
<td>±.33</td>
</tr>
<tr>
<td>10</td>
<td>34</td>
<td>4.21</td>
<td>.83</td>
<td>±.28</td>
</tr>
<tr>
<td>11</td>
<td>33</td>
<td>3.23</td>
<td>1.11</td>
<td>±.38</td>
</tr>
</tbody>
</table>
DIGITAL LOGIC LAB 5

Purpose: To learn to develop a circuit diagram starting from a Boolean expression for the circuit. To develop a method for testing to see whether or not two circuits are equivalent. To implement this method by actually wiring some circuits.

1. For each of the following Boolean expressions, give a circuit diagram which implements the expression.

(a) \( a \land (b \lor c) \)

(b) \( (a \land b) \lor (a \land c) \)

2. Use input/output tables to determine whether or not these circuits are equivalent.

3. Wire the circuit \( a \land (b \lor c) \) (from 1(a) above).

4. Suppose we use the following block diagrams to represent the circuits \( a \land (b \lor c) \) and \( a \land (b \lor c) \):

Wire the \( (a \land b) \lor (a \land c) \) circuit (from (b) above) on the same breadboard on which you already constructed the circuit \( a \land (b \lor c) \) by using the following scheme which will make it easy for you to compare the two circuits.

Note that the block diagram for your entire circuit is

which is the first two-output circuit you have wired!
5. Demonstrate this circuit to your instructor explaining whether it shows that the two circuits \( a \land (b \lor c) \) and \( (a \land b) \lor (a \land c) \) are equivalent or not equivalent.

6. Since the 74LS08 and the 74LS32 chips have exactly the same pinouts, you can just swap the 74LS08 and the 74LS32 chips in the circuit above to check to see whether or not the circuit \( a \lor (b \land c) \) is equivalent to the circuit \( (a \lor b) \land (a \lor c) \).

7. Before testing to see if these new circuits are equivalent, predict the answer using circuit tables.

8. Demonstrate this new circuit to your instructor explaining whether it shows that the two circuits \( a \lor (b \land c) \) and \( (a \lor b) \land (a \lor c) \) are equivalent or not equivalent.

9. Are either of the circuits of #8 the same as either of the circuits of #5? Explain your answer.
Appendix B

Coding Lab 1

Purpose: To introduce the principles of error correcting codes.

Step 1: Construct two 3-bit sequences. The first sequence will represent the word "ONE" and the second will represent the word "TWO". You agree on this code with your lab partner.

Questions:

1. Can you tell if he has made a change or not?

2. Suppose that you know that he has made a change. Can you tell what the correct message is?

Step 2: Try this out first with your partner and then with Dr. Zimmerman or Mr. Smith. If you cannot answer both questions above with a YES, then start over on Step 1 and redesign your code.

Step 3: Explain why your code works. If you cannot do this, find a code, which doesn't work and explain why it fails to work.

Step 4: Construct a 5-bit code, which will encode four items, "apples", "bananas", "cherries" and "oranges". Design your code so that you can detect if one bit has been changed and correct it.

Step 5: Try this out first with your partner and then with Dr. Zimmerman or Mr. Smith. If you cannot answer both questions above with a YES, then start over on Step 3 and redesign your code.

Step 6: Explain why your code works. If you cannot do this, find a code which doesn't work and explain why it fails to work.

Hand in both codes that you came up with and your explanation of why it works.
Appendix C

Final Projects

1. A Hamming encoding circuit. The 4 input bits give the message to be sent and the 7 bits of the output give the valid Hamming code for this 4-bit message.

2. Part 1 of a Hamming decoder. This project has 7 input bits and 3 output bits. The 7 input bits give the received Hamming code of a message. This could be the valid code word for a message or it may differ from some valid code word by one bit. The 3 output bits give the binary form of the subscript of the error bit if there is an error or are all 0 if there is no error. For example, if there is an error in $a_5$, the 3 output bits are 101.

3. Part 2 of a Hamming decoder. This project has 7 input bits and 4 output bits. Three of the input bits are the 3 error location bits generated by Part 1 above. The remaining 4 input bits are the 4 uncorrected message bits of the received message from Part 1. The project generates as its 4 output bits the correct message.

Note: Part 1 and part 2 above when wired together form a complete Hamming decoder.

4. A 2-bit adder. This project has 4 input bits and 3 output bits. The 4 input bits are grouped into 2 pairs with 2 bits in each pair. Each of these pairs represents a 2-bit binary number and the 3 output bits represent the sum of these two numbers. For example, if one of the input pairs is 10 and the other input pair is 11, the output bits are 101 (since $2 + 3 = 5$).

5. A 1-bit arithmetic logic unit (ALU). This project has 3 input bits $a$, $b$ and $s$ and 1 output bit $x$. If $s$ is 0 then the output bit $x$ is $a \& b$ and if $s$ is 1 the output bit $x$ is $a \lor b$.

6. A 3 bit 1's counter. This project has 3 input bits and 2 output bits. The output bits display in binary form the number of the input bits that are 1. For example, if the input bits are 101, the output bits are 10.

7. A Hamming distance evaluator. This project has 6 input bits and 2 output bits. The input bits are grouped into 2 pairs of 3 bits each. The output bits give in binary form the Hamming distance between the two pairs of input bits. For example, if one input pair is 011 and the other input pair is 110, the output bits are 10 since the Hamming distance between 011 and 110 is 2.
Appendix D

Assessment Results

<table>
<thead>
<tr>
<th>Question</th>
<th>SP1998</th>
<th>SP1999</th>
<th>SP2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>