UNIVERSITY OF VICTORIA
EXAMINATIONS APRIL 2002

C SC 230 Computer Architecture and Assembly Language

NAME (print): _____ REG NO. _____

SIGNATURE: _____ DURATION: 3 hours

INSTRUCTOR: D. M. Miller

TO BE ANSWERED ON THIS EXAMINATION PAPER.

STUDENTS MUST COUNT THE NUMBER OF PAGES IN THIS EXAMINATION PAPER BEFORE BEGINNING TO WRITE, AND REPORT ANY DISCREPANCY IMMEDIATELY TO THE INVIGILATOR.

THIS EXAMINATION HAS **SEVEN** PAGES PLUS THIS COVER PAGE.

ATTEMPT EVERY QUESTION. ANSWER IN THE SPACES PROVIDED (YOU DO NOT NECESSARILY HAVE TO USE ALL LINES PROVIDED AND MAY USE OTHER AREAS ON THE **FRONTS** OF THE PAGES IF NECESSARY). USE THE BACKS OF THE PAGES FOR ROUGH WORK.

THIS IS A CLOSED BOOK EXAMINATION. NO COURSE NOTES, BOOKS OR CALCULATORS, ARE PERMITTED.

YOU **ARE** PERMITTED TO USE THE HANDOUT PROVIDED.

QUESTION	MAX. MARK	STUDENT'S MARK
1	15	
2	13	
3	6	
4	6	
5	20	
6	20	
7	20	
TOTAL	100	

1. (15 marks) Circle the correct answer for each of the following:

(a) A hardware stack is essential for an effective RISC desogn. True False

(b) On an 8 MHz 6811 processor TCNT overflows at intervals of 2^{13} micro-seconds. True False

(c) Parity allows for the detection and correction of single bit errors. True False

(d) On the 6811, the external address and data buses are synchronous. True False

(e) Memory-mapped I/O requires dedicated I/O instructions. True False

(f) A processor must have a hardware stack in order to support a jump to subroutine instruction. True False

(g) For the 6811, subroutine parameters can not be passed on the stack when the code is to be stored in ROM. True False

(h) On-processor memory can share addresses with off-processor memory. True False

(i) An arithmetic shift always preserves the sign of a numeric value. True False

(j) For the 6811, unless explicitly programmed by the user, an interrupt source can interrupt the service of an interrupt of a different type. True False

(k) Interrupts should never be used if polling can be used instead. True False

(l) The 6811 is an example of a RISC design. True False

(m) The PENTIUM is an example of a CISC design. True False

(n) The data bus to main memory must have the same number of bits as the word size. True False

(o) Certain PENTIUM processors use SIMD techniques to implement MMX. True False

2. (13 marks) Answer each of the following in the space provided:

(a) (3 marks) Why does *caching* improve processor performance?

(b) (3 marks) What does the term *branch prediction* mean?

(c) (5 marks) State five significant features of RISC design that are intended to contribute to execution speed.

(d) (2 marks) What does it mean to say a processor is (a) *scalar*, (b) *superscalar*?

(a) _____

(b) _____

3. (6 marks) Explain the term *pipeline* in relation to processor design:

4. (6 marks) Assuming ACCB is available for use, write the shortest sequence of 6811 instructions you can to multiply the contents of ACCA by 10 without using the MUL instruction.

5. (20 marks) Consider the following segment of 6811 code:

```
LDX  #LIST
PSHX
LDAA N
PSHA
JSR  AMIN
INS
INS
INS
```

(*continued on next page*)

(*Question 5 continued*) LIST is the starting address of a list of **two-byte** two's complement integers (stored high-endian) and N is the address of an unsigned **one-byte** value that is the number of values in the list. You are to write a **6811 assembly language function** called AMIN which accepts two parameters as shown above and that **returns in IY the address** of the smallest value in the list. If the smallest value occurs more than once, your function should return the address of the **last** occurrence in the list. You are to assume the list contains at least one value, i.e. N > 0. The values in the list are not in any specific order. ACCA, ACCB and IX should be protected. Note that it is not necessary to sort the list.

(start answer in the column)

(use this column if necessary)

6. (20 marks) Consider the following programs:

```
SBASE   equ     $DFFF
        org     $D000
str1    fcc     "THIS IS A TEST"
        fcb     0
        org     $C000
start   lds     #SBASE
        ldx     #str1
        pshx
        jsr     scroll
        ins
        ins
        stop
;-------------------------------------------------
scroll
        end
```

```
#include <hc11.h>

void scroll(char *);
void lcdchar(int ,char );

void main(void)
{
  char *str = 'THIS IS A TEST";

  display(str);
}

void scroll(char string)
{
}
```

You are to write the subroutine 'scroll' **(in 6811 assembly language or C – not both)** so that it scrolls the message across the Wookie 4-character display. The display of the message is to begin by showing just the first character of the message in the rightmost position of the display and is to end with a display showing just the last character of the message in the leftmost position of the display. The message is scrolled one character position every second. **NOTE**: The functionality is a bit different from the question on assignment 5.

You are to assume you have a routine with no parameters called *dlysec* which when called delays for one second and then returns. You are also to assume you have a subroutine called *lcdchar* that displays a character on the Wookie display. The C header is shown above. In assembly language, it accepts two one-byte parameters on the stack: the position (3 – 0) and the character to be displayed where the position should be pushed first onto the stack followed by the desired character. Recall that position 0 is the rightmost character position for the display. **You do not have to write dlysec or lcdchar**.

(*answer on the next page*)

(*answer for question 6*) ***Include comments to explain what your code is doing.***

(start answer in the column) (use this column if necessary)

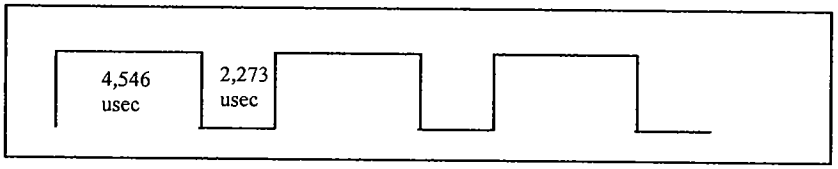
7. (20 marks) You may answer this question using **C or 6811 assembly language**. If you use C, you may include the file hc11.h. ***Assume the processor has a 2MHZ clock***.

   Write a complete C or 6811 assembly language program that behaves as follows:

   (a) After appropriate initialization, a **RISING** edge on input capture pin 1 (IC1), **detected by an interrupt**, causes the program to produce the wave depicted below on bit 6 of Port A.

   (b) A **FALLING** edge on IC1, **detected by an interrupt** stops the production of the wave as quickly as possible i.e. the program does not wait to complete the current cycle. Bit 6 of Port A is to be set to 0 when stopping the wave production.

   The wave to be produced is a repeating signal for which each cycle consists of high for 4,546 usec. and then low for 2,273 usec. as shown below: Use **OC3 and polling** for the required timing



Include comments to explain what your code is doing.

(start answer in the column) (use this column next)

(continue here if necessary)

(use this column last)

End of Examination