

CS 273
Midterm Exam
March 3, 2004

The following exam is open book and open notes. You may feel free to use whatever additional reference material you wish, but **no electronic aids** are allowed. Please note the following instructions. There will be a ten point deduction for failure to comply with them:

- start each problem on a new sheet of paper
- write your social security number, but not your name, on each sheet of paper you turn in
- show your work whenever appropriate. There can be no partial credit unless I see how answers were arrived
- be succinct. You may lose points for facts that, while true, are not relevant to the question at hand

You have until 11:20 to finish the exam.

1. (10 points) Assuming the following equ's appear in a program:

```
claudius equ 10
gertrude equ $13
polonius equ $f903
```

translate the following assembly language statements into machine code.

- (a) `ldaa #claudius`
 - (b) `staa gertrude`
 - (c) `adda #polonius`
 - (d) `cmpb claudius,y`
2. (10 points) Translate the following machine code instructions into assembly language. You can leave any addresses or other constants as hexadecimal "magic numbers" (all of the numbers in the problem are given in hexadecimal)
- (a) `1B`
 - (b) `18 a4 37`
 - (c) `c1 13`
 - (d) `ce f8 03`
3. (25 points)
- (a) Convert the following decimal number into eight bit (signed) hexadecimal: -40
 - (b) Add it to the hexadecimal number `$ce` (giving an eight bit result). What is the result of the addition, and what would the condition codes be?
 - (c) If the next instruction is a `bpl`, will the branch be taken?
 - (d) Translate your signed result from step 3b to decimal.

4. (20 points) How many cycles will it take to execute the following code?

```
    ldab #10
loop inca
    decb
    bne loop
    staa bogus
```

(assume bogus is in RAM).

5. (35 points) Translate the following fragment of high level language code into HC11 code. Assume the variables are in RAM. You don't need to write the equ's, org's, and so forth; just translate this fragment in isolation.

```
richard = 43;
while (richard > 0) {
    if (richard < kyle)
        kyle = richard + kyle;
    richard = richard - 1;
}
```