ELE538 Quiz Study Guide

Prof. K. Clowes

Prof. V. Geurkov

4 October 2004

Revision History

Revision 1.0	4 October 2004	First public distribution.

Table of Contents

What is a Study Guide?

This study guide is made up of questions relating to many of the main topics that will be covered in the Quiz. The Quiz includes the material covered in the first 3 weeks of lectures and the first 2 labs.

The study guide is **not exhaustive** (i.e. not all topics you are responsible for are covered.)

You **cannot assume** that Quiz questions will bear any resemblance to the questions in this Study Guide. Memorizing canned answers is not a good idea; use the questions/answers to help identify areas you need to bone up on.

You should attempt to answer the questions **before** peeking at the answers. If you cannot understand an answer after reviewing the core material, ask you prof for help.

The questions should be considered as “closed book”—i.e. you should be able to answer them without reference to a text book or data sheet. However, some of the questions do require technical details that you are not expected to have memorized. (For example, what is the machine language translation of `ldaa #3` or which bit is used to tell the A/D subsystem to perform continuous conversions?) For these cases, a *Reference Material* section is included.

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

Q: General: bus specs

A typical Pentium PC has data and address bus widths of 32 bits and a bus cycle time of 2.5 ns (400 MHz). The 6811 in the lab has a bus cycle time of 1 µs (1 MHz).

1. What are the widths of the address and data buses for the 68hc11 microprocessor?
2. Based on knowledge of the bus specifications alone, how much faster is the typical PC than the 68hc11 in terms of maximum possible information transfer rate.
3. Suppose both systems are connected to an Ethernet local area network using the IEEE 802.11b standard (whose maximum transfer rate is 11 Mbps.) Which of the two systems (if any) can keep up with this speed? If a system can keep up, what percentage of the maximum bus transfer rate is used for the Ethernet connection?

A: 1. The 6811 has a 16-bit address bus and 8-bit data bus.

2. The PC can transfer data at a maximum rate (in Mega bits per second) of 400*32 = 12800 Mbps; the 6811's maximum rate is 8 Mbps. Hence the PC is 1600 times faster.

3. The 6811 cannot keep up. The PC would require less than 0.1% of its bus capacity (0.08593% if you insist on more precise numbers.)

Q: General (Lab 1): results

Given the following assembly code:

```
ldaa #5
ldab #3
mul
```

What are the contents of AccA and AccB after the `mul` instruction has been executed? Give your answer in hexadecimal.

A: The A and B accumulators will be:

AccA: 0x00
AccB: 0x0F

Q: General: simple loops with indexed addressing

Given the following source code:;

```
      org $6000    ;Line 1
      clra         ;Line 2
      ldx #data    ;Line 3
loop               ;Line 4
      tst 0,x      ;Line 5
      bmi done     ;Line 6
      adda 0,x     ;Line 7
      inx          ;Line 8
      bra loop     ;Line 9
done  swi          ;Line 10

      org $7000    ;Line 11
data  fcb  3       ;Line 12
      fcb  0       ;Line 13
      fcb  1       ;Line 14
      fcb  -1      ;Line 15
      fcb  128     ;Line 16
      fcb  -1      ;Line 17
```

Assume that the 6811 starts executing instructions at address 0x6000.

1. What is the value of index register X following the "`ldx #data`" instruction on line 3?

2. How may times will the instruction `inx` be executed?

3. What will the contents of AccA be just before executing the `swi` instruction?

4. Suppose that Line 15 were changed to `fcb 0` (instead of `fcb -1`) and the code run again from the beginning. What would AccA contain before the `swi` instruction?

A: 1. Index register X will have the value 0x7000 (i.e. the value of the symbol `data`.)

2. It will be executed 3 times.

3. 4.

4. 4. (Note `fcb 128` would be interpreted as a negative number!)

Q: General: simple loops

Given the following source code:;

```
      org $6000 ;Line 1
      ldaa #3   ;Line 2
      clrb      ;Line 3
loop            ;Line 4
      addb #2   ;Line 5
      deca      ;Line 6
      bne loop  ;Line 7
      swi       ;Line 8
```

1. When the `swi` is executed, what are the contents of AccA and AccB?

2. At what address will the machine code `ldaa #3 ;Line 2` be assembled? How many bytes does the machine language use and what are these byte(s)?

A: 1. Just before the `swi` instruction, we have:

AccA: 0x00
AccB: 0x06

2. It will be assembled at 0x6000; the machine language requires 2 bytes: `0x8603`.

Q: General: Bit manipulation

Write assembly language code to modify Accumulator A so that the most significant bit is set to a '1', the least significant bit is cleared to '0' and all the other bits are inverted. (For example, if the initial contents were 11110111, they would become 10001000.)

A: There are many ways to answer this question.

A straightforward way is:

```
ora #$80        ;Set MSB
anda #$FE       ;Clear LSB
eora #$7E       ;Invert other bits
```

NOTE: the order of execution of these 3 instructions is irrelevant.

Another (slightly more efficient) way is:

```
anda #%01111110   ;Clear MSB and LSB
eora #%11111110   ;Invert MSB (to ONE) and invert middle 6 bits
```