Effectiveness of Karnaugh Mapplet Use in Student Learning of Digital Logic Skills

Phillip A. Mlsna, Erica Liszewski
Electrical Engineering / Computer Science
PO Box 15600, Northern Arizona University
Flagstaff, AZ USA
Phillip.Mlsna@nau.edu

Introduction

Our core course in digital logic at Northern Arizona University (NAU) aims to develop a set of key skills needed throughout the electrical and computer engineering curriculum. Digital logic covers the analysis and design of combinational and sequential digital logic circuits using the standard principles of Boolean algebra. The subject forms a vital part of the technical foundation that enables our students to contribute to the field of modern digital hardware. Students are often called upon in later courses to apply a key set of digital logic analysis and design skills to the advanced material being covered. Those who are weak in these skills are at a decided disadvantage.

Experience at NAU indicates that many students who do well in the digital logic course often have trouble applying this material in later courses. Colleagues at other universities have expressed similar concerns. This indicates that their knowledge is often rather fragile. There are several likely causes:

1) *Insufficient student practice with critical skills and concepts*. Most students require extensive practice to develop consistency, accuracy, and speed. In the traditional course, practice primarily takes the form of assigned homework problems.
2) *Long delays in providing students individual constructive feedback about their work*. The traditional homework cycle can interfere with the ability to identify and address student errors quickly and so delays the learning process. Incorrect techniques and bad habits often become entrenched before the instructor notices.
3) *Spotty constructive feedback*. The burden is traditionally placed heavily upon the student to seek help and clarification. Most students tend not to seek assistance until the situation has persisted for some time.

Approach

The minimization of Boolean expressions using the Karnaugh map (K-map) is a critical skill that is developed and used throughout the course. Since many students have difficulty with K-map techniques, we decided to target these skills first. The primary goal has been to provide a means for extensive student practice with K-map problems while providing immediate constructive feedback. Sum-of-products (SOP) and product-of-sums (POS) forms should be given equal emphasis.

Experience has shown that students tend to make the following mistakes on Karnaugh map problems:

- Incorrect translation of the truth table entries into the K-map grid
- Illegal groupings
- Redundant groupings
- Not recognizing possible wrap-around groupings
- Including all don't-care cells into groupings
- Producing an incorrect Boolean expression from a correct map

To address these issues, we have implemented the K-Mapplet, a Java applet designed for student practice with Karnaugh map problems. Our K-Mapplet allows students to practice a wide variety of computer-generated problems at their own pace and at an appropriate level of difficulty. Unlike other Karnaugh map software currently available commercially or on the web [1-3], the K-Mapplet checks student responses and gives general feedback. It does not do the work for the student.

K-Mapplet

The K-Mapplet contains a variety of levels, each with varying degrees of difficulty. It supports from 3 to 6 variables, both SOP and POS forms, and don't cares. All problems begin with a truth table and an uninitialized K-map grid of appropriate size. Problems can require the student to translate the truth table to the Karnaugh map grid only, translate and select a set of minimized groupings, or proceed all the way to the entry of the minimized Boolean equation. Points are awarded for correct answers, with more difficult problems awarding more points than simple problems. Students are allowed to select a level in which to practice their skills. Figure 1 illustrates the dialog from which students select a problem level.

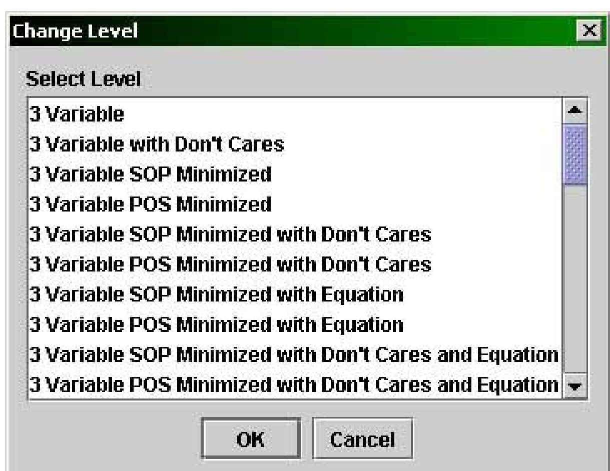

Figure 1: Problem selection dialog box

As an example, let's walk through all three possible stages of a problem. We will use a 4-variable problem with don't-cares and do translation, grouping, and equation entry in SOP form. A problem not requiring all three stages would perform checks only for the required stages.

A new problem is generated based on the level of difficulty selected by the student. A truth table is built using randomly generated outputs. If don't-cares are allowed, they will appear as X entries in the truth table outputs. The applet draws the K-map grid to match the number of variables specified by the problem and initializes all cells to zero. Figure 2 shows our 4 variable example problem as it initially appears.

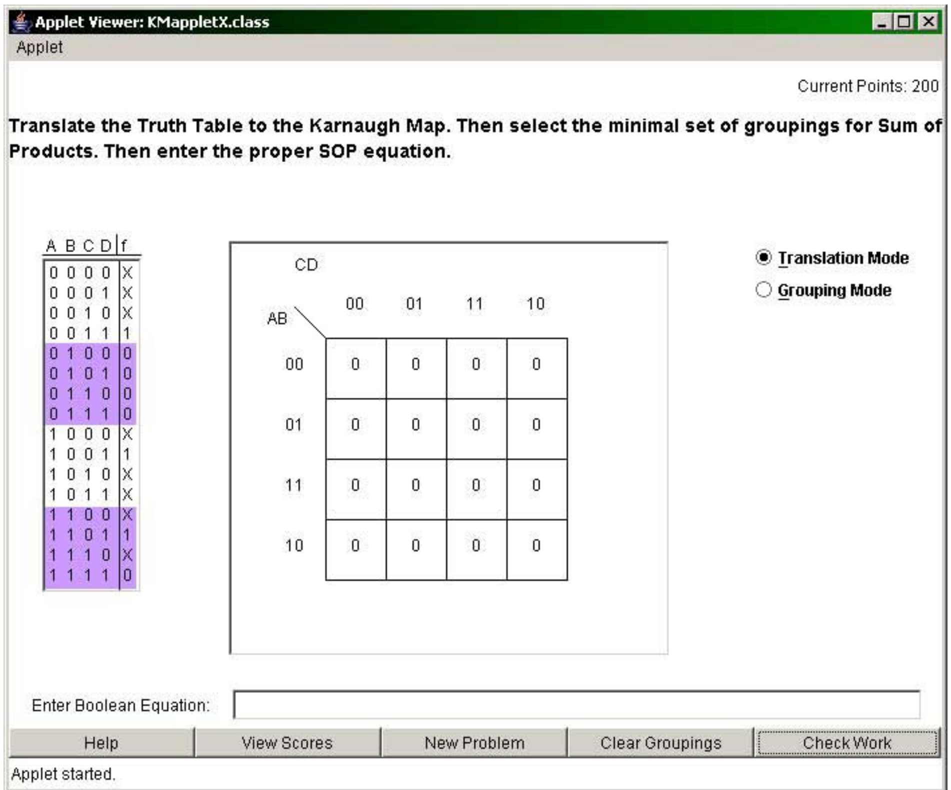

Figure 2: Initial presentation of a 4-variable SOP problem

The first stage requires the student to translate the truth table to the Karnaugh map grid. The student can change values in the grid by clicking on cells. Cell values cycle through valid values, with the X value appearing only in problems allowing don't-cares. Figure 3 shows our problem with the truth table translated to the K-map grid.

The second stage requires the student to designate minimized sum-of-products groupings on the grid. The student must first lock their grid entries by changing from Translation Mode to Grouping Mode. A group is created by selecting a set of desired cells. This is done by pressing and holding the CTRL key and either clicking individual cells or dragging the mouse to create a rubber band around a group of adjacent cells. In order to allow groupings to span the edges of the grid, or across multiple 4x4 grids in a 5- or 6-variable problem, the grouping is ended only on the release of the CTRL key. Figure 4 shows our example with grouping designation in progress. An already designated group is shown with a gray rubber band. The shaded cells show the selected cells of a grouping in progress before the release of the CTRL key.

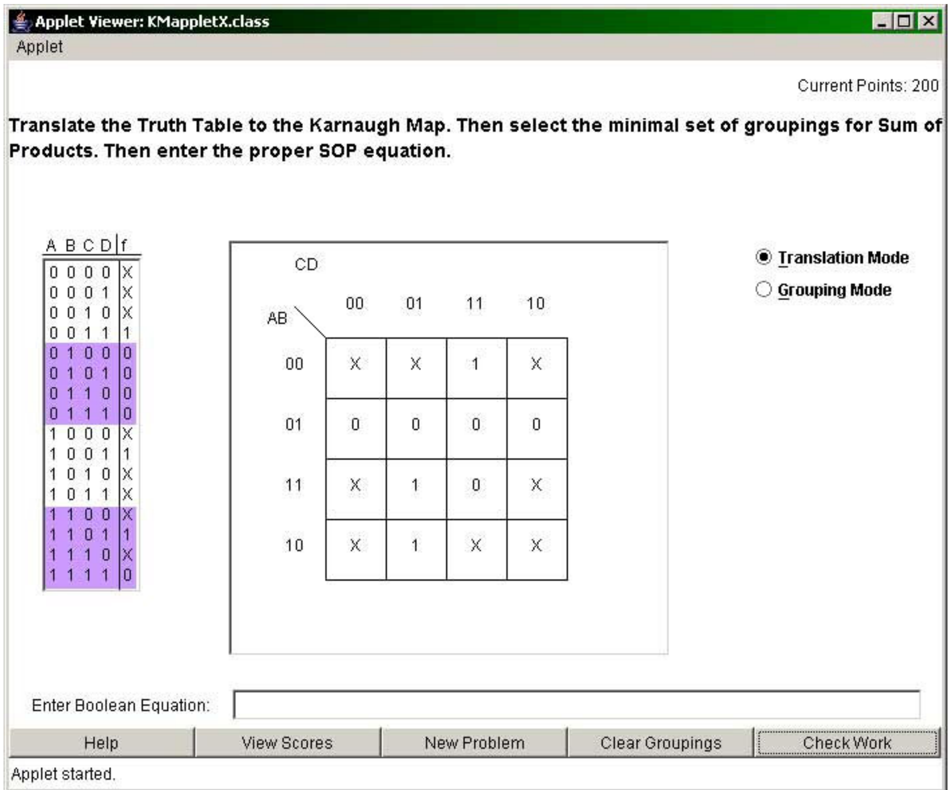

Figure 3: The K-map grid has been transcribed from the truth table

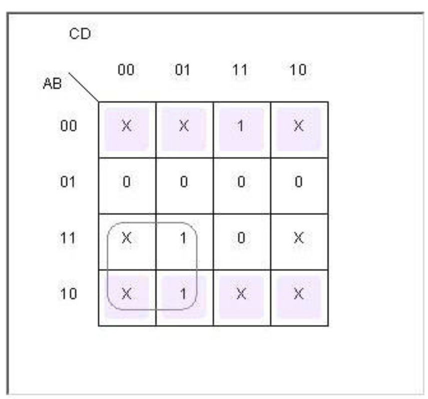

Figure 4: Group selection in progress

At the release of the CTRL key, the newly designated grouping is checked for validity. A valid group is one where: the group contains a legal number of cells, all cells are logically adjacent, all cell values are legal values (1 and X for SOP, 0 and X for POS), and the group represents a prime implicant. The Boolean equation for the groupings the student designates is constructed internally as groupings are made. The equation must be created based on the student's groupings to ensure the student's entered equation matches the designated groupings in cases where there are multiple correct minimizations. The student is permitted to select invalid groupings, but these will result in an incorrect solution. Figure 5 shows our problem with correctly selected groupings.

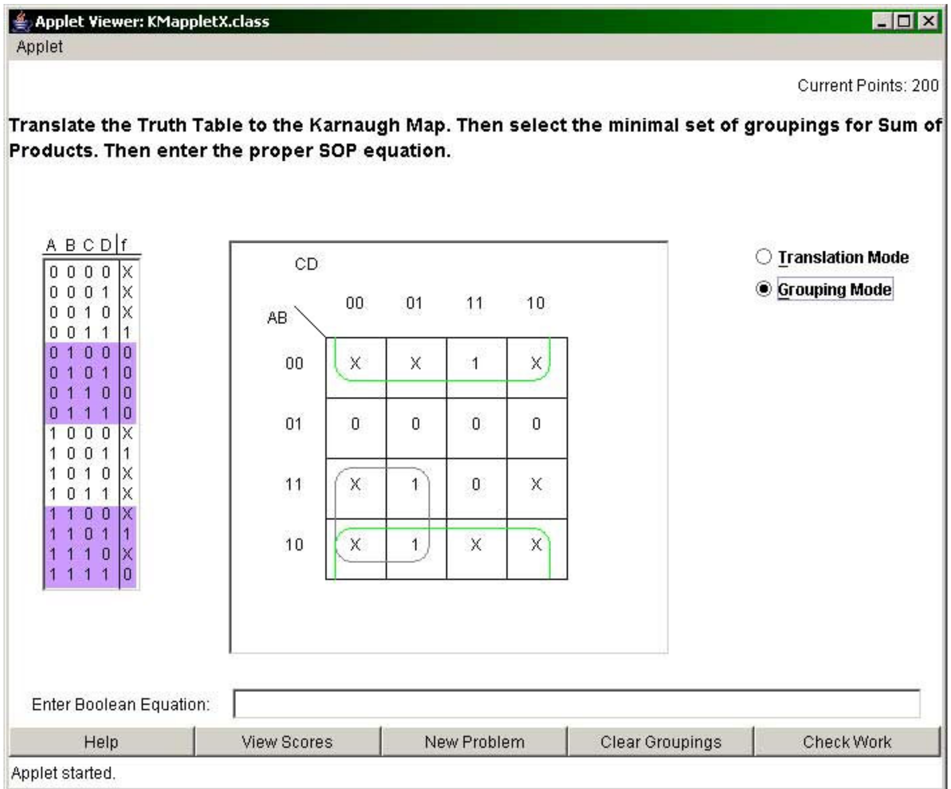

Figure 5: All valid groupings have been chosen

The third stage requires the student to enter the correctly minimized Boolean equation into the text box below the grid. Proper syntax must be used when entering the equation. Syntax requires leading "not" designation, parentheses around POS terms, and no parentheses around SOP terms. The order of terms, the number of spaces between characters, and the order of variables within terms are all unconstrained. Our problem is shown in Fig. 6 with the equation entered.

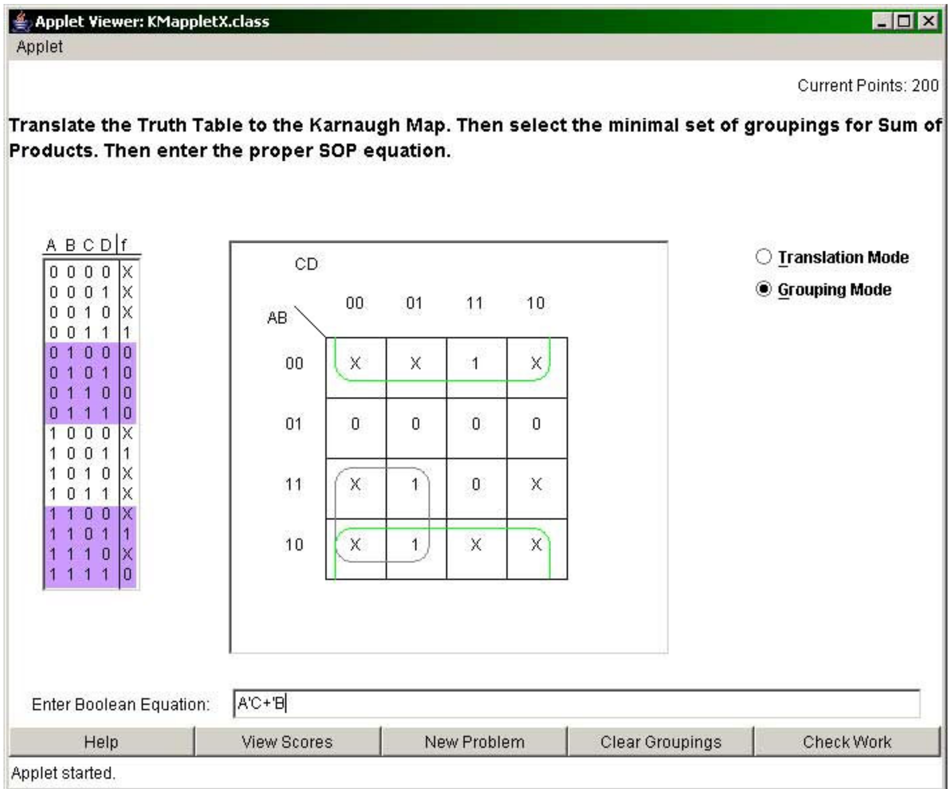

Figure 6: The SOP equation has been entered

When the student believes she has entered the correct answer, she presses the Check Work button. The applet compares the K-map data with the truth table data to check the translation step. Groupings have already been checked for validity and are now checked to ensure no grouping is redundant. Groupings cannot be checked for redundancy prior to the end because a group may not be redundant when it is selected, but might become redundant due to a later selection. Both the entered and generated Boolean equations are parsed and compared to check the student's entered equation. If all required checks pass, the student is informed and given options on what to do next. Points are awarded for the correct answer. Figure 7 shows the dialog generated for a correct answer.

Figure 7: Dialog box for correct response

If a check fails, the student is informed and pointed to the earliest erroneous step. A check can fail in any of the three stages: translation, grouping, or equation entry. The student may try the problem again for reduced points, or skip to a new problem and lose points. An example dialog for an error in the translation of the truth table is shown in Fig. 8.

Figure 8: Example dialog box describing a student error

Results

The K-Mapplet was first integrated into our digital logic course in Fall 2003. From then on, the students were given roughly three hours of laboratory time during the semester to practice K-map problems using the applet. During this structured time, the professor and a teaching assistant were present to answer questions. Grades for this activity were based on the level of participation, not performance. Students were also strongly encouraged to practice at other times via the web. As expected, the additional practice activity peaked strongly just before each exam.

Subjectively, most students considered the K-Mapplet a very positive part of the course. Many treated the applet like a game and became quite involved in the pursuit of large point scores, even after being informed that the point value had little meaning and would not affect their grade. Several expressed that they thought it was a better way to learn. Others said the applet was more time efficient and less frustrating for them than using pencil and paper.

But does the K-Mapplet produce improved learning? To assess the efficacy of the K-Mapplet approach in improved student learning, a direct quantitative comparison of student performance on final exam K-map problems was done. The Fall 2003 and 2004 classes used the K-Mapplet, while the Fall 2002 class did not. All three instances were otherwise very much alike: same professor, similar lectures, and similar homework assignments. Identical K-map problems appeared on the final exams for all three semesters. These consisted of two 4-variable K-map problems with don't cares: one SOP and one POS. Since final exam papers are kept on file and not returned to the student, past use of these specific problems should not have influenced the results.

Table 1 presents the number and proportion of the students who made various types of errors on the SOP final exam problem. Table 2 shows similar results for the POS final exam problem. The "translation to map" error indicates the student failed to properly transcribe the truth table entries into the K-map grid. "Map arrangement" usually means an incorrect row or column order in the map. The "Eq. not in SOP form" means the student produced a POS expression when SOP was required. "Good map, bad equation" means the student incorrectly produced a Boolean expression from a set of good K-map groupings. "Not fully minimized" indicates redundant groupings or incorrect prime implicants.

Table 1: Student performance on final exam SOP problem

Student error type	Fall 2002 44 students		Fall 2003 18 students		Fall 2004 42 students	
	number	rate	number	rate	number	rate
Correct, no errors	28	0.636	14	0.778	32	0.762
Translation to map	8	0.182	0	0	1	0.024
Map arrangement	1	0.022	1	0.056	2	0.048
Eq. not in SOP form	1	0.022	0	0	1	0.024
Good map, bad eq,	2	0.045	1	0.056	2	0.048
Not fully minimized	4	0.091	2	0.111	4	0.095

Table 2: Student performance on final exam POS problem

Student error type	Fall 2002 44 students		Fall 2003 18 students		Fall 2004 42 students	
	number	rate	number	rate	number	rate
Correct, no errors	15	0.341	10	0.556	26	0.619
Translation to map	3	0.068	0	0	2	0.048
Map arrangement	1	0.022	2	0.111	2	0.048
Eq. not in POS form	3	0.068	1	0.056	1	0.024
Good map, bad eq,	14	0.318	0	0	1	0.024
Not fully minimized	8	0.182	5	0.278	10	0.238

From the data in Tables 1 and 2, it is possible to estimate the confidence level for the following hypothesis:

> Use of the K-Mapplet tools resulted in a statistically significant improvement in students' abilities to correctly perform Karnaugh map problems at the end of the course.

The number of students from 2003 and 2004 can be combined to form a sample size $n_2 = 60$ students who used the K-Mapplet. The original $n_1 = 44$ students from 2002 are the control group, prior to the introduction of the applet. The proportions of students who made no mistakes on the SOP problem are $p_2 = 0.767$ for the K-Mapplet group and $p_1 = 0.636$ for the control group. Using the large sample confidence interval to estimate the difference between the two binomial parameters [4], we obtain better than 90% confidence that the above hypothesis is true for the SOP problem and better than 99% confidence for the POS problem. It is clear that the use of the K-Mapplet produces a statistically significant improvement in student learning.

Other conclusions are suggested by the data in Tables 1 and 2. There appears to be a significant reduction in translation errors compared to the control group. There was also a major reduction in "good map, bad equation" errors in the POS problem. The control group had considerable difficulty with this, but the K-Mapplet group had almost none. There was no improvement in the "not fully minimized" error rate. We plan to investigate these questions more formally in the near future.

Conclusions and Future Plans

The results of this work have proven very effective in improving student learning and skills with Boolean expression minimization using the Karnaugh map. Students have largely enjoyed the approach and have performed measurably better on their final exam problems.

We would like to invite interested colleagues to use the K-Mapplet in their courses. Besides improved student learning, benefits should include improvements in the applet software and gathering improved assessment information. Please contact the author by e-mail.

Work is underway to assess students' abilities with key digital logic skills during the first week of a later course. This way, retention of the material can be measured and the information used to further improve the digital logic course. An additional applet involving timing diagrams and propagation delays has been introduced and is being assessed in a similar manner. The K-Mapplet is planned to be part of an outreach effort to high schools in order to motivate and attract future engineering students.

Acknowledgements

The authors wish to thank both the Arizona Board of Regents' Learner-Centered Education program and the Hewlett Foundation for supporting the work. We wish to thank Billy Olsen, Sean Collins, Navenkanth Subramanian, and David Tucker for their valuable assistance in the development and implementation of this project.

Bibliography

[1] OneGram Software, "Karnaugh Map 4" freeware and commercial software, http://www.puz.com/sw/karnaugh/

[2] Roth, C. H., "*LogicAid, CAD Software for Logic Design, Version 3.0 for Windows: Users Guide & Reference Manual*," Brooks Cole, 1999.

[3] Shuriksoft Software, "Karnaugh Minimizer", commercial software, http://karnaugh.shuriksoft.com/

[4] Walpole, R. E. and Myers, R. H., "*Probability and Statistics for Engineers and Scientists*," 4th ed., Macmillan, New York, 1989.

Biographic Summaries

Phillip A. Mlsna, Ph.D.
Dr. Mlsna is an assistant professor in NAU's electrical engineering department and has taught the digital logic course many times. He has extensive industry experience as a computer hardware designer for Hewlett-Packard and an image processing algorithm developer for Raytheon. His technical research interests are in image analysis, computer vision, and image processing.

Erica Liszewski
Erica is the very capable programmer who wrote the K-Mapplet software under Dr. Mlsna's supervision. She will complete her B.S. degree in computer science at NAU in May 2005. She is investigating opportunities for graduate work to develop her interests in human-computer interaction, computer graphics, image processing, and computer art.