Flipping over Flipped Instruction

Mark Fienup
Computer Science Department
University of Northern Iowa
Cedar Falls, Iowa 50614-0507
fienup@cs.uni.edu

Abstract

Flipped instruction is a relatively recent instructional technique which inverts the traditional in-class and out-of-class activities. Activities traditionally completed in class such as instructor supplying course content are now completed on the students' own time **before** class. Thus, freeing class time for traditionally out-of-class activities that apply course content to aid its assimilation by homework-like problem-solving or a deeper discussion.

This paper describes my experience in flipping a Computer Organization class. It describes the rationale for flipping the class, tools used to prepare course materials, an outline of a typical week of the flipped course, and student feedback to the flipped course. Additionally, it presents suggestions for other instructors interested in experimenting with flipping a Computer Science course.

1 Background

Flipped instruction is an instructional technique which inverts the traditional in-class and out-of-class activities. Activities traditionally completed in class such as instructor supplying course content are now completed on the students' own time **before** class. Thus, freeing class time for traditionally out-of-class activities that apply course content to aid its assimilation by homework-like problem-solving or a deeper discussion.

Key elements of a flipped classroom are summaries by Brame [2] as:
1. Provide an opportunity for students to gain first exposure to new material prior to class.
2. Provide an incentive for students to prepare for class.
3. Provide a mechanism to assess student understanding.
4. Provide in-class activities that focus on higher level cognitive activities.

Prior exposure to new material before class allows students to do the lower-levels of Bloom's Taxonomy [1] (knowledge and comprehension) on their own so class time can focus on assimilating that knowledge through activities aimed at higher-levels of Bloom's Taxonomy (application, analysis, synthesis, and evaluation). Student pre-class preparation is key so tying points to pre-class activities provides an incentive to come prepared. Pre-class online quizzes allow an instructor to tailor class activities to focus on areas where students are struggling and help students pinpoint areas where they need help.

This paper describes my experience in flipping my Computer Organization class at the University of Northern Iowa (UNI). I'm currently in my second semester of teaching this course via a flipped approach. It describes my rationale for flipping the class, tools used to prepare course materials, weekly outline of the flipped course, and student feedback to the flipped course. Additionally, it presents suggestions for other instructors interested in experimenting with flipping a Computer Science course.

2 Rationale for Flipped Computer Organization

The Computer Science (CS) curriculum at UNI is very flexible with Computer Organization being the only required "systems" course that all underclass CS majors must take as part of their CS core. It also serves as a prerequisite for all the upper-level "computer systems" electives: Operating Systems, Networking, and Computer Architecture. Therefore, the goals of the course are broad with a lot of material to cover. After this course, students should understand:

- how data is represented and manipulated on the computer,
- simple combinational and memory circuits used to build computer components,
- how these circuits are organized to build a computer,
- how to program in assembly language,
- how high-level language programming languages are implemented with respect to the run-time stack and built-in data structures such as arrays and records, and
- general concepts of hardware support necessary for an operating system.

I have always been able to cover all of the material, but always felt that students would benefit by more practice applying the concepts to get a deeper understanding. Thus, the main impetus for flipping the class was to allow students more class-time for activities aimed at higher-levels of Bloom's Taxonomy (application, analysis, synthesis, and evaluation).

3 Tools used to Flip the Course

There is some truth to the Mark Twain quote:

"To a man with a hammer, everything looks like a nail."

There are lots of tools available to aid in flipping a class. My choices were driven by availability of tools supported by my University and comfort of use to fit my teaching style.

Before flipping, I always assumed students came unprepared for class, i.e., did not do the assigned textbook reading. They never disappointed me. They were being practical since I was going to tell them what was important in class. Plus, I believe that most have never mastered the skill of learning from a textbook, but use it as an "as needed reference." My clear choice for pre-class first exposure to new material is video.

There are some basic Computer Organization related videos available on the Internet, e.g., converting from decimal to binary numbers, but I decided to create my own to tailor them to my course. The literature suggestions some guidelines for video including:

- keep videos short (5 - 10 minutes) and focused on a single topic.
- incorporate active learning activities if possible so students are not passively watching video but are actively engaged.

Initially, I wanted to split my pre-class activities into several short videos with quiz questions interspersed after each video. I know of at least one commercial tool, SoftChalk (softchalk.com), that supports this functionality.

Unfortunately, UNI's learning management system (LMS), BlackBoard 9.1, does not suppose tightly coupled interleaving of video and quiz questions. I decided on longer pre-class videos (25 - 60 minutes) before each class followed by a pre-class, multiple-choice, on-line quiz. Since I always teach the class on a Tuesday and Thursday schedule, I prepared 20 video-quiz pairs during the summer of 2014 to cover the first two-thirds of the course.

The last one-third of the course was left unchanged which for me is not traditional lecture. For years, my non-flipped teaching style has consisted of in-class mini-lectures (~10 minutes) followed by groups/pairs of students actively applying the mini-lecture material to supplied questions which concludes with a whole-class discussion of the correct answers. This cycle is repeated several times throughout a 75-minute class period. The questions from these in-class activities were mostly reused when flipping the class. Over winter break, I flipped the last one-third of the course which we have not gotten to yet this semester.

Recording of the pre-class videos was accomplished using a document camera (borrowed from a classroom for the summer) at my desk in conjunction with Panopto (www.panopto.com) lecture capture software supported by UNI. This allowed me to create "Khan Academy" style narrated videos of my voice and hand while writing with a pencil on paper under the document camera. I scanned the written papers using the departmental photocopy machine to create PDF documents for posting with the videos on the LMS. The Panopto software was also used to capture the live in-class sessions which I posted after each class.

Each pre-class-video quiz was created using BlackBoard's Assessment/Test tool. These were typically 5 question multiple-choice quizzes. They were automatically graded with due dates set to expire at the start of each class. The quizzes comprised 10% of a student's overall course grade which seems to be enough incentive to encourage students to prepare for class by watching the videos.

4 Flipping in Practice: a Typical Week

To get a better feel for the flipped course I'll walk you through a typical week. I always try to teach all my courses on a Tuesday/Thursday schedule. The longer 75 minute (vs. 50 minutes on MWF) class periods work better for the active-learning group activities. At the start of the semester, I have students buy a course-packet from our local CopyWorks (~$20) which is organized chronologically by lecture number. It contains the syllabus and reading schedule, test review materials (old tests and list of review topics), but for each lecture it has:

- handwritten pre-class video notes generated while creating the pre-class video. This allows students to watch the video quicker by annotating my notes instead of starting from scratch.
- questions that students should try to answer **before** taking the pre-class video quiz (these questions are denoted in **bold**). These questions target the lower-levels of Bloom's Taxonomy (knowledge and comprehension).
- questions that students will answer in-class as part of their active-learning group activities. These questions target the higher-levels of Bloom's Taxonomy (application, analysis, synthesis, and evaluation) or topics that are particularly confusing that I'd rather explain in person.
- supplemental descriptions of course topics which are either not covered or weakly covered in the textbook. For example, I created a MIPS assembly-language programming guide which is only briefly mentioned in the textbook.

Having the course-packet bound in chronological order helps to keep the students organized. They just need to bring the course-packet to class every day. Appendix A contains a sample collection of handouts for Lecture 4.

Before each class, I recommend that students follow the below steps. I typically only track the online quiz results, but using the eLearning system I have looked at individual video usage. The recommended pre-class steps are:

1. scan the pre-class questions (denoted in **bold**) on the lecture handout in their course packet so they know "what's important" from the video.
2. watching the pre-class video (25 - 60 minutes) a day or two before class

3. attempt the pre-class questions (denoted in **bold**) while watching the video or shortly after. The video can be pausing and partially re-watching as needed.
4. before class take the multiple-choice, on-line quiz.

I'm fairly confident that most students take the pre-class video and online quiz serious. This semester's class has had 16 quizzes so far with an average score of 3.96 out of 5 and 88.5% of quizzes taken. Figure 1 shows the Panopto statistics for video usage for the past month.

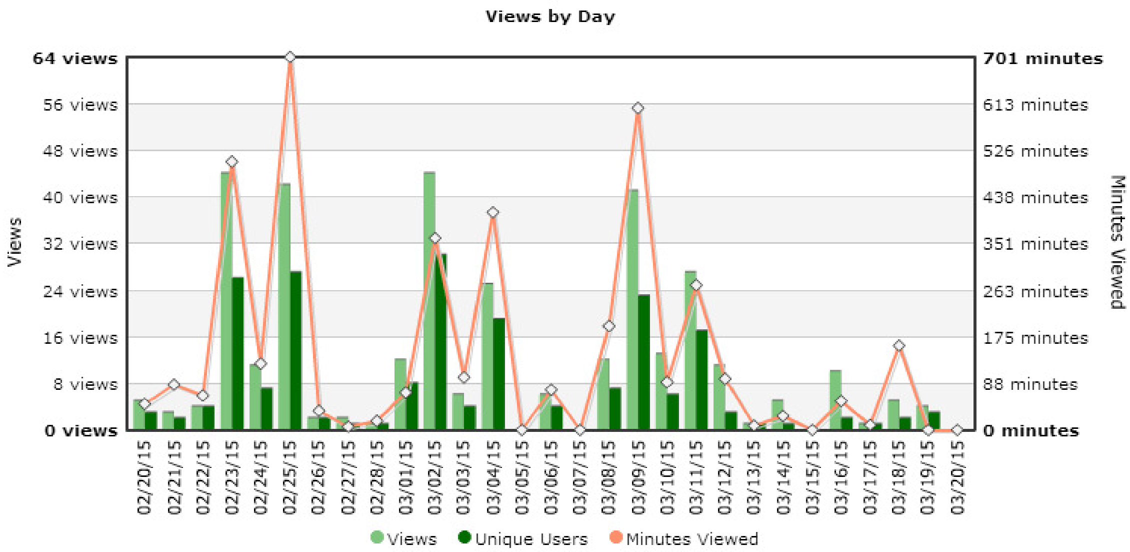


Figure 1: Video Usage for the Previous Month

The first three weeks show expected spikes on Mondays and Wednesdays. The last week (3/15/15 - 3/20/15) was our Spring break with an electronically submitted assignment due on Wednesday, 3/18/15.

I typically start class by briefly summarizing the key points of the pre-class video often in the context of reviewing the answers to the pre-class questions. The goal is **not** to redo the video lecture, but to clear up any confusion about the video material. Then what I do varies somewhat. I might have students:

- work on another problem over the video material, or
- I might do a mini-lecture (~10 minutes) over advanced material followed by groups/pairs of students actively applying the new material to supplied questions which concludes with a whole-class discussion of the correct answers.

This second option is repeated several times throughout a 75-minute class period. In-class participation counts as 10% of a student's final grade.

After every week, students have individual homework assignments to provide further practice and allows for assessment of student understanding. These assignments primarily target the higher-levels of Bloom's Taxonomy (application, analysis, synthesis, and evaluation), and count as 20% of a student's final grade.

Two mid-term tests and a comprehensive Final examination split the class into thirds. Each test counts as 20% of a student's final grade. The number and difficulty of these tests has not changed over the non-flipped version of the course.

5 Student Feedback to the Flipping Course

I was told to expect student resistance to the flipped course since students are accustom to the traditional lecture format. Fortunately, I got virtually no negative feedback with many informal positive oral comments. My Fall 2014 Computer Organization course's anonymous UNI Student Evaluation of Teaching had several related free response comments in the "Summary Comments" of the course evaluations:

- "Great class → the backwards method and supplement helped a lot. The videos may have been dry, but the information was thorough and effective for studying for tests as well as the online quizzes."
- "The flipped class style worked great, & the lectures were very well taught."
- "I really enjoyed the flipped class, as I am more interested in the advanced topics we started to cover at the end."
- "Good class didn't like the flipped part of it though."

6 Suggestions and Conclusions about Flipping a Course

Flipping a course can be a very time intensive process. I probably spent over 200 hours flipping the whole Computer Organization class, but I made videos, online quizzes, and repackaged my in-class activities. I would recommend that you start small and experiment by flipping a week or two unit of the course. You don't necessarily need to develop your own videos. You might find appropriate online video, or even ask the students to read a section of their textbook. Just make it clear to students that they need to prepare for class and providing some incentive for students to prepare for class.

Overall, I'm pleased with my flipped Computer Organization course, but it matched my pre-flipped teaching style of mini-lectures separated by active-learning questions done by students in class. By flipping the class, more class time can be devoted these activities.

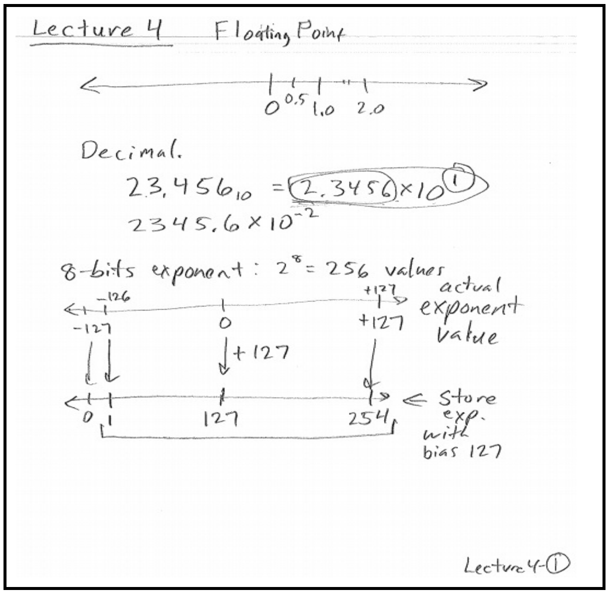
References

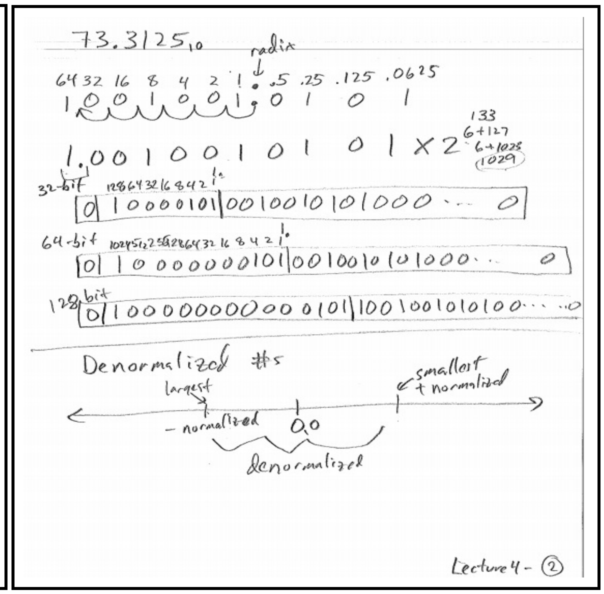
[1] Bloom, B. S.; Engelhart, M. D.; Furst, E. J.; Hill, W. H.; Krathwohl, D. R. *Taxonomy of educational objectives: The classification of educational goals*. Handbook I: Cognitive domain. New York: David McKay Company. 1956.

[2] Cynthia J. Brame, *Flipping the Classroom*. Retrieved from http://cft.vanderbilt.edu/guides-sub-pages/flipping-the-classroom/

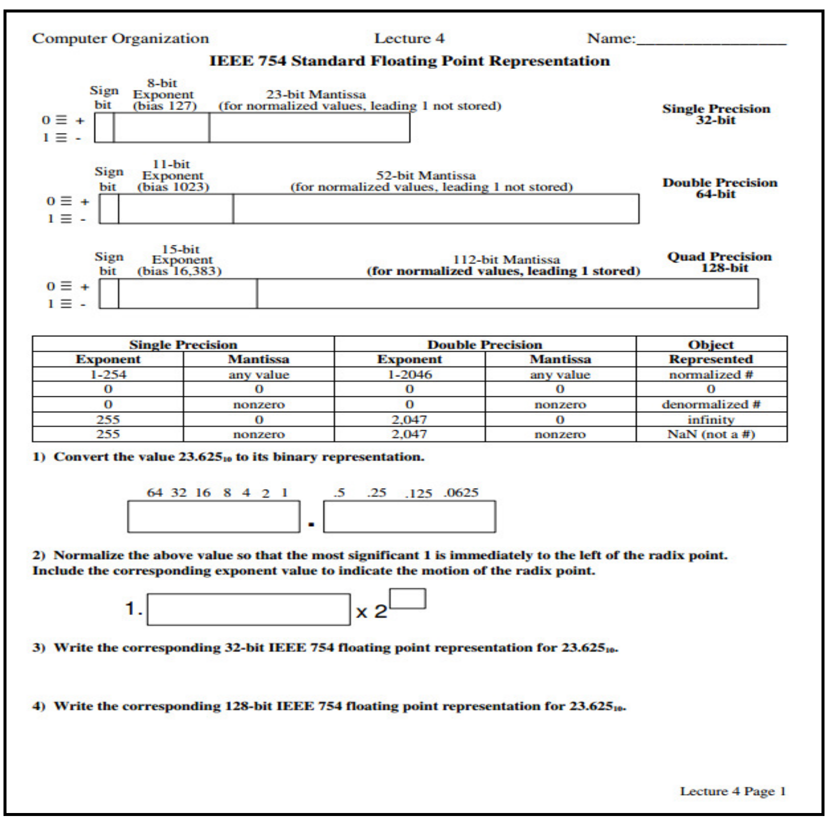
Appendix A: Sample Material from a Flipped Class

Handwritten pre-class video notes generated while creating the pre-class video:





Pre-class questions that students should answer **before** class while watching the pre-class video.



Computer Organization Lecture 4 Name:________________

IEEE 754 Standard Floating Point Representation

Sign bit | 8-bit Exponent (bias 127) | 23-bit Mantissa (for normalized values, leading 1 not stored) — **Single Precision 32-bit**

0 ≡ +
1 ≡ -

Sign bit | 11-bit Exponent (bias 1023) | 52-bit Mantissa (for normalized values, leading 1 not stored) — **Double Precision 64-bit**

0 ≡ +
1 ≡ -

Sign bit | 15-bit Exponent (bias 16,383) | 112-bit Mantissa **(for normalized values, leading 1 stored)** — **Quad Precision 128-bit**

0 ≡ +
1 ≡ -

Single Precision		Double Precision		Object
Exponent	**Mantissa**	**Exponent**	**Mantissa**	**Represented**
1-254	any value	1-2046	any value	normalized #
0	0	0	0	0
0	nonzero	0	nonzero	denormalized #
255	0	2,047	0	infinity
255	nonzero	2,047	nonzero	NaN (not a #)

1) Convert the value 23.625_{10} to its binary representation.

64 32 16 8 4 2 1 . .5 .25 .125 .0625

2) Normalize the above value so that the most significant 1 is immediately to the left of the radix point. Include the corresponding exponent value to indicate the motion of the radix point.

1.[   ] x 2[ ]

3) Write the corresponding 32-bit IEEE 754 floating point representation for 23.625_{10}.

4) Write the corresponding 128-bit IEEE 754 floating point representation for 23.625_{10}.

Lecture 4 Page 1

In-class questions that students will answer as part of their active-learning group activities.

Computer Organization Lecture 4 Name:________________

5) How would you add two normalized IEEE 754 floating point numbers?

6) Consider adding 1.011×2^{40} and 1.01×2^{5}.

a) How many places does the second number's mantissa get shifted?

b) After we add these two numbers and store the results back into a 32-bit IEEE 754 value, what would be the result?

7) How would you multiply two normalized IEEE 754 floating point numbers?

8) What would be the smallest positive normalized 32-bit IEEE 754 floating point value?

9) What would be the largest positive denormalized 32-bit IEEE 754 floating point value?
(denormalized: exponent is still bias 127, but no implied "1." (i.e., "leading 1 not stored"), but "0." instead)

10) What would be the smallest positive denormalized 32-bit IEEE 754 floating point value?

Lecture 4 Page 2