

CS 273
Midterm Exam
October 9, 2009

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 Banner ID number, but not your name, on each sheet of paper you turn in

Also, please note the following:

- show your work whenever appropriate. There can be no partial credit unless you show how you derived your answers
- 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. The questions are equally weighted.

1. Suppose you have the following assembly language fragment, defining symbols for a program:

```
RAM      equ 00
EEPROM   equ $f800

          org RAM
i        rmb 5
j        rmb 1

          org EEPROM
k        fcb 2, 3
l        fcb 7
```

- (a) Assemble each of the following instructions.

```
i. ldaa l
ii. suba #j
iii. asl i
```

- (b) Disassemble the following instructions. As on HW2, use symbols in your operands (e.g. use *i*, not 0)

```
i. d0 00
ii. b8 f8 00
iii. 18 60 00
```

2. Convert the following numbers to 2's complement, 8-bit hexadecimal.

- (a) 39
(b) -53

3. Convert the following 2's complement, 8-bit hexadecimal numbers to decimal.

- (a) d4
(b) 3a

4. Perform each of the following 8-bit hexadecimal additions, obtaining both the value and condition codes. Then determine whether the branch instruction is taken.

(a) $3a + 77$, ble

(b) $8c + 63$, bhi

5. If the following code sequence is executed, what is the final value of any registers and memory locations that are changed by it (including PC and condition codes)?

```
org $f800
ldaa #$5a
asla
staa 04
```

6. Assuming variable *i* is in RAM and variable *a* is in the A accumulator, translate the following high level language program fragments to assembly code. Assume the variables are signed. These are only fragments, so you *don't* need to give me any of the assembler directives that would be necessary in a complete program (no equ, org, end, etc.)

(a) if ($a < i$)

```
    j = a;
```

(b) do {

```
    a = a - 1
```

```
    } while (a != 0);
```