# EEAP 282
# INTRODUCTION TO MICROPROCESSORS

## COURSE OUTLINE

| Topic | Class notes | Ford & Topp, 2nd Ed. |
|---|---|---|
| Informational (1 lecture) | | |
| a. Course syllabus | 1-4 | |
| b. grading policies | 5 | |
| c. CWRUnet ethics code | 6-8 | |
| d class notes | 9 | |
| e. CWRUnet | 10 | |
| 1. connecting via PPP | | |
| 2. connecting to the Kern Lab | | |
| f. Where microprocessors came from | 11 | |
| Number systems and computer math (3 lectures) | | |
| a. representing numbers in binary | 12 | 16-17 |
| b. decimal-binary conversion,flow charts and pseudo code | 12-18 | 17-19,21-22 |
| c. binary fractions, hexadecimal | 19,20 | 17-18 |
| d. binary arithmetic, carrys, and overflow | 21-23 | 24-25 |
| e. fixed length binary numbers, 2's complement numbers | 24-25 | 25-37 |
| f. signed overflow, sign extension | 26-32 | 37-45 |
| g. representing characters in binary | 33 | 45-47 |
| h. logical functions | 34-35 | 51-53 |
| UNIX (1 lecture) | | |
| a. commands | 37-38 | |
| b. file structure | 39-43 | |
| c. help | 44 | |
| d mail | 45 | |
| e. news | 46 | |
| f. vi editor | 47-49 | |
| Basic computer architecture & operation (3 lectures) | 53-54 | |
| a. fetch-execute cycle | 55 | 59-61 |
| b. CPU | 56-57 | 61,64-65 |
| c. memory | 62 | 61-64 |
| d. bus | | 66-68 |
| e. program counter | 60 | |
| f. instruction coding | 61 | |

| | | |
|---|---|---|
| the 68000 microprocessor architecture (2 lectures) | | |
| a. basic 68000 architecture | 58 | 73-74 |
| b. 68000 register set | 59 | 75-78 |
| c. memory organization | 62 | 74-75 |
| d. basic 68000 machine code programming | 63-65 | 79-80 |
| e. MOVE, ADD, SUB instructions | 66-68 | 80-83 |
| f. disassembly of instructions | 69-73 | 90-92 |
| g. Programmer's Reference Manual | 74-77 | |
| h. extension words, machine cycles | 79-83 | 87-90 |
| Basic 68000 assembly language programming (3 lectures) | | |
| a. the debugger | 78 | |
| b. Example: simple program | | 102-104 |
| c. 68000 programming syntax | | 104-107 |
| d. assembler directives | 90-95 | 107-112 |
| e. the Status Register | 84-89 | 149-153, 643-646 |
| f. how to run programs | 96-97 | 112-118 |
| g. Example: how to assemble, link and debug a program | 98-110 | |
| Advanced 68000 assembly language programming (8 lectures) | | |
| a. addressing modes | | |
| 1. direct, immediate, absolute | 111-122 | 119-123 |
| 2. address register direct | 123-126 | 123-126 |
| 3. address register indirect | 127-135 | 126-128, 213-216,223,228 |
| b. Example: program w/labels | 136-143 | |
| c. error messages | 144-145,148 | 136-139 |
| d. Examples: programs w/errors | 149-151 | |
| e. Kern Lab i/o routines | 152-154 | 129-133 |
| f. simple branching | 155-158 | 153-160 |
| g. bit manipulation | 157 | 196-197 |
| h. structured programming | 159 | 170-174 |
| i. DBcc instruction | 160-165 | 180-183 |
| j. signed branching | 166-167 | 164-170 |
| k. unsigned branching | 155-158 | 176-178 |
| l. Example: parity program | 168-170 | 197-199 |
| m. Examples | 171-177 | |
| n. Rotate and shift instructions | 178-183 | 160-164 |
| o. Examples: rotate & shift | 184-194 | 188-189 |
| p. Math Instructions | 195-199 | 189-194 |
| q. Examples: math instructions | 200-205 | |

| | | |
|---|---|---|
| Stacks and subroutines (7 lectures) | | |
| a. stack and stack operations | 212-213 | 217-221 |
| b. Example: backward echo | 214-216 | |
| c. Example: RPN calculator | 217-219 | 221-223 |
| d. PC relative addressing | 220 | 231-236 |
| e. MOVEM instruction | 221-222, 227,251-252 | 276-281 |
| f. subroutines | 223-225 | 269-276 |
| g. passing parameters | 253,226-235 | 287-291 |
| h. recursion | 237-238,255-256 | 475-487 |
| i. stack frames | 239-243 | 291-296 |
| j. Example: LINK/UNLK | 244-247 | 296-300 |
| k. C/Pascal subroutine calling conventions | 248-250 | 465-472 |
| l. jump tables | 254 | 455-465 |
| m. Example: display 68000 registers | 257-260 | 300-302 |
| Systems programming (7 lectures) | | |
| a. microprocessor states (NORMAL, EXCEPTION and HALTED.) | 261-262 | 642-645 |
| b. user/supervisor mode, priviledged instructions | 262-263 | 645-648 |
| c. exception processing | 264-266 | 648-655 |
| d. Example: trace exception | 267-268 | 673-676 |
| e. Examples: reset/startup exceptions | 269-273 | 655-659 |
| f. traps & error handling | 274-278 | 660-664, 676-686 |
| g. Example: single step exception | 279-283 | 689-694 |
| h. Examples: 1010 and 1111 instruction traps | 285-288 | 664-669 |
| Advanced Microcomputer Architectures (7 lectures) | | |
| a. RISC and pipelined architectures | | |
| b. 40x/60x RISC processors | | |
| c. Intel Pentium | | |
| d. Texas Instruments TMS320 | | |

The total is 42 lectures with three reserved for in-class exams. This outline is tentative. If more time is spent on certain topics, the time spent on the remaining topics will be reduced.

Textbook:

**MC68000 Assembly Language & Systems Programming for the 68000 Family, Second Edition**

William Ford and William Topp

This textbook is available from the University Bookstore.

The following references will be available for purchase from the CWRU Electronic stores (3rd floor of Glennan) shortly after school starts:

M68000 Family Programmer's Reference Manual

M68000/AC Rev 2 Programmer's Reference Card

Office Hours:

Room 515 Glennan.

Monday & Wednesday 2:30- 3:15 10 minute meetings, Be prepared, be concise. Other times by appointment.

Honor Code:

All students are expected to abide by the CWRU Computing Code of Ethics and Student Code of Ethics. Sharing of ideas and comparison of answers on homework is acceptable, but copied work will not be accepted. In-class exams will be closed book and closed notes unless specifically stated. All work must be your own. Any reasonable suspicion of an honor violation will be reported to the Office of Collegiate Affairs.

Grading Policies:

Homework assignments, quiz due dates, etc. will be assigned via email and bulletin board. Failure to read your e-mail will not be an acceptable excuse. Assignments will be due at 5PM on the specified day. Late assignments will be penalized 10% per day late (weekends and holidays will not be counted for penalty purposes). Solutions will be posted on the Web. No student who fails to turn in two programming assignments without supporting evidence (medical excuse, etc.) will pass the course. Handing in an assignment late or incomplete is better than not doing it. Appropriate exceptions will be made for network and other computer failures.

Grade Changes:
A student requesting that the grade of an assignment or exam problem be reconsidered must provide the instructor with the following information **in writing** within two class periods after the work is returned to the class:

1. The number of the problem(s) to be reconsidered;
2. A description of your mistakes made in the problem(s); and
3. The reason that you feel that you should receive additional points for the problem(s), i.e. material the instructor did not see that was on the exam or homework paper.

Note: The entire exam or homework may be reconsidered by the instructor at this time.

Grading
This course uses outcome based grading. You are expected to show a minimum competency in computer usage (including CWRUnet and the Web), as well as competency in microprocessor architecture and programming. Your grade will be based upon the following scale. A modest number of extra credit assignments may be made throughout the course of the semester.

| | |
|---|---|
| Total points >90% | A |
| Total points >80% | B |
| Total points >70% | C |
| Total points >60% | D |

Tentative Lab Assignments and due dates:

| | |
|---|---|
| 1. Learning to use the lab hardware/software | September 12th* |
| 2. 68000 instructions and the SR | September 19th* |
| 3. polled i/o | October 6th* |
| 4. packet construction | October 17th* |
| 5. floating point division routine | October 31st* |
| 6. exception driven time sharing kernel | November 17th* |
| 7. unimplemented instruction | December 1st* |

NOTE: These labs may be replaced with others at my discretion.

EXAMS
ALL regular exams will be closed book, closed notes. Students will ONLY be allowed to use the 68000 Microprocessor Programmer's Reference Manual and Programmers Reference Card during exams; no other reference materials will be allowed.

## GRADING POLICY

| | | |
|---|---|---|
| Exam #1 | 12% | September 26th* |
| Exam #2 | 12% | October 15th* |
| Exam #3 | 12% | November 12th* |
| Exam #4 | 12% | December 5th* |
| Programming assignments | 30% | |
| Computer Quizzes† | 22% | |

†These are open book exams which will be adminstered and graded over the Web. This is an experimental component of the class. We plan to have the on-line quizzes ready shortly and will use the tentative schedule for quizzes:

| | |
|---|---|
| Quiz #1 | September 10th |
| Quiz #2 | September 15th |
| Quiz #3 | September 22nd |
| Quiz #4 | October 6th |
| Quiz #5 | October 17th |
| Quiz #6 | October 24th |
| Quiz #7 | November 3rd |
| Quiz #8 | November 10th |
| Quiz #9 | November 21st |

* All dates are tentative and subject to change.

The teaching assistants are:


**Kapil Sharma** **kxs81@eeap.cwru.edu**

**Wenhui Zhao** **wxz14@eeap.cwru.edu**

**Zhao Xiang Pan** **zxp5@eeap.cwru.edu**