

---

# ELE538 Quiz/Answers (2004)

Name: \_\_\_\_\_ Student #: \_\_\_\_\_ Time: 30 minutes

Chun, Clowes, Guerkov

## Table of Contents

Reference Material .....	1
Instructions .....	1
A/D system .....	1
Questions .....	2

Answer all questions. All questions have equal weight.

## Reference Material

This material contains technical details that may be required to answer certain questions.

## Instructions

**Table 1. Instruction Details (Abridged)**

Assembler	Mode	Encoding	Cycles
ldaa	IMM	86 ii	2
ldab	IMM	C6 ii	2
mul	INH	3D	10

## A/D system

The bits in the Control/Status register (ADCTL, mapped to address 0x1030) are:

**Figure 1. AD Control/Status Register**

7	6	5	4	3	2	1	0
CCF	-	M*	S	0	n	n	n

The interpretation of the bits is:

CCF    0: conversion NOT complete; 1: conversion complete.

M\*    0: Convert 4 channels; 1: Convert single channel.

S      0: continuous conversion; 1: one-shot conversion.

nnn Channel number (0-7).

## Questions

1. Given the following program:

```
                org $6000
main:           ldx #stuff
                ldab #0

loop:           ldaa 0,x
                bmi done
                addb 0,x
                inx
                bra loop
done:           swi

                org $7000
stuff           fcb 3, 1, 4, 0, -1, 2, 7
```

Assume that the CPU begins executing at address 0x6000.

- a. What value (in hex) will be in index register X following the execution of the instruction `ldx #stuff`?
- b. What will the values in index register X, AccA and AccB be just before the `swi` instruction is executed?

## ANSWER

- a.  
\$7000
- b.  
AccA: -1 (i.e. \$FF)  
AccB: 8

IX: \$7004

2. The following program uses the A/D converter subsystem to read some voltages. The program does work. (i.e. there are no logical or syntactical errors.)

```
; A simple program using adc module.
; Author: Foo Bar
; Date: October 6, 2004

ADCTL equ $1030      ;address of ADC Control register
ADR1  equ ADCTL+1     ;address of first result register
ADR2  equ ADCTL+2     ;address of second result register
ADR3  equ ADCTL+3     ;address of third result register
ADR4  equ ADCTL+4     ;address of fourth result register

        org $6000
main:
        ldaa #%00010000
        staa ADCTL

        jsr foo

        ldaa ADR1
        ldab ADR4
        swi

foo:
        tst ADCTL
        bpl foo
        rts
```

- a. The subroutine "foo" performs an essential task. The name of the subroutine, however, is poorly chosen since it does not hint at the task it performs.

What is a better name for the subroutine?

- b. Suppose that all 8 analog channels are connected to DC voltages as follows (assume that "full scale analog voltage" is 5.0 V):

Channel 1: 2.5 V  
Channel 2: 1.25 V  
Channel 3: 3.75 V  
Channel 4: 5.0 V

Channel 5: 0.0 V  
Channel 6: 3.75 V  
Channel 7: 5.0 V  
Channel 8: 2.5 V

The program is run from address 0x6000. What values will be in Acc. A and Acc B. when the "swi" instruction is encountered?

## ANSWER

- a. A better name would be something like `WaitConvDone`.
  - b. Channels 0-4 are converted. Channel 1 is 2.5 volts (analog), converted to digital %10000000. Channel 4 is 5.0 volts (analog), converted to digital %11111111. Since AccA reads Channel 1, it is \$80; since AccB reads Channel 4, it is \$FF.
3. Write a code fragment that performs the following:
- a. Divides the unsigned 8-bit binary number in Acc by 2. (For example, if AccA were 8, it would be 4 after division.)
  - b. If Acc A is an even number, convert it to the next bigger odd number. (For example, a 4 would become a 5.)
  - c. Invert the bits 2 and 1 of the result. (For example, 5—00000101 in binary— would become 3—00000011 in binary.)

## ANSWER

```
lsra      ;Part a: logical right shift divides by 2 (unsigned)
oraa #1    ;Part b: no effect on odd numbers; adds 1 to even numbers
eor a #110 ;Part c: XORing with 1 inverts bit at same position
```