



**COLORADO SCHOOL OF MINES
ELECTRICAL ENGINEERING & COMPUTER SCIENCE DEPARTMENT**

**CSCI-410
Elements of Computing Systems
Spring 2014**

ECS-08A

Project ECS-08 differs from the reference ECS-08 (i.e., the one in the book) in two ways:

- 1) The project is broken into two stages that are to be submitted separately. This is consistent with the recommended approach described in the book (Sec 8.5) in which you first implement the program flow (ECS-08A) and then implement the function calling commands (ECS-08B).
- 2) Your translator must add annotations to the assembly code that is generated. These annotations are in addition to those incorporated as part of ECS-07.

Stage I (for ECS-08A)

Your submission needs to support all of the program flow commands. The applicable test files are BasicLoop and Fibonacci.

Stage II (for ECS-08B)

Your submission needs to support all of the function calling commands as well as generate the bootstrap code. The applicable test files are FibonacciElement and StaticsTest. Note that the SimpleFunction test is an intermediate step in that it tests the function and return commands but it does not require the bootstrap code. As a result, once you have passed this test and move on to implementing the full and complete translator, this (and all of the earlier) tests will no longer pass because they assume that the bootstrap code is not there.

Required Annotation

In addition to the header, VM file, and VM command comments required in ECS-07, your translator must now output summary information at the bottom of each VM file translation and at the end of the overall program translation. This summary is merely a tabulation of the input and output commands. It should look similar to the following (the exact format is up to you).

Suggested Summary Format

```
// -----  
// VM commands  
// -----  
// add..... 45  
// sub..... 321  
// neg..... 134  
// eq..... 23  
// lt..... 15  
// gt..... 73  
// and..... 3  
// or..... 0  
// not..... 12  
// push..... 455  
// pop..... 432  
// label..... 97  
// goto..... 124  
// if-goto.... 223  
// function... 5  
// call..... 24  
// return..... 5  
// -----  
// ASM instructions  
// -----  
// A-type..... 67  
// C-type..... 1458  
// L-type..... 75  
// -----
```

You should use the same table format for the summary that follows each VM file translation and the final summary at the very end. The final summary should reflect ALL of the code in that was translated and/or put into the asm file, including the bootstrap code.

You might consider outputting additional comments, such as a table giving the push/pop tally for each memory segment and/or a table listing all of the functions by name.

File Submission

Be sure to adhere to the File Header Formats and the Submission Procedures. Due to the bootstrap code, submitting the same files for both subprojects is not an option.